Uniwersytet Ekonomiczny w Krakowie

Dziedzina nauki: Nauki społeczne

Dyscyplina naukowa: Ekonomia i finanse

Piotr Bednarski

Wykorzystanie algorytmów NLP do automatycznej analizy treści informacji finansowych w procesach prognozowania notowań giełdowych

Rozprawa doktorska

Promotor: prof. UEK dr hab. inż. Janusz Morajda

Promotor pomocniczy: dr Roman Huptas

Kraków, 2024

Podziękowania

Składam serdeczne podziękowania Promotorom – Panu prof. UEK dr. hab. inż. Januszowi Morajdzie oraz Panu dr. Romanowi Huptasowi za opiekę naukową w czasie studiów doktoranckich, liczne dyskusje, a także nieocenioną pomoc merytoryczną podczas tworzenia niniejszej pracy. Dziękuję za poświęcony czas i wsparcie.

Pragnę też podziękować całej mojej Rodzinie za cierpliwość, wyrozumiałość i wiarę we mnie - szczególnie mojej Mamie za niezliczone korekty edytorskie oraz mojej Dziewczynie Ani za weryfikowanie sensowności moich pomysłów w trakcie naszych spacerów.

Spis treści

Wykaz użytych skrótów 5

Lista symboli 8

Wstęp 9

Rozdział 1. Informacja na rynkach finansowych oraz jej wykorzystanie w procesach prognozowania 18

1.1. Charakterystyka rynków finansowych 18

1.1.1. Pojęcie rynku 18

1.1.2. Rola rynków finansowych 21

1.1.3. Państwo a rynki finansowe 22

1.1.4. Kategoryzacja rynków finansowych 22

1.1.5. Instrumenty finansowe 25

1.1.6. Uczestnicy rynku 27

1.1.7. Hipoteza rynku efektywnego 30

1.2. Źródła informacji rynkowej 32

1.3. Wpływ informacji na decyzje inwestorów 37

1.4. Ekonometria finansowa a informacje na rynkach finansowych 42

1.5. Podsumowanie 47

Rozdział 2. Modelowanie i prognozowanie jednowymiarowych finansowych szeregów czasowych 49

2.1. Modele klasy ARMA-GARCH 49

2.1.1. Modele ARMA 50

2.1.2. Modele GARCH jako narzędzie analizy zmienności finansowych szeregów czasowych 52

2.2. Modele uczenia maszynowego 59

2.3. Podsumowanie 64

Rozdział 3. Współczesne techniki NLP oraz możliwości ich zastosowania do analizy informacji finansowych 65

3.1. Historia rozwoju NLP 65

3.1.1. Początki - tłumaczenie maszynowe i systemy regułowe 65

3.1.2. Podejście statystyczne 67

3.1.3. Sieci neuronowe 69

3.2. Zadania NLP 70

3.2.1. Reprezentacja tekstu 71

3.2.2. Uzupełnianie tekstu 75

3.2.3. Ekstrakcja informacji zawartych w tekście 77

3.2.4. Analiza sentymentu 77

3.2.5. Pozostałe zadania NLP 78

3.3. Współczesne osiągnięcia w przetwarzaniu języka naturalnego 79

3.3.1. Budowa modeli językowych 79

3.3.2. *Fine-tuning* modeli NLP 82

3.3.3. Benchmarki modeli językowych 83

3.4. Narzędzia i modele stosowane obecnie w NLP 84

3.4.1. Język programowania i biblioteki 84

3.4.2. Infrastruktura obliczeniowa 86

3.4.3. Zbiory danych i pre-trenowane modele 86

3.5. Automatyczna analiza wiadomości finansowych 87

3.5.1. Statystyczna analiza tekstu wiadomości 88

3.5.2. Predykcja oparta o wydarzenia 90

3.5.3. Analiza sentymentu 91

3.5.4. Pozostałe podejścia 92

3.6. Podsumowanie 94

Rozdział 4. Proponowana metodologia badań wykorzystania technik NLP w analizie treści wiadomości finansowych 96

4.1. Metodyka weryfikacji hipotez badawczych 96

4.2. Architektura autorskiego systemu prognozowania notowań giełdowych 99

4.2.1. Przygotowanie danych 100

4.2.2. Estymacja i ewaluacja modeli 108

4.3. Propozycja modelu predykcji wspomaganej analizą treści wiadomości 111

4.4. Trening i walidacja modeli 120

4.4.1. Podział na zbiór treningowy, walidacyjny i testowy 120

4.4.2. Uczenie transferowe 122
4.4.3. *Fine-tuning* modeli 123
4.5. Budowa i ocena krótkookresowych prognoz stóp zwrotu i cen analizowanych instrumentów finansowych 127
4.5.1. Badane instrumenty finansowe 127
4.5.2. Schemat prognozowania i walidacja modeli 134
4.5.3. Metody oceny trafności prognoz 136
4.6. Podsumowanie 140

Rozdział 5. Badania efektywności predykcji finansowych szeregów czasowych wspomaganej narzędziami NLP 141

5.1. Wybór modeli referencyjnych 141
5.2. Analiza efektywności autorskich modeli 145
5.2.1. Ogólne porównanie modeli LSTM-FENE z modelami bazowymi 145
5.2.2. Porównanie zachowania modeli po wystąpieniu nowych wiadomości . . 152
5.3. Rodzaje wiadomości i ich wpływ na notowania giełdowe 165
5.3.1. Ocena jakości różnych źródeł informacji 165
5.3.2. Badania siły efektu wiadomości na różnych typach rynków 167
5.4. Zależność jakości predykcji od liczby wiadomości finansowych 168
5.4.1. Badania wpływu dziennej liczby wiadomości na podatność wycen na zmiany 168
5.4.2. Długość pojedynczych wiadomości a ilość niesionych informacji 172
5.5. Jakość proponowanych modeli w zależności od hiperparametrów 174
5.5.1. Dobór hiperparametrów modelu NLP 174
5.5.2. Dobór hiperparametrów modeli szeregów czasowych 176
5.6. Wykorzystanie nagłówków wiadomości zamiast ich pełnej treści 179
5.7. Model LSTM bez wiadomości finansowych 184
5.8. Podsumowanie wyników badań 188

Rozdział 6. Wykorzystanie modeli hybrydowych ARMA-GARCH-FENE oraz analiza ich efektywności 191

6.1. Propozycja architektury hybrydowych modeli 191
6.2. Estymacja i wybór specyfikacji hybrydowych modeli 193
6.3. Analiza efektywności hybrydowych modeli 197
6.3.1. Porównanie modeli hybrydowych z modelami bazowymi 197
6.3.2. Porównanie modeli hybrydowych z autorskim modelem LSTM-FENE . 203
6.4. Podsumowanie wyników badań 208

Rozdział 7. Konkluzje i podsumowanie 209

7.1. Metodologia badawcza 210

7.2. Wyniki eksperymentów 211

7.2.1. Autorski model LSTM-FENE 211

7.2.2. Hybrydowe modele wykorzystujące techniki NLP 212

7.2.3. Implikacje teoretyczne i praktyczne 214

7.3. Ograniczenia i perspektywy dalszych badań 217

7.4. Podsumowanie 220

Bibliografia 222

Aneks 233

A.1. Autorskie kody źródłowe programów 233

A.1.1. Pobieranie treści wiadomości finansowych 233

A.1.2. Pobieranie historycznych notowań giełdowych 246

A.1.3. Generowanie zmiennych modelowanych 253

A.1.4. Kod TensorFlow autorskiego modelu 259

A.2. Wybór specyfikacji i rzędów opóźnień bazowych modeli ARMA-GARCH . . . 263

A.3. Wybór specyfikacji i rzędów opóźnień modeli ARMA-GARCH-FENE 274

A.3.1. Modele z enkoderem o długości $d_{encoder} = 4$ 279

A.3.2. Modele z enkoderem o długości $d_{encoder} = 8$ 293

A.3.3. Modele z enkoderem o długości $d_{encoder} = 16$ 301

A.3.4. Podsumowanie wartości kryterium BIC dla modeli hybrydowych . . . 311

Wykaz użytych skrótów

ACF funkcja autokorelacji - ang. *Autocorrelation Function*. 50, 51, 121

AIC Kryterium Informacyjne Akaikego - ang. *Akaike Information Criterion*. 51, 58, 141–143, 193–197

APT teoria arbitrażu cenowego - ang. *Arbitrage Pricing Theory*. 30

AR model autoregresyjny - ang. *Autoregressive model*. 50

ARCH model autoregresyjny z warunkową heteroskedastycznością - ang. *Autoregressive Conditional Heteroscedascity*. 44, 53

ARIMA autoregresyjny zintegrowany model średniej ruchomej - ang. *Autoregressive Integrated Moving Average*. 43, 44, 50, 58, 64

ARMA autoregresyjny model średniej ruchomej - ang. *Autoregressive Moving Average*. 49, 50, 52–54, 57, 58, 96, 98, 99, 103, 104, 108–111, 120, 121, 140–152, 160, 161, 163, 165–168, 174–189, 191–213, 216, 217, 311–313

BERT model oparty o dwukierunkowe enkodery z siecią typu transformer - ang. *Bidirectional Encoder Representations from Transformers*. 69, 91, 92, 172, 175, 189, 210

BIC Bayesowskie Kryterium Informacyjne Schwarza - ang. *Bayesian Information Criterion*. 51, 58, 141, 142, 144, 193–195, 311–313

BoW Worek Słów - ang. *Bag of Words*. 72, 73

CAPM model wyceny aktywów kapitałowych - ang. *Capital Asset Pricing Model*. 30

CBOW Ciągły Worek Słów - ang. *Continuous Bag of Words*. 76

CPU centralna jednostka przetwarzająca - ang. *Central Processing Unit*. 71, 85

EGARCH wykładniczy model GARCH - ang. *Exponential GARCH*. 55, 141, 196, 208

FENE Enkoder Wiadomości Ekonomicznych i Finansowych - ang. *Financial and Economic News Encoder*. 11, 14–16, 96–99, 108–114, 118, 120, 140, 145–152, 160, 161, 163, 165–168, 174–189, 191–218, 220, 311–313

GARCH uogólniony model autoregresyjny z warunkową heteroskedastycznością - ang. *Generalized Autoregressive Conditional Heteroscedascity*. 43, 44, 47, 49, 53–58, 64, 96, 98, 99, 103, 104, 108–111, 120, 121, 140–152, 160, 161, 163, 165–168, 174–189, 191–213, 216, 217, 311–313

GPU procesor graficzny - ang. *Graphics Processing Unit*. 71, 81, 85, 86, 88, 110, 111

LM model językowy - ang. *Language Model*. 79, 83, 84, 87, 114, 124–126

LSTM Długa Pamięć Krótkotrwała - ang. *Long Short-Term Memory*. 62, 63, 69, 78, 88, 92, 96–99, 108–110, 112–114, 118, 119, 140, 145–152, 160, 161, 163, 165–168, 174–189, 192, 197, 198, 203–215, 217, 218, 220

MA model średniej ruchomej - ang. *Moving Average model*. 50

MACD porównanie krótko- i długoterminowych średnich wykładniczych - ang. *Moving Average Convergence / Divergence*. 33

MAE średni błąd bezwzględny - ang. *Mean Absolute Error*. 137, 138, 145–148, 150, 151, 166–173, 175–180, 182, 184, 185, 187, 197–199, 201–204, 206, 211, 215

MAPE średni bezwzględny błąd procentowy- ang. *Mean Absolute Percentage Error*. 138, 147, 148, 151, 167, 168, 175, 177, 178, 180, 182, 185, 187

ME średni błąd prognozy - ang. *Mean Error*. 137

ML uczenie maszynowe - ang. *Machine Learning*. 9, 59, 60, 64, 73, 77, 79, 86, 87, 122

MPE średni błąd procentowy- ang. *Mean Percentage Error*. 138

MSE błąd średniokwadratowy - ang. *Mean Squared Error*. 137, 138

NER rozpoznawanie jednostek referencyjnych - ang. *Named Entity Recognition*. 70, 71, 94, 125

NLP Przetwarznie Języka Naturalnego - ang. *Natural Language Processing*. 10–16, 46, 48, 65–71, 76, 77, 79, 83–89, 91–98, 114, 124–126, 140, 165, 208–210, 214, 216, 217, 219, 220

OCR optyczne rozpoznawanie pisma - ang. *Optical Character Recognition*. 70

PACF funkcja autokorelacji cząstkowej - ang. *Partial Autocorrelation Function*. 51, 121

RMSE pierwiastek błędu średniokwadratowego - ang. *Root Mean Squared Error*. 93, 138, 145–160, 166–173, 175–189, 197–207, 211, 212, 215

RNN Rekurencyjna Sieć Neuronowa - ang. *Recurrent Neural Network*. 61–63, 69

RSI oscylator siły trendu - ang. *Relative Strength Index*. 33

SARIMA sezonowy autoregresyjny zintegrowany model średniej ruchomej - ang. *Seasonal Autoregressive Integrated Moving Average*. 44

TL uczenie transferowe - ang. *Transfer Learning*. 69, 83, 120, 122, 123

TPU procesor tensorowy - ang. *Tensor Processing Unit*. 71, 81, 85, 86, 88, 110, 111

VAR model wektorowej autoregresji - ang. *Vector Autoregression model*. 46

Lista symboli

ε_t	Wartość zaburzenia losowego w chwili t.
$\mathbb{E}$	Operator wartości oczekiwanej.
h	Horyzont czasowy prognozy.
μ	Wartość oczekiwana modelowanego procesu stochastycznego.
$p_{j,t}$	Cena instrumentu finansowego j w chwili t.
R_t	Stopa zwrotu w chwili t.
r_t	Logarytmiczna stopa zwrotu w chwili t.
σ	Odchylenie standardowe.
σ^2	Wariancja.
t	Czas.
X_t	Wartość procesu stochastycznego w chwili t.

Wstęp

Prognozowanie finansowych szeregów czasowych i możliwość przewidywania przyszłych stóp zwrotu różnych aktywów jest zagadnieniem wyjątkowo ciekawym i wartościowym, zarówno z naukowego jak i praktycznego punktu widzenia. Początkowo było rozwijane głównie na gruncie klasycznej ekonometrii finansowej, jednak z czasem zaczęły pojawiać się wpływy modelowania komputerowego, a w końcu także uczenia maszynowego, wykorzystującego głównie ateoretyczne modele.

Ilościowe spojrzenie na współczesną ekonomię finansową składa się między innymi z rozważań na temat przewidywalności zwrotów z inwestycji, analiz portfelowych, modeli behawioralnych oraz hipotezy rynku efektywnego. Jedną z publikacji, która przybliża wymienione zagadnienia jest książka Cuthbertsona K. i Nitzschego D. z 2004 roku [33]. Bardziej techniczne podejście do metod modelowania szeregów czasowych oraz finansowej ekonometrii zostało opisane przez Brooksa C. Jego publikacja jest w zasadzie podręcznikiem, pokazującym zbiorczo wszystkie najważniejsze pojęcia w dziedzinie klasycznego prognozowania finansowych szeregów czasowych [14].

Obecnie coraz częściej w ekonometrii stosowane są techniki uczenia maszynowego. Sposób, w jaki uczenie maszynowe - ang. *Machine Learning* (ML) wzbogaca współczesne podejście do modelowania w makroekonomii, został opisany przez m.in. Coulombe'a P. i in. w publikacji z 2020 roku [32].

Skuteczny model prognozowania przyszłych stóp zwrotów jest kluczowy dla inwestorów, którzy świadomie budują swoje portfele i nie polegają wyłącznie na intuicji. Całkowite wyeliminowanie elementu losowego z predykcji przyszłych stóp zwrotu jest niemożliwe choćby przez niecałkowitą racjonalność wszystkich inwestorów na rynkach finansowych. Uwzględnienie zatem w modelach predykcyjnych - oprócz danych ilościowych pochodzących z notowań - także informacji pochodzących z publikowanych na bieżąco informacji rynkowych, na których inwestorzy opierają swoje decyzje, może

pozwolić na poprawę jakości prognoz. Fakt ten stanowi mocną przesłankę dla podjęcia tej problematyki we współczesnych badaniach naukowych, m.in. w ramach niniejszej dysertacji.

W dobie Internetu przepływ informacji między uczestnikami rynków jest wyjątkowo dynamiczny. Spółki publikują swoje raporty finansowe, informacje o podziałach (*splitach*) akcji, przejęciach konkurencji lub przez konkurencję. Serwisy informacyjne (gazety, magazyny, strony z wiadomościami, blogi) na bieżąco dostarczają już przetworzone informacje o spółkach, nowościach o produktach, skandalach z udziałem ważniejszych pracowników i zarządu spółek, wiadomościach gospodarczych i politycznych. W mediach społecznościowych inwestorzy (głównie detaliczni) wymieniają się poglądami na temat sytuacji aktywów, tworząc związany z nimi sentyment.

Według hipotezy rynków efektywnych Famy E., część lub wszystkie powyższe informacje są natychmiast odzwierciedlone w wycenach aktywów [57]. W rzeczywistości jednak, przepływ informacji nie może być natychmiastowy, a interpretując wszystkie publicznie dostępne wiadomości szybciej niż inni, możliwe jest wykorzystanie tych dodatkowych informacji do predykcji krótkoterminowych ruchów cen. Fakty te wymagają zatem uwzględnienia, zwłaszcza w krótkoterminowych modelach predykcyjnych i decyzyjnych na rynkach finansowych.

Jeszcze do niedawna odpowiednio szybka i/lub automatyczna analiza rosnącego strumienia danych tekstowych pojawiających się codziennie w Internecie nie była możliwa. Obecnie, dzięki zaawansowanym metodom przetwarzania języka naturalnego, staje się to celem realnym do osiągnięcia, który może pomóc przewidzieć i wytłumaczyć zachowanie rynków finansowych z większą niż kiedykolwiek dokładnością. Mimo bardzo dużego postępu metod automatycznego przetwarzania tekstu w sposób całościowy, w ekonomii wciąż najbardziej popularne są jedynie te podstawowe metody – analiza sentymentu, wyszukiwanie słów kluczowych czy rozpoznawanie jednostek referencyjnych.

W niniejszej rozprawie zaproponowana została autorska metoda prognozowania finansowych szeregów czasowych z uwzględnieniem najbardziej aktualnych algorytmów wykorzystujących Przetwarznie Języka Naturalnego - ang. *Natural Language Processing* (NLP), takich jak mechanizm uwagi, technologia *word embeddings* i głębokie sieci neuronowe. Przeprowadzone zostały badania skuteczności ekstrakcji wartościowych informacji z tekstu pisanego i odpowiedniego wykorzystania ich do poprawy skuteczności predykcji. Do przeprowadzenia ekstrakcji informacji wykorzystywany jest autorski mo-

duł nazwany Enkoder Wiadomości Ekonomicznych i Finansowych - ang. *Financial and Economic News Encoder* (FENE).

Oprócz tego, przeanalizowano możliwość połączenia kilku różnych podejść do analizy tekstu oraz kilku metod prognozowania w jeden złożony model uwzględniający różne zjawiska. De facto jest to próba zautomatyzowania analizy światowych wydarzeń, mogących mieć wpływ na rynki finansowe i włączenia jej wyników do klasycznej analizy szeregów czasowych.

Głównym i nadrzędnym celem niniejszej pracy jest **zbadanie możliwości zastosowania najnowszych rozwiązań z dziedziny NLP do zautomatyzowanej analizy wiadomości oraz postów o tematyce finansowej, gospodarczej i politycznej wraz z uwzględnieniem ich w finansowych modelach predykcyjnych**, a także stworzenie narzędzi (oprogramowania), które wspomagają wyliczanie takich prognoz.

Wśród szczegółowych celów rozprawy należy wymienić:

1. Określenie źródeł danych tekstowych (kierowanych do żywych odbiorców), wywołujących najsilniejsze reakcje na rynkach finansowych. Badając szybkie i stosunkowo duże, dla danego aktywa, zmiany cen, można prześledzić, jakie źródła wiadomości są najbardziej opiniotwórcze i na których można polegać w dalszych etapach modelowania.
2. Zbadanie możliwości efektywnego wykorzystania głębokich sieci neuronowych i innych współczesnych algorytmów NLP do automatycznej analizy wiadomości i postów w mediach społecznościowych pod kątem przewidywania dynamiki notowań giełdowych. Wyjątkowa elastyczność sieci neuronowych pod względem różnorodności przyjmowanych danych wejściowych oraz możliwości ich przetwarzania, a także ogromny wybór ich architektury, dają niemal nieograniczone możliwości - przy założeniu posiadania odpowiedniej ilości danych treningowych.
3. Zaproponowanie oraz implementacja autorskich hybrydowych modeli łączących klasyczną predykcję szeregów czasowych z predykcją na podstawie automatycznej analizy publikowanych na bieżąco wiadomości i postów w mediach społecznościowych (część analizy fundamentalnej) oraz zbadanie ich efektywności. Zastosowanie połączenia odpowiedzi kilku różnych modeli przy pomocy odpowiedniej funkcji, może w efekcie dać model o wyższej skuteczności [161]. Ponieważ nie istnieje ograniczenie co do typu łączonych modeli, ulepszenie autorskiej sieci neuronowej przez połą-

czenie z klasycznymi modelami ekonometrycznymi znacznie poszerza wybór poza ateoretyczne modele uczenia maszynowego.

4. Zbadanie zależności, w jaki sposób wyceny różnych aktywów reagują na pojawiające się informacje na ich temat, a także czas trwania wywołanego zaburzenia. Pozwoli to określić, czy jeden model może opisywać grupę aktywów, czy dla każdego z nich konieczna jest osobna instancja modelu z własnym zestawem parametrów dobranych dla każdego instrumentu finansowego indywidualnie.
5. Porównanie skuteczności zaproponowanych metod z bieżącymi rozwiązaniami w literaturze.
6. Napisanie autorskich programów komputerowych w ogólnie dostępnych środowiskach programistycznych. Umożliwi to przeprowadzenie badań z wykorzystaniem modeli hybrydowych z autorskim modułem do analizy wiadomości finansowych w zakresie powyższych celów badawczych.

Sformułowano również pytania badawcze, na które odpowiedzi można udzielić na podstawie przeprowadzonych doświadczeń i analiz. Poszukiwanie odpowiedzi na poniższe pytania pomogło ukierunkować badania i osiągnąć cele rozprawy.

— **Jaka architektura sieci neuronowej lub innego algorytmu uczenia maszynowego pozwala na najdokładniejszą predykcję notowań giełdowych z wykorzystaniem technik NLP w analizie treści wiadomości?** Czy wydłużony czas obliczeń większego modelu jest uzasadniony przez odpowiednio wyższą skuteczność?

— **Czy analiza całej wiadomości wraz z komentarzami użytkowników serwisu oraz ewentualnych reakcji na nią w mediach społecznościowych pozwala na otrzymanie dokładniejszej predykcji niż wyłącznie analiza nagłówków?** Istnieją publikacje [42, 43], w których poddano analizie prognozowanie na podstawie treści nagłówków wiadomości finansowych. Jednak czy same nagłówki zawierają wszystkie informacje, które można wyekstrahować z pełnego artykułu?

— **Czy wyprzedzenie, z jakim można przewidzieć ruchy notowań giełdowych na podstawie wiadomości (finansowych, geopolitycznych, etc) i wywołanych przez nie reakcji w mediach społecznościowych, pozwala na wystarczająco szybką reakcję inwestora?** Po pewnym czasie od publikacji wiadomości, rynki osiągają ekwilibrium. Czy czas, jakiego potrzebuje algorytm do zinterpreto-

wania informacji i przewidzenia dużych wzrostów lub spadków, jest wystarczająco krótki, aby zdążyć wykonać odpowiednie operacje kupna lub sprzedaży?

— **Czy modele hybrydowe łączące klasyczną ekonometrię finansową z metodami NLP stosowanymi do analizy treści wiadomości stanowią bardziej efektywne narzędzie prognozowania finansowego?** Czy efektywność prognoz modeli hybrydowych jest większa niż prognoz otrzymanych przy użyciu podstawowych metod?

Formułując hipotezy badawcze w niniejszej dysertacji, oparto się na przesłance, iż inwestorzy podejmują decyzje na podstawie posiadanych informacji. Wiele wiadomości jest publicznie dostępnych dzięki Internetowi, a nawet te, które nie są łatwo dostępne, w końcu znajdują swoje odzwierciedlenie w wycenach – jest to część hipotezy rynku efektywnego. Jednak informacji nie wystarczy „posiadać" – należy ją odpowiednio przetworzyć, zrozumieć i na jej podstawie podjąć decyzję. Z tego powodu szybkość przetwarzania dużych wolumenów informacji, które w znacznej części podawane są w formie pisanego tekstu zrozumiałego dla człowieka, stanowić może przewagę rynkową i prowadzić do większych zysków.

Zautomatyzowanie procesu przetwarzania informacji i użycia ich do prognozowania wycen aktywów stanowi obecnie lukę badawczą, a jednocześnie istotny problem do rozważenia. Planowana praca wypełni tę lukę poprzez uzupełnienie najbardziej skutecznych modeli predykcji szeregów czasowych o modele NLP według aktualnego stanu wiedzy. Takie połączenie pozwoli modelom wykorzystywać nie tylko informacje zawarte w historycznych wycenach aktywów, ale także dostosowywać prognozę do bieżących wydarzeń na świecie.

Sformułowano następujące hipotezy badawcze:

— **Możliwe jest efektywne zautomatyzowanie analizy finansowych wiadomości oraz postów w mediach społecznościowych poprzez użycie współczesnych metod przetwarzania języka naturalnego – głównie głębokich sieci neuronowych.** Wyjątkowo szybki rozwój metod uczenia maszynowego oraz technologii komputerowych pozwolił na powstanie wyjątkowo skutecznych i wydajnych algorytmów przetwarzania języka naturalnego, jednak konieczny jest test, który określi, czy najnowsze metody NLP są w stanie interpretować wiadomości tak, jak inwestorzy.

— **Wykorzystanie treści całej wiadomości w analizie pozwala na uzyskanie lepszych (dokładniejszych) prognoz punktowych cen i stóp zwrotu wybranych aktywów finansowych niż w przypadku wykorzystania wyłącznie treści ich nagłówków.** Pełna treść wiadomości może zawierać więcej istotnych informacji przez co dokładność predykcji jest większa.

— **Modele oparte na głębokich sieciach neuronowych z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z modelami wykorzystującymi klasyczne metody sztucznej inteligencji.** Dodanie modułu FENE do sieci neuronowych przetwarzających finansowe szeregi czasowe prowadzi do poprawienia jakości prognoz.

— **Wykorzystanie szeregu czasowego notowań o 5 minutowej częstotliwości jest wystarczające, aby zaobserwować zmiany wielkości stóp zwrotu analizowanych aktywów finansowych spowodowane publikacją wiadomości.** Chwilowa nieefektywność rynku i szybka, zautomatyzowana analiza treści wiadomości pozwala na przewidzenie kierunku zmian szybciej niż poprzez obserwację poziomu cen.

— **Zautomatyzowana analiza finansowych wiadomości i postów w mediach społecznościowych za pomocą algorytmów NLP w połączeniu z obecnie stosowanymi metodami prognozowania finansowych szeregów czasowych pozwala na poprawienie efektywności predykcji.** Możliwość połączenia modeli ekonometrycznych z sieciami neuronowymi musi zostać przyjęta bądź odrzucona.

— **Klasyczne modele ekonometrii finansowej typu ARMA-GARCH z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z klasycznymi modelami typu ARMA-GARCH bez tego modułu.** Wykorzystanie danych wyjściowych modułu FENE jako wejściowych danych egzogenicznych dla klasycznych modeli ekonometrycznych prowadzi do poprawienia jakości prognoz.

— **Klasyczne modele ekonometrii finansowej typu ARMA-GARCH z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z modelami opartymi na głębokich sieciach neuronowych z autorskim modułem FENE.** Wykorzystanie modułu FENE w połączeniu z klasycznymi modelami ekonometrycznymi daje lepsze rezultaty niż w połączeniu z głębokimi sieciami neuronowymi.

Punktem wyjścia w zakresie metodycznym jest przegląd literatury z dziedzin ekonomii, ekonometrii finansowej, uczenia maszynowego oraz przetwarzania języka naturalnego. Na tej podstawie możliwe było ustalenie dalszego zakresu prac i wykorzystanie najnowszych osiągnięć wszystkich wymienionych dziedzin.

Głównym narzędziem wykorzystywanym w niniejszej rozprawie są głębokie sieci neuronowe. Sieci te zbudowane są z dwóch części: klasycznej sieci rekurencyjnej służącej do modelowania finansowych szeregów czasowych oraz połączonego z nią dużego modelu językowego ekstrahującego informacje zawarte w wiadomościach giełdowych - autorskiego modułu FENE. Jakość otrzymanych prognoz została oceniona przez porównanie ich z prognozami klasycznych modeli ekonometrycznych ARMA-GARCH.

Metody oceny prognoz punktowych w niniejszej pracy obejmowały wykorzystanie mierników opartych o błędy prognoz: średni błąd bezwzględny (MAE), średni bezwzględny błąd procentowy (MAPE) oraz pierwiastek błędu średniokwadratowego (RMSE). Oprócz tych mierników dla każdej pary porównywanych modeli przepropwadzony został statystyczny test Diebolda-Mariano.

Wszystkie operacje na danych (pobranie ich i wstępna obróbka) oraz obliczenia w niniejszej pracy zostały przeprowadzone przy użyciu języka Python wraz z bibliotekami *NumPy*, *Pandas* i *TensorFlow*. Wyjątek stanowią modele ARMA-GARCH, które były estymowane z wykorzystaniem języka R i pakietu *rugarch*.

W rozdziale pierwszym opisano ogólną charakterystykę rynków finansowych - czym są, jaka jest ich rola i ich klasyfikacja. Przedstawiono zarys instrumentów finansowych, które podlegają sprzedaży, a także opis uczestników rynku. Następnie przedstawiono źródła informacji, na których uczestnicy rynku bazują w swoich decyzjach oraz sposób ich wykorzystania, wraz z odniesieniem do hipotezy rynku efektywnego, której rzeczywista realizacja (stopień efektywności) jest wyjątkowo istotna w kontekście niniejszej pracy.

Rozdział drugi daje ogólny zarys obecnych metod wykorzystywania informacji w ekonometrii finansowej oraz szczegółowy opis metod prognozowania finansowych szeregów czasowych stosowanych w klasycznej ekonometrii, jak również ateoretycznych modeli uczenia maszynowego.

Trzeci rozdział zaczyna się od przedstawienia historii rozwoju technik NLP. Następnie opisane są zastosowania NLP oraz współczesne osiągnięcia - najlepsza osiągnięta jakość modeli i metryki, którymi została zmierzona. W kolejnym podrozdziale przedstawiono narzędzia używane do rozwoju nowych modeli i modele, które mogą

posłużyć do tzw. *transfer learningu*. Rozdział zakończony jest przeglądem literatury i podsumowaniem zastosowania NLP w przetwarzaniu wiadomości finansowych i użyciu ich do poprawy predykcji szeregów finansowych.

W czwartym rozdziale opisano metodologię badań użycia NLP do przetwarzania treści wiadomości finansowych. W pierwszym podrozdziale znajduje się opis autorskiego modelu predykcji LSTM-FENE wykorzystującego techniki NLP i uzasadnienie jego wyboru. Kolejny podrozdział omawia sposób, w jaki zostaną zweryfikowane hipotezy badawcze. Następnie przedstawiona jest ogólna architektura całego systemu prognozowania i sposobu, w jaki poszczególne elementy współpracują. W dwóch kolejnych podrozdziałach zostały przedstawione szczegóły budowy komponentu do gromadzenia danych tekstowych i ich wstępnej obróbki, a także architektura właściwego modelu do predykcji notowań giełdowych na podstawie pobranych wiadomości finansowych oraz finansowych szeregów czasowych. Rozdział zakończony jest konkluzjami z przeprowadzonych badań.

Rozdział piąty to właściwe badania nad efektywnością predykcji finansowych szeregów czasowych przy użyciu automatycznej analizy treści wiadomości z wykorzystaniem zaproponowanego modelu predykcyjnego z modułem FENE. Rozdział rozpoczyna się analizą wpływu wiadomości na zachowanie aktywów w zależności od kategorii wiadomości, jak i typu rynku, na którym notowane są te aktywa. Przedstawiony jest również wpływ ilości i długości dostępnych wiadomości na jakość predykcji. Następnie został opisany wpływ doboru hiperparametrów na efektywność modelu, a także jego stabilność. Ostatni podrozdział to podsumowanie osiągniętych wyników.

W rozdziale szóstym przedstawiono propozycje połączenia autorskiego modułu FENE ze znanymi już modelami poprzez wykorzystanie jego danych wyjściowych jako danych egzogenicznych dla klasycznych modeli ekonometrycznych. W pierwszym podrozdziale zostały zebrane propozycje modeli możliwych do połączenia, zaś w kolejnym znajduje się porównanie uzyskanych dzięki nim wyników. Rozdział kończy podsumowanie efektywności hybrydowych modeli oraz możliwości ich użycia do zbudowania rzeczywistej strategii inwestycyjnej.

Rozdział siódmy to zbiorcze podsumowanie wyników badań empirycznych uzyskanych dla zastosowanych modeli dla różnych klas aktywów finansowych, a także analiza porównawcza skuteczności zaproponowanych metod z bieżącymi rozwiązaniami w literaturze. W rozdziale zebrano również wnioski dotyczące możliwości praktycznego

zastosowania proponowanej metodologii. Dodatkowo przedstawiono propozycje kontynuacji badań i możliwe dalsze kroki.

Aneks zawiera autorskie kody źródłowe programów wykorzystanych do gromadzenia i przetwarzania danych, a także analizę otrzymanych wyników.

Rozdział 1

Informacja na rynkach finansowych oraz jej wykorzystanie w procesach prognozowania

Zachowanie uczestników rynków finansowych może być postrzegane jako proces decyzyjny. Często zakłada się, że każdy z uczestników jest racjonalnym agentem i dąży do zmaksymalizowania swojej funkcji użyteczności. Skuteczna optymalizacja tego procesu decyzyjnego wymaga jednak, by posiadać informacje o stanie swoim i otoczenia. W przypadku rynków finansowych, agenci nie są w stanie obserwować większości opisujących je zmiennych i dociera do nich tylko część informacji (często z dużym opóźnieniem) - historyczne ceny, wielkości makroekonomiczne, wiadomości o spółkach notowanych na giełdzie, ich konkurencji i wynikach finansowych.

Przy tak ograniczonej informacji trudno jest w jednoznaczny sposób przewidzieć przyszłość i konieczne jest budowanie matematycznych modeli predykcyjnych, które mogą pozwolić lepiej ocenić, czy oczekiwana stopa zwrotu z danej inwestycja uzasadnia ponoszone ryzyko. Zrozumienie charakterystyki rynków finansowych, a także źródeł informacji, które można wykorzystać do prognozowania przyszłości, jest konieczne, aby lepiej zorientować się w sytuacji.

1.1. Charakterystyka rynków finansowych

1.1.1. Pojęcie rynku

Współczesny świat, z punktu widzenia ekonomii, najczęściej zorganizowany jest jako gospodarka rynkowa. Rynek to ogół celowych transakcji kupna-sprzedaży oraz warunków, na jakich zostały one zrealizowane [34]. Jednym z najważniejszych warunków ustalanych przez obie strony jest cena. Kupujący (zwykle) chcą kupić jak najtaniej, zaś sprzedający sprzedać jak najdrożej. Każda transakcja zawierana jest w sposób dobrowolny, zatem w przypadku, gdy stronom nie uda się wspólnie ustalić ceny (oraz innych warunków), może do niej w ogóle nie dojść. Stronom zgłaszającym popyt i tym

zgłaszającym podaż zależy na dokonaniu transakcji, zatem kilkukrotne niezawarcie transakcji może skutkować obniżeniem oczekiwań i w zależności od konkurencji na rynku, przesunięciu swoich oczekiwań co do ceny (obniżeniu przez sprzedających lub podniesieniu przez kupujących). Podaż i popyt większości dóbr i usług nie jest stała w czasie, a zatem cena również podlega fluktuacjom.

Gospodarka jest to zbiór wielu pojedynczych rynków, które z punktu widzenia przedmiotu wymiany można kategoryzować w następujący sposób [56]:

— rynek produktów,
— rynek nieruchomości,
— rynek pracy,
— rynek finansowy.

Rynki produktów, nieruchomości i pracy charakteryzują się przepływem dóbr i pieniądza w przeciwnych kierunkach. Jest to naturalny stan rzeczy, w którym dobra mające faktyczną użyteczność i wartość wymieniane są na pieniądz, który właściwie jest jedynie miernikiem wartości dóbr, niemającym wartości samej w sobie. Jednak w rozwiniętych gospodarkach można zaobserwować, że przepływ pieniądza uniezależnia się od przepływu dóbr.

Rynek finansowy to ogół zjawisk dotyczących transakcji w sferze czysto pieniężnej. Mimo że nie istnieje na nim przepływ dóbr, rynek ten zachowuje się podobnie do pozostałych, ponieważ rządzi nim uniwersalne prawo popytu i podaży. Cechy specyficzne tego rynku to większa podatność na manipulacje, a ryzyko jest znacznie większe. Przedmiotem obrotu jest de facto siła nabywcza, a jej ceną (w dużym uproszczeniu) stopa procentowa.

W celu sformalizowania sprzedaży siły nabywczej, konieczne jest wprowadzenie jakiegoś dokumentu poświadczającego zawarcie transakcji - obietnica oddania pieniędzy w terminie i uścisk dłoni nie zawsze są wystarczające, szczególnie, kiedy obrót jest na większą skalę, a strony nie znają się osobiście. Przedmiotem obrotu są więc instrumenty finansowe, czyli zobowiązania finansowe podmiotów, które pożyczają pieniądze, względem podmiotów posiadających oszczędności i chętnych je pożyczyć na pewien okres. Instrumentami finansowymi są:

— depozyty bankowe,
— papiery wartościowe,
— polisy ubezpieczeniowe,

— jednostki uczestnictwa w funduszach inwestycyjnych,

— opcje.

Najczęściej używane są papiery wartościowe. Ich zaletą jest pokrycie w aktywach podmiotu, który je wyemitował. W przeważającej części przypadków osoba kupująca papiery wartościowe ma prawo do dywidendy z tytułu ich posiadania [167].

W każdym momencie pewne podmioty posiadają oszczędności, zaś inne potrzebują dodatkowych środków. Cały mechanizm funkcjonowania rynku finansowego polega na tym, że podmioty posiadające nadwyżki finansowe godzą się czasowo udostępnić wolne środki za określoną opłatą. Cena wynika z konieczności powstrzymania się od konsumpcji na czas udzielenia pożyczki, a także ryzyka, że nie zostanie ona spłacona (w części lub całości).

Podmiotami posiadającymi nadwyżki finansowe mogą być: państwo, podmioty zagraniczne, przedsiębiorstwa, gospodarstwa domowe. Jednocześnie te same typy podmiotów mogą potrzebować dodatkowych środków. Oprócz tego, trzecim typem uczestników rynków finansowych są wszelkiego rodzaju pośrednicy, ułatwiający obrót instrumentami finansowymi. W większości przypadków, bezpośrednie transakcje są możliwe pomiędzy pożyczkodawcami i pożyczkobiorcami, jednak we współczesnych gospodarkach istnieje duża liczba instytucji finansowych ułatwiających zawieranie transakcji. Pośrednicy oferują różnego rodzaju usługi, przy czym najbardziej istotne jest zorganizowanie i ustrukturyzowanie przepływu wolnego kapitału, a także zapewnianie bezpieczeństwa stron transakcji (w tym zapobieganie praniu pieniędzy, unikaniu opodatkowania, itp.) [47].

W ogólności udostępnienie nadwyżek finansowych podmiotom, które ich potrzebują, może odbyć się na trzy sposoby:

1. Bezpośrednio - podmioty posiadające nadwyżki kontaktują się bezpośrednio z podmiotami potrzebującymi dodatkowych środków i przekazują je w zamian za odpowiednie papiery wartościowe,
2. Przez zorganizowane rynki finansowe - podmioty również dokonują transakcji w sposób bezpośredni, jednak dzięki zorganizowanemu rynkowi znacznie ułatwiony jest kontakt stron,
3. Przez pośredników finansowych - w tym przypadku podmioty nadwyżkowe przekazują swoje oszczędności pośrednikowi (np. funduszowi inwestycyjnemu), który emituje własne zobowiązania finansowe (np. jednostki uczestnictwa w funduszu), a

następnie udostępnia podmiotom deficytowym uzyskany kapitał w zamian za ich zobowiązania finansowe (papiery wartościowe).

Należy dodać, że pośrednicy również mogą być uczestnikami zorganizowanego rynku - emitując i sprzedając swoje zobowiązania finansowe, a także skupując zobowiązania innych podmiotów.

1.1.2. Rola rynków finansowych

Rynki finansowe powstały, by przede wszystkim umożliwić uniezależnienie konsumpcji od posiadanego w danym momencie kapitału. Bez rynków finansowych podmioty, które chcą się rozwijać, ale nie posiadają środków, byłby zmuszone do oczekiwania na zgromadzenie ich, a w ekstremalnych przypadkach do upadku. Podmioty, które posiadają w danej chwili nadwyżki, a nie mają możliwości lub woli prowadzenia jakiegoś typu działalności, wciąż mogą otrzymywać dochód udostępniając swoje środki. Dzięki temu uczestnicy rynku mogą rozplanować w czasie sposób zarządzania swoimi aktywami finansowymi. Nic nie stoi na przeszkodzie, aby w jednym okresie być na rynku podmiotem nadwyżkowym, a w innym deficytowym [34].

Udostępnianie środków finansowych w ten sposób prowadzi do zwiększenia efektywności wykorzystania zasobów w całej gospodarce [47]. Gdyby rynki finansowe nie istniały, wszelkie podmioty posiadające nadwyżki byłyby zmuszone do gromadzenia i przetrzymywania ich w postaci gotówki lub aktywów rzeczowych. W ten sposób kapitał danego podmiotu mógłby jedynie oczekiwać na moment, kiedy będzie znowu potrzebny. Dzięki rynkom finansowym możliwe jest wykorzystanie tego kapitału czasowo przez kogoś innego.

Trzecią funkcją rynków finansów jest umożliwienie dywersyfikacji, a dzięki temu ograniczanie ryzyka. Jako przykład można podać małego przedsiębiorcę, który kieruje firmą zajmująca się jakąś branżą. Prowadzenie firmy jest kosztowne czasowo i zwykle konieczne jest skupienie się na jednym lub co najwyżej kilku obszarach. W przypadku kryzysu w danym sektorze gospodarki, właściciel spółki jest narażony na duże straty, jednak poprzez udostępnienie swoich oszczędności przedsiębiorstwom działającym w innych branżach, może mieć nadzieję na zabezpieczenie swoich zysków, otrzymując dywidendy z biznesów mniej dotkniętych kryzysem.

Różnego typu aktywa finansowe charakteryzują się różnym poziomem ryzyka, dlatego budując odpowiedni portfel, możliwe jest dopasowanie całkowitego ryzyka do oso-

bistych preferencji każdego z podmiotów. W ten sposób rynki finansowe pozwalają nie tylko zmniejszać ryzyko, ale (w ograniczonym zakresie) regulować je w górę i w dół.

1.1.3. Państwo a rynki finansowe

Bardzo ważnym uczestnikiem rynku finansowego jest państwo. Spełnia ono trzy role: reguluje i kontroluje procesy zachodzące na rynku finansowym, wywiera bezpośredni wpływ na ogólną gospodarkę, zawiera zwyczajne transakcje na tym rynku. Rynki finansowe są przedmiotem regulacji ustawowej - sektor bankowy, ubezpieczeniowy i publiczny rynek papierów wartościowych - są ściśle kontrolowane przez państwo, ale także przez przepisy międzynarodowe. Dodatkowo istnieją instytucje nadzorcze (np. komisje papierów wartościowych, nadzór ubezpieczeniowy, nadzór bankowy), które stale kontrolują przestrzeganie prawa, a także wydają odpowiednie uprawnienia dla podmiotów zainteresowanych prowadzeniem działalności w sektorze finansowym [165].

Państwa mogą wpływać na sytuację gospodarczą poprzez politykę pieniężną i politykę fiskalną. Stosując politykę pieniężną bank centralny może wykorzystywać rynki finansowe i wpływać na sytuację, która na nich panuje, np. przez kontrolę podaży pieniądza i stóp procentowych. Wydarzenia na rynkach finansowych mają bezpośredni wpływ na gospodarkę realną.

Państwa są również zwykłymi uczestnikami rynków finansowych. W większości przypadków występują jako podmioty deficytowe - rządy zwykle nie posiadają wystarczających funduszy na planowane wydatki i są zmuszone do pożyczania pieniędzy na pokrycie deficytu budżetowego. Wysokość skumulowanego zadłużenia państw może sięgać powyżej 100% PKB, dlatego stanowi znaczną część popytu na pieniądz na danym rynku finansowym. Z tego powodu państwa mogą w znacznym stopniu wpływać na poziom stóp procentowych, a także dostępność kapitału dla innych podmiotów [34].

1.1.4. Kategoryzacja rynków finansowych

Rynki finansowe można kategoryzować na wiele sposobów. Jednym z podziałów jest klasyfikacja na rynek pierwotny i wtórny. Na rynku pierwotnym dochodzi do transakcji między emitentem papierów wartościowych a inwestorami, którzy są pierwszymi właścicielami danych aktywów (często w tym procesie uczestniczy również pośrednik). Podczas takiej transakcji na rynek trafiają nowe (dodatkowe) aktywa. Sprzedaż odbywa się po uzgodnionej wcześniej między stronami cenie emisyjnej. W przypadku, gdy każdy może nabyć emitowane papiery wartościowe, jest to oferta publiczna. Możliwa jest

również sprzedaż do konkretnej grupy inwestorów poprzez ofertę prywatną. W każdym z przypadków, oferta jest ograniczona czasowo, a także ilościowo - może się zdarzyć, że zlecenia kupna będą zredukowane przez duży popyt [47].

Papiery wartościowe mogą być przedmiotem obrotu między inwestorami na rynku wtórnym. W tym przypadku cena kształtowana jest wyłącznie przez popyt i podaż. Inwestorzy mogą tworzyć zlecenia kupna i sprzedaży podając zadowalająca ich cenę i czekając aż znajdzie się przeciwna strona transakcji, która jest skłonna ją zaakceptować. Na rynku wtórnym jest więc możliwe osiąganie korzyści jedynie przez wzrost wyceny papierów wartościowych. W przypadku tego obrotu, pieniądze nie trafiają do emitenta i nie ma on bezpośredniego wpływu na ceny wyemitowanych w przeszłości papierów wartościowych. Obrót rynku wtórnego najczęściej odbywa się na giełdzie papierów wartościowych. Istnieje również rynek pozagiełdowy, na którym możliwa jest sprzedaż papierów nie dopuszczonych do obrotu publicznego (np. małych spółek) [41].

Klasyfikacja na rynek publiczny i prywatny wynika z dostępności aktywów dla inwestorów. Na rynku publicznym inwestować może w zasadzie każdy. Ze względu na dobro inwestorów, rynek ten jest mocno kontrolowany i regulowany przepisami. Nie każda oferta może być na nim wystawiona. Jednym z ograniczeń może być wielkość spółek, które mogą emitować swoje papiery wartościowe. Najczęściej rynek publiczny ma formę giełdy papierów wartościowych. Rynek prywatny znacznie częściej dotyczy indywidualnych transakcji pomiędzy bankami, małymi firmami (start-upami), a także przekazywania walorów miedzy spółkami lub też łączenia ich [34, 167].

Kolejnym podziałem jest klasyfikacja na rynek transakcji natychmiastowych (rynek kasowy) i rynek terminowy. W przypadku rynku kasowego, efektem transakcji jest natychmiastowe przekazanie kapitału. Na rynku terminowym, przedmiotem obrotu jest zobowiązanie do przyszłej zapłaty w określonym przez strony terminie, czyli instrumenty pochodne.

Jednak najbardziej podstawową kategoryzacją rynków finansowych, jest podział na rodzaj instrumentów finansowych, które są na nich wymieniane:

— rynek pieniężny,
— rynek kapitałowy,
— rynek walutowy,
— rynek kryptowalut,
— rynek instrumentów pochodnych.

Ze względu na specyfikę kryptowalut, ich rynek zostanie opisany oddzielnie od rynku walutowego.

Rynek pieniężny to miejsce obrotu krótkookresowymi instrumentami - ich termin wykupu wynosi do 1 roku. Dzięki temu rynkowi możliwe jest kontrolowanie płynności przez podmioty. Przekazanie pieniędzy odbywa się przez udzielenie kredytu lub sprzedaż krótkoterminowych obligacji. Kredytu podmiotom deficytowym (instytucjom finansowym, organom państwowym, spółkom, a także osobom fizycznym) najczęściej udzielają banki. Drugą metodą jest emisja krótkoterminowych papierów wartościowych - państwa emitują weksle skarbowe, przedsiębiorstwa weksle komercyjne, a banki certyfikaty depozytowe. Mimo że z założenia są to wierzytelności krótkoterminowe, istnieje możliwość długoterminowego przekazania pieniądza poprzez rolowanie - podmioty deficytowe emitują kolejne serie papierów wartościowych i z uzyskanych pieniędzy wykupują poprzednie emisje, inwestorzy zaś nie wycofują swoich pieniędzy po umorzeniu posiadanej emisji, lecz lokują je w kolejnej. Rynek pieniężny jest ważnym narzędziem polityki pieniężnej. Dzięki niemu bank centralny może kupować i sprzedawać krótkoterminowe papiery wartościowe i w ten sposób zwiększać lub zmniejszać rezerwy systemu bankowego. [171]

Na rynku kapitałowym dochodzi do transakcji kupna i sprzedaży instrumentów średnio i długoterminowych. Jego zasada działania jest bardzo podobna do rynku pieniężnego, jednak jest silniej regulowany ze względu na dłuższy okres lokowania kapitału, a przez to większe ryzyko, jakie ponosi inwestor. Ogólna gospodarka zyskuje na tym, ponieważ rozproszone wolne środki inwestorów mogą być przekazane do podmiotów deficytowych w momentach, kiedy ich potrzebują i zasilają ich fundusze inwestycyjne. Dla pojedynczego podmiotu deficytowego rynek kapitałów jest zewnętrznym źródłem finansowania działalności i inwestycji. Z punktu widzenia inwestora rynek kapitałowy pozwala ulokować czasowo wolne oszczędności [34].

Rynek walutowy to rynek, na którym można wymienić walutę jednego kraju na walutę innego. Celem tego rynku jest umożliwienie przepływu pieniądza pomiędzy gospodarkami państw, które bez tego mechanizmu byłyby zamknięte lub bardzo ograniczone. Największą częścią tego rynku jest sektor międzybankowy. Banki centralne mogą kupować i sprzedawać inne waluty nie wpływając na poziom ich ceny (płynny kurs walutowy) lub manipulować cenami, aby osiągnąć odpowiedni ich poziom (kurs sztywny). Kurs walut jest bardzo istotny dla inwestycji zagranicznych, a także importu

i eksportu za granicę, ponieważ wpływa na ich opłacalność i stanowi dodatkowe ryzyko dla podmiotów prowadzących działalność poza granicami swojego kraju [47].

Rynek kryptowalut powstał i rozwinął się w ciągu ostatnich kilku lat w odpowiedzi na malejące zaufanie do rządów i jednocześnie zaawansowany rozwój technologii internetowych. W przeciwieństwie do tradycyjnych walut, kryptowaluty nie są emitowane przez żadne państwo (państwo nie gwarantuje wartości kryptowalut), lecz polegają (przynajmniej w teorii) na zdecentralizowanej technologii *blockchain*. Pierwszą rozpowszechnioną kryptowalutą był Bitcoin. Bezpieczeństwo transakcji zapewniają techniki kryptograficzne, które uniemożliwiają fałszowanie transakcji lub też kradzież zasobów. Dodatkową zaletą miała być z góry określona podaż pieniądza, a przez to regulowana inflacja. Z czasem powstała olbrzymia ilość różnych kryptowalut oraz giełdy kryptowalut umożliwiające ich wymianę. Ze względu na dalszy wzrost popularności, kursy poszczególnych kryptowalut były coraz mniej stałe i podlegały częstszym spekulacjom. Obecnie, mimo ponad dekady od powstania Bitcoina, używanie kryptowalut do oficjalnych rozliczeń wciąż nie jest powszechne (choć często używane w szarej strefie), rynki kryptowalut podlegają minimalnej regulacji, a ich zmienność jest tak duża, że jedynym zastosowaniem dla większości użytkowników kryptowalut są krótkoterminowe spekulacje [49, 58].

Rynek instrumentów pochodnych pozwala na obrót instrumentami opartymi o instrumenty bazowe. Instrumentami bazowymi mogą być aktywa finansowe z rynku natychmiastowego. W przypadku, gdy na rynku natychmiastowym jest duża zmienność cen, inwestorzy mogą chcieć zabezpieczyć się na przyszłość i uzgodnić cenę zakupu wcześniej. Umożliwiają to instrumenty pochodne - opcje, kontrakty forward i futures, swapy. Drugim czynnikiem uzasadniającym użycie instrumentów pochodnych jest zabezpieczenie się przed zmianami kursowymi w przypadku handlu międzynarodowego. Instrumenty pochodne mogą służyć do zabezpieczania się na wypadek niekorzystnych zmian cen, ale również do spekulacji (zwłaszcza przy użyciu dźwigni finansowej) [56].

1.1.5. Instrumenty finansowe

Instrumenty finansowe to kontrakty pomiędzy dwiema stronami, które ustalają warunki rozliczeń finansowych między nimi. Można je podzielić na papiery wartościowe, waluty (oraz kryptowaluty) i instrumenty pochodne. Najbardziej powszechnymi papierami wartościowymi są papiery dłużne [56]:

— obligacje,
— weksle,
— czeki,
— bony skarbowe,
— certyfikaty depozytowe,
— papiery udziałowe w spółkach, czyli akcje.

Niniejsza praca skupia się głównie na prognozowaniu cen instrumentów bazowych publicznego rynku wtórnego. W większości przypadków będą to akcje spółek i kursy kryptowalut, ponieważ strumień informacji dotyczących przedsiębiorstw i kryptowalut jest najbardziej dostępny dla inwestorów, a dzięki temu model może go najdokładniej odwzorować. Z tego powodu wyłącznie te dwie klasy aktywów zostaną opisane.

Akcje to papiery wartościowe przynoszące zmienny dochód. Każda akcja to dokument poświadczający, że jej właściciel ma udział w części majątku spółki oraz dodatkowe prawa - do udziału w zyskach, do udziału w zarządzaniu spółką, do podziału majątku w przypadku likwidacji oraz prawo poboru. Udział w zyskach odbywa się poprzez wypłacanie akcjonariuszom dywidend, najczęściej raz w roku. Decyzja o wielkości dywidendy (lub jej braku) jest podejmowana przez walne zgromadzenie akcjonariuszy. Dzięki prawu do zarządzania spółką, akcjonariusze mogą uczestniczyć w walnych zgromadzeniach, a waga głosu każdego z nich jest proporcjonalna do ilości posiadanych akcji. W przypadku likwidacji spółki kwota uzyskana ze sprzedaży majątku (po pokryciu zobowiązań spółki) jest rozdzielana między akcjonariuszy proporcjonalnie do ilości posiadanych akcji. Dzięki prawu poboru, akcjonariusze mają prawo pierwokupu akcji podczas kolejnych emisji [47].

Akcje nie są formą pożyczki. Akcjonariusze po określonym czasie nie odzyskują zainwestowanych pieniędzy od spółki, mogą natomiast akcje sprzedać. Podczas emisji akcje mają wartość nominalną, wynikającą z danych księgowych spółki,a także cenę emisyjną, która musi być od niej nie mniejsza. Kiedy akcje z danej emisji zostaną sprzedane, mogą trafić na rynek wtórny, a ich obrót odbywa się po stale zmieniającej się cenie rynkowej, wynikającej między innymi z sytuacji spółki, ale również warunków panujących w całej gospodarce, sentymentu inwestorów lub sytuacji konkurencyjnych firm. Akcje pozwalają spółkom na zgromadzenie kapitału potrzebnego do działania i rozwoju, a inwestorom uzyskiwanie dochodu od posiadanego kapitału bez konieczności osobistego prowadzenia działalności [34].

Kryptowaluty to cyfrowe aktywa wykorzystujące zaawansowane metody kryptograficzne do przechowywania informacji o posiadanych funduszach i transakcjach w sposób zdecentralizowany. Kryptowaluty powstały jako sposób płatności, który nie może być kontrolowany przez państwo. Z tego też powodu większość państw nie uznaje kryptowalut jako środka płatności, a wiele giełd kryptowalut wciąż znajduje się w szarej strefie. Ze względu na szybki rozwój tego rodzaju instrumentów finansowych w ciągu ostatniej dekady, ilość dostępnych kryptowalut wzrosła do kilku tysięcy. Kryptowaluty powszechnie wykorzystują Internet i nowoczesne technologie, dlatego obrót nimi jest wyjątkowo szybki i łatwy, co częściowo wpływa na ich ogromną zmienność (transakcje można wykonywać z dużą szybkością w porównaniu do klasycznych aktywów). Mimo ponad dziecięciu lat od ich powstania, nie przyjęły się jako powszechny sposób płatności i obecnie stanowią aktywa, które nie posiadają wartości wewnętrznej, a ich cena wynika prawie wyłącznie ze spekulacji. Dodatkowym ryzykiem podczas inwestowania w kryptowaluty jest ich sytuacja prawna - w niektórych państwach (Algieria, Bangladesz, Chiny, Irak, Egipt, Katar, Maroko, Oman, Tunezja) są całkowicie nielegalne, a w innych krajach sytuacja może zmieniać się bardzo dynamicznie.

1.1.6. Uczestnicy rynku

Uczestników rynków finansowych można podzielić na pięć kategorii:

— emitenci,
— inwestorzy,
— pośrednicy,
— instytucje regulujące rynek,
— instytucje organizujące rynek.

Pewne podmioty mogą znajdować się jednocześnie w kilku kategoriach lub zmieniać swoje przypisanie do nich w czasie. Dwie najistotniejsze grupy to emitenci i inwestorzy, ponieważ to właśnie oni uzasadniają istnienie rynków finansowych. W kontekście niniejszej pracy najbardziej istotnymi uczestnikami rynku są inwestorzy oraz pośrednicy, ponieważ te dwie grupy poprzez transakcje kupna i sprzedaży wpływają na cenę rynkową. Ważni są również emitenci, którzy poprzez sposób zarządzania przedsiębiorstwem i podejmowane działania dostarczają inwestorom i pośrednikom informacji, na podstawie których podejmują decyzje inwestycyjne [34].

Emitenci to w ogólności podmioty poszukujące dodatkowych środków. Część z nich nie emituje papierów wartościowych samodzielnie - na przykład gospodarstwa domowe w celu pozyskania środków najczęściej zaciągają kredyty w bankach, które z kolei już bezpośrednio emitują swoje papiery wartościowe. Różnego rodzaju przedsiębiorstwa finansowe emitują krótko-, średnio- i długoterminowe papiery wartościowe. Ich cena bardzo mocno zależy od czynników makroekonomicznych i ogólnej sytuacji gospodarczej.

Rządy i samorządy emitują papiery wartościowe, których cena również najmocniej zależy od czynników gospodarczych i makroekonomicznych.

Przedsiębiorstwa mogą emitować akcje i obligacje, w zależności od tego, w którym przypadku pozyskanie kapitału będzie tańsze. W momencie emisji akcji ich cena jest ustalana w sposób bezpośredni przez przedsiębiorstwo, jednak w chwili, gdy trafiają na rynek wtórny, cena zależy od popytu i podaży. Sposób, w jaki przedsiębiorstwo jest zarządzane, działania konkurencji, ogólna sytuacja w branży i całej gospodarce, publikacja wyników finansowych, a nawet skandale związane z osobami powiązanymi - wszystko to ma wpływ, w jaki sposób inwestorzy postrzegają spółkę i jej perspektywy, a tym samym czy cena wzrośnie, czy spadnie. Z tego powodu zmiany cen są bardziej dynamiczne niż w przypadku podmiotów państwowych i przedsiębiorstw finansowych [47].

Podmioty zagraniczne również mogą być rozpatrywane jako dodatkowa kategoria emitentów, która decydując się na emisję swoich papierów wartościowych na obcym rynku, musi uwzględniać ryzyko walutowe. Tym samym cena ich papierów wartościowych dodatkowo zależy od kursu wymiany.

Inwestorzy to podmioty, które posiadają nadwyżki finansowe i chcą na określony termin pożyczyć je (zainwestować) podmiotom deficytowym w celu osiągnięcia dochodu. Najczęściej inwestorami są gospodarstwa domowe, ponieważ w gospodarkach kapitalistycznych ostatecznie większość aktywów jest prywatną własnością (lub państwową i samorządową, jednak te podmioty występują najczęściej jako emitenci). Wybierając inwestycję, gospodarstwa domowe mogą zdecydować się na samodzielne zarządzanie inwestycją lub skorzystać z usług pośredników finansowych.

Przedsiębiorstwa niefinansowe inwestują środki we własny rozwój (czasami mogą inwestować w inne spółki jako forma *hedgingu*). Natomiast instytucje finansowe stanowią największą część inwestycji (jeśli chodzi o ilość transakcji i obracany kapitał). Inwestowane przez nich środki w rzeczywistości najczęściej należą do gospodarstw domowych

(lub przedsiębiorstw niefinansowych), a sposób inwestowania podlega regulacjom prawnym.

W przypadku inwestorów również można wyróżnić osobną kategorię, jaką są podmioty zagraniczne - mogą to być gospodarstwa domowe oraz przedsiębiorstwa (finansowe i niefinansowe). Zwykle podmioty z krajów wysoko rozwiniętych i posiadające większe nadwyżki finansowe, decydują się na inwestycje na rynkach mniej rozwiniętych, oczekując szybszych wzrostów, choć okupionych większym ryzykiem.

Pośrednicy finansowi w najprostszym ujęciu mogą byś traktowani jako jednocześnie emitenci i inwestorzy. Mimo to pełnią ważne funkcje na rynkach finansowych. Ułatwiają emitentom i inwestorom przepływ środków, a także zarządzają inwestycjami podmiotów, które nie muszą dzięki temu posiadać specjalistycznej wiedzy finansowej. Dodatkowo, inwestorzy posiadają nadwyżki finansowe o charakterze zwykle krótkoterminowym lub mogą wycofać się z inwestycji w nieokreślonym z góry momencie, zaś emitenci najczęściej mają potrzeby długoterminowe. Pośrednicy finansowi stanowią swojego rodzaju bufor - udzielają pożyczek długoterminowych, a sami dokonują emisji krótkoterminowych [165].

Instytucje regulujące i nadzorujące rynek dbają o to, aby wszelkie podmioty obecne na rynkach finansowych przestrzegały odpowiednich przepisów. Ma to na celu między innymi zwalczanie monopolu, prania pieniędzy, oszustw podatkowych czy wpływu asymetrii informacji między stronami transakcji, czyli w skrócie spójności i stabilności funkcjonowania rynków finansowych i zabezpieczenie podmiotów (głównie inwestorów) przed niewypłacalnością innych (głównie emitentów). Utrzymaniem odpowiednich instytucji zajmuje się państwo, więc koszt tak naprawdę ponoszony jest przez wszystkich uczestników rynku w postaci podatków. Zmiany regulacji, a zwłaszcza zaostrzanie ich, mogą oddziaływać na rynek i wywoływać zmiany popytu i podaży, a tym samym cen [34].

Instytucje organizujące rynek, czyli instytucje, które nie zajmują pozycji na rynkach finansowych, ale ułatwiają innym podmiotom zawieranie transakcji to głównie giełdy papierów wartościowych i domy maklerskie. Giełda papierów wartościowych to miejsce, obecnie często w formie cyfrowej, gdzie w określonym czasie strony mogą zawierać transakcje, a ceny zawartych transakcji są publiczne. Domy maklerskie to instytucje, które umożliwiają swoim klientom założenie rachunku inwestycyjnego i przechowywanie na nim informacji o zdeponowanych środkach oraz posiadanych papierach wartościowych.

Operacje kupna i sprzedaży dokonywane przez inwestorów, wydarzenia gospodarcze i polityczne, akcje podejmowane przez zarządy spółek notowanych na giełdzie, a także wprowadzane regulacje mają wpływ na ceny aktywów na rynkach finansowych. Jednak aby podjąć decyzję o kupnie lub sprzedaży, konieczna jest informacja czy w danej chwili podmiotowi opłaca się takie działanie. Ze względu na różnice w dostępie do tych informacji różne podmioty mogą podejmować w tej samej chwili różne decyzje. Zarządy spółek mają bezpośredni wgląd w sytuację swojej spółki, instytucje finansowe zatrudniające profesjonalnych analityków mogą wiedzieć nieco mniej, a do detalicznego inwestora docierają już tylko częściowe informacje. W obecnych czasach dzięki powszechności Internetu, propagacja informacji jest szybsza niż kiedykolwiek, jednak coraz większy strumień wiadomości może utrudniać ich interpretację, zwłaszcza dla mniejszych inwestorów. Od szybkości **propagacji i interpretacji** informacji giełdowych zależy efektywność rynku [57].

1.1.7. Hipoteza rynku efektywnego

Za twórcę hipotezy rynku efektywnego uważa się Eugena Famę, który opisał ją w swojej publikacji z 1970 roku [57]. Empiryczne potwierdzenie prawdziwości hipotezy wymaga przyjęcia modelu rynku kapitałowego. Rynek efektywny informacyjnie to taki, w którym ceny wszystkich aktywów w każdej chwili w pełni i natychmiast poprawnie uwzględniają wszystkie informacje, które mogą mieć na nie wpływ.

Zweryfikowanie czy informacje zostały **poprawnie** uwzględnione, wymaga użycia konkretnego modelu wyceny aktywów, np. model wyceny aktywów kapitałowych - ang. *Capital Asset Pricing Model* (CAPM) lub teoria arbitrażu cenowego - ang. *Arbitrage Pricing Theory* (APT). Również sformułowanie "wszystkie informacje, które mogą mieć wpływ na wycenę" jest niejasne ze względu na niedookreślenie, jakie konkretnie to informacje i skąd pochodzą. W celu ułatwienia empirycznych badań, wyróżnia się trzy poziomy efektywności rynku:

— **Słaba efektywność rynku** występuje, gdy ceny aktywów w pełni i natychmiast uwzględniają informacje zawarte w ich historycznych cenach. W takim przypadku nie jest możliwe zbudowanie strategii inwestycyjnej, która opiera się wyłącznie na historycznych cenach i osiąga zyski większe niż wynikają z ryzyka portfela, tzn. nie da się skutecznie stosować analizy technicznej.

— **Półsilna efektywność rynku** to sytuacja, gdy ceny instrumentów na rynku w pełni i natychmiast uwzględniają wszystkie informacje dostępne publicznie, np. raporty o przychodach spółek, sprawozdania z przepływów środków pieniężnych, czy czynniki makroekonomiczne. Półsilna efektywność rynku implikuje, że nie jest możliwe osiąganie ponadnormalnych zysków wykorzystując analizę fundamentalną.

— **Silna efektywność rynku** to hipotetyczna sytuacja, w której ceny aktywów natychmiast uwzględniają informacje dostępne publicznie i prywatnie. W takim przypadku koszt pozyskania informacji dostępnych jedynie prywatnie musiałby być zerowy. W przeciwnym przypadku, dodatkowe koszty nie uzasadniałyby użycia dodatkowych informacji. Silna efektywność rynku została na tej podstawie mocno skrytykowana w publikacji Grossman S. i Stiglitz J. [75].

W ogólności hipotezę rynku efektywnego można przedstawić w postaci wzoru 1.1.

$$\mathbb{E}(\tilde{p}_{j,t+1}|\Phi_t) = [1 + \mathbb{E}(\tilde{R}_{j,t+1}|\Phi_t)]p_{j,t}, \tag{1.1}$$

gdzie $\mathbb{E}$ jest operatorem wartości oczekiwanej, $p_{j,t}$ ceną aktywa j w chwili t, $\tilde{p}_{j,t+1}$ ceną aktywa j w chwili $t+1$, $\tilde{R}_{j,t+1}$ jednookresową stopą zwrotu aktywa j w chwili $t+1$, a Φ_t zbiorem wszystkich (zależnie od tego, czy przyjęto słabą, półsilną, czy silną wersję hipotezy) dostępnych informacji. Symbol tyldy oznacza, że $p_{j,t+1}$ i $R_{j,t+1}$ są zmiennymi losowymi w chwili t.

E. Fama zebrał wiele dowodów empirycznych na potwierdzenie hipotezy rynku efektywnego, szczególnie w jej słabej, ale również półsilnej wersji. W każdym jednak przypadku konieczne są dodatkowe założenia. Wiele z badań empirycznych używało danych o niskiej czasowej rozdzielczości, głównie dziennej. Rynki natomiast są efektywne dopiero w stanie ekwilibrium. Oznacza to, że docierające do inwestorów informacje nie są rzeczywiście "natychmiast" odzwierciedlane w cenach aktywów. Informacje powstają i są propagowane przy użyciu różnych kanałów. Samo przeprowadzenie operacji kupna lub sprzedaży również wymaga pewnego czasu. Część inwestorów znajduje się w różnych strefach czasowych i rynki, na których działają, mogą nie być otwarte w momencie powstania informacji lub (głównie w przypadku inwestorów detalicznych, których odsetek jest największy w historii) mogą nie monitorować sytuacji na bieżąco. Ostatecznie jednym z wniosków ze Szczególnej Teorii Względności Einsteina A. [48], szybkość propagacji informacji (w tym ekonomicznych i finansowych) jest ograniczona przez prędkość światła w próżni c. W związku z powyższymi ograniczeniami, informa-

cja w rzeczywistym świecie nie może w sposób natychmiastowy zostać uwzględniona w wycenach.

Sposób w jaki informacje mogą być interpretowane zależy również od modelu, jakiego używa podmiot. Te same wiadomości mogą dawać zupełnie różne wnioski na temat akcji, które należy podjąć, aby maksymalizować zysk. Inwestorzy mają różne horyzonty czasowe i różne możliwości dokonywania transakcji - w zależności od wielkości prowizji, opłat i podatków, podjęcie akcji może nie być równie opłacalne dla wszystkich.

Także założenie, że wszyscy inwestorzy zachowują się racjonalnie nie jest możliwe do zrealizowania. W wielu przypadkach decyzje podejmowane są nie tyle na podstawie twardych faktów, co emocji - szczególnie lęku przed utratą kapitału. Powoduje to, że posiadając te same informacje np. o silnych spadkach cen posiadanego aktywa, część inwestorów postanowi wyjść z pozycji, a część pozostanie.

Podsumowując, istnieją silne przesłanki, by przypuszczać, że hipoteza efektywności rynkowej nie jest w mocy przynajmniej w krótkich okresach, kiedy jeszcze nie wszyscy uczestnicy rynku zdążyli zareagować na nowe informacje. Niniejsza praca skupia się na prognozowaniu cen aktywów właśnie w tych momentach, kiedy rynek nie osiągnął jeszcze stanu ekwilibrium.

1.2. Źródła informacji rynkowej

Kolejne pytanie, które nasuwa się podczas analizy wykorzystania informacji na rynkach finansowych, to jakie są jej źródła? Inwestorzy mogą korzystać z różnych mediów, od czego mocno uzależniona jest szybkość przekazu. Do najbardziej powszechnych środków przekazu można zaliczyć:

— **Przekaz ustny** - przed powstaniem Internetu, najszybszą formą przekazania informacji o korzystnych inwestycjach było opowiedzenie o nich osobiście drugiej osobie. Oprócz szybkości przekazu, zastosowanie się do czyichś rad zdecydowanie zmniejsza koszt pozyskiwania informacji. Ivković Z. opisał proces, w którym informacje o inwestycjach są propagowane w małych społecznościach [96].

— **Tekst drukowany** - wiadomości na temat spółek od dziesięcioleci są publikowane w branżowych czasopismach, a bardziej głośne wydarzenia również w gazetach codziennych. Vora P. opisał jak strajk pocztowy w Stanach Zjednoczonych w 1970 roku wpłynął na przepływ informacji do inwestorów przez niedostarczone gazety

The Wall Street Journal [186]. Wiele oficjalnych dokumentów spółki akcyjne publikują również w formie papierowej.

— **Telefon (i telegraf)** - przed powstaniem Internetu, wiele zleceń kupna i sprzedaży realizowane było przez telefon. Również informacje na temat aktywów, inwestorzy mogli otrzymać telefonicznie od swojego maklera. Był to pierwszy sposób masowej komunikacji na odległość, a jednocześnie o niemal zerowym opóźnieniu.

— **Internet** - od czasów powstania Internetu szybkość transmisji informacji wzrosła bardziej niż kiedykolwiek. Komunikacja stała się niemal natychmiastowa, a forma przekazu jest bardzo elastyczna - pozwala przesyłać tekst, obrazy, wideo, tabele, wykresy. Posiadając internetowe konto maklerskie, można zlecać transakcje dużo łatwiej niż przez telefon lub w placówce.

Mimo ewolucji form przekazu i rozpowszechnianiu się coraz szybszych metod komunikacji, starsze formy nie zanikają całkowicie, ale zostają mocno zmarginalizowane. Ze względu na wolumen dostępnych informacji, a także łatwiejszy dostęp do historycznych wiadomości i możliwość bezpośredniego włączania ich do modelu, niniejsza praca skupia się wyłącznie na wiadomościach pochodzących z internetowych źródeł.

Próbując osiągać zyski, inwestorzy od lat starają się wykorzystać różnego rodzaju informacje do usprawnienia swojego procesu decyzyjnego. Poniżej wymieniono najbardziej powszechnie stosowane kategorie, z których korzysta się współcześnie.

— **Historyczne ceny aktywów** - Obserwując historyczny przebieg cen większości aktywów, można zauważyć, że w krótkim terminie element losowy jest bardzo znaczny, jednak przy dłuższym horyzoncie zaczynają być widoczne trendy i powtarzające się formacje. Jedną z podstawowych technik inwestycyjnych jest analiza techniczna, czyli wykorzystanie podobieństw w przebiegu krótkoterminowej historii do przebiegów z odleglejszej przeszłości i wykorzystanie dalszych trajektorii po zaobserwowaniu do prognozowania przyszłości. Oprócz samych wartości cen popularne jest wyliczanie i korzystanie z wielu wskaźników takich jak: średnie ruchome, oscylator siły trendu - ang. *Relative Strength Index* (RSI), porównanie krótko- i długoterminowych średnich wykładniczych - ang. *Moving Average Convergence / Divergence* (MACD), itp. Według hipotezy rynków efektywnych, nawet w wersji słabej, stosowanie tej strategii nie powinno dawać ponadnormalnych zwrotów. Mimo to, zdarzają się okresy, w których pewne strategie działają (nie dając gwarancji na długoterminową skuteczność). Historyczne ceny są udostępniane nieodpłatnie

przez wiele serwisów internetowych, a także internetowe konta brokerskie, jednak z pewnymi ograniczeniami. Zwykle jest to: niska rozdzielczość czasowa (brak danych poniżej 1 minuty), opóźnienie (brak gwarancji dostarczenia danych nowszych niż 15 minut) oraz niepełna historia (brak dostępnych danych sprzed pewnej daty). Dostęp do danych lepszej jakości najczęściej jest dodatkowo płatny.

— **Raporty finansowe spółek** - Spółki notowane na giełdach papierów wartościowych mają obowiązek regularnie (zwykle co kwartał) raportować swoje wyniki finansowe. Sprawozdania te zawierają między innymi informacje o przychodach i kosztach, wartości posiadanych aktywów. Na podstawie tych wiadomości możliwe jest ocenienie kondycji finansowej spółki. W większości przypadków data publikacji raportu znana jest z góry, dlatego w tym dniu można oczekiwać silniejszych reakcji inwestorów po konfrontacji oczekiwań względem spółki z jej realnymi wynikami. Inwestowanie w oparciu o dane z raportów finansowych jest jedną z najczęściej stosowanych metod obok analizy technicznej.

— **Prospekty emisyjne** - Emitenci papierów wartościowych publikują prospekty emisyjne, czyli dokumenty zawierające informacje na temat sytuacji finansowej spółki, przewidywanego rozwoju, kondycji finansowej i potencjalnego ryzyka. Oprócz tego w prospektach emisyjnych znajdują się też opisy zakresu świadczonych usług lub oferowanych produktów, a także dane na temat zespołu zarządzającego firmą. Emitent informuje również jak zostaną wykorzystane zgromadzone środki i jak planowane jest wypłacanie dywidend. Są to informacje szczególnie cenne, gdy spółka nie jest jeszcze notowane na giełdzie i trudno jest w inny sposób ocenić potencjał inwestycyjny.

— **Prospekty informacyjne** - Fundusze inwestycyjne są zobligowane do publikowania prospektów informacyjnych, w których opisują między innymi prawa uczestników funduszu, warunków kupna i sprzedaży jednostek uczestnictwa, polityki inwestycyjnej, ponoszonego ryzyka, a także wysokości opłat i prowizji z uwzględnieniem podziału na różne kategorie jednostek uczestnictwa. Oprócz tego, zawarte są historyczne wyniki. W swojej funkcji prospekt informacyjny przypomina prospekt emisyjny.

— **Strony internetowe** - Emitenci papierów wartościowych, czyli najczęściej spółki i fundusze inwestycyjne, mają prawny obowiązek składać odpowiednie raporty, jednak często publikują wiele dodatkowych informacji na swoich stronach internetowych. Ze względu na dobrowolność prowadzenia takich stron, najczęściej można

znaleźć na nich wiadomości o sukcesach, nowych produktach, przejęciach konkurencji, itp. Mimo stronniczego zamieszczania informacji, monitorowanie oficjalnych stron może pomóc w predykcji ruchów notowań, zwłaszcza przy ruchach w górę spowodowanych korzystnymi zdarzeniami. Obecnie wiele firm decyduje się również na prowadzenie profili w mediach społecznościowych, gdzie publikują częściej informacje, a oficjalne strony służą do publikowania bardziej formalnych wiadomości.

— **Raporty analityków** - Domy maklerskie i fundusze inwestycyjne zatrudniają zawodowych analityków, którzy na podstawie bieżącej sytuacji i posiadanych informacji pomagają swoim pracodawcom podjąć decyzje inwestycyjne. Te rekomendacje często są publikowane i w zależności od autorytetu oraz historycznych wyników analityka, mogą mniej lub bardziej wpłynąć na popyt pewnych aktywów. Należy jednak uwzględnić, że w wielu przypadkach domy maklerskie zarabiają na prowizjach od transakcji i rekomendacje ich analityków mogą być nastawione na sztuczne zwiększenie zaangażowania inwestorów.

— **Wiadomości o tematyce finansowej** - Monitorowanie wszystkich wydarzeń o tematyce finansowej jest praktycznie niemożliwe do zrealizowania przez pojedynczą osobę. Ilość aktywów na różnych rynkach (krajowych i zagranicznych) oraz wydarzeń, które ich dotyczą i mogą wpływać na cenę, jest tak duża, że powstały specjalistyczne serwisy zajmujące się wyłącznie agregacją, interpretacją i publikacją tych informacji. Początki tego typu serwisów sięgają jeszcze papierowych gazet, obecnie coraz częściej jednak wykorzystywany jest Internet. Znacznym ułatwieniem w wykorzystaniu takich informacji jest fakt, że są już wstępnie przefiltrowane i zinterpretowane. Największą wadą jest opóźnienie z jakim informacja dociera do inwestora, ponieważ konieczny jest czas na zinterpretowanie jej i opisanie przez autora artykułu.

— **Wiadomości o tematyce polityczno-gospodarczej** - Serwisy informacyjne o tematyce polityczno-gospodarczej często są mocno powiązane z tymi o tematyce finansowej (a często najważniejsze dla inwestorów informacje są również zawarte w wiadomościach finansowych). Zmiany prawa, podatków, relacje między państwami (np. stan wojny), działania banków centralnych, itp. mogą mieć olbrzymie znacznie dla wycen aktywów na rynkach konkretnych państw. Większość tych informacji również jest dostępnych w portalach internetowych.

— **Wiadomości o tematyce ogólnej** - Informacje ogólne podawane przez doniesienia agencyjne posiadają osobne działy na tematy finansowe i gospodarcze, ale

można w nich znaleźć również recenzje oferowanych przez firmy usług i produktów, a także plotki i skandale na temat członków zarządu lub bardziej znanych pracowników. Mimo niebezpośredniego związku z wycenami aktywów, mogą wpływać na sentyment inwestorów, szczególnie detalicznych.

— **Blogi** - Mniej oficjalnym źródłem wiadomości są blogi osób zainteresowanych finansami. Przekazywana na nich wiedza często nie podlega żadnej weryfikacji i jest mniej pewna, a opóźnienie często jest większe niż w oficjalnych źródłach. Jednak blogi, ze względu na większe nastawienie na bezpośrednią relację autora treści i jej odbiorców, mogą lepiej służyć do badania sentymentu inwestorów.

— **Fora i grupy dyskusyjne** - W Internecie powstało wiele kanałów do bezpośredniej komunikacji w formie forów i grup dyskusyjnych zrzeszających użytkowników zainteresowanych konkretnym tematem. Część z nich jest publicznych, reszta wymaga logowania lub zaproszenia przez innego użytkownika. W przeciwieństwie do blogów, użytkownicy są z założenia równi sobie (nie ma autorów i odbiorców, jak w przypadku np. blogów). Dzięki takiej formie możliwe jest prowadzenie dyskusji na konkretny temat, bez dodatkowych wiadomości z innych dziedzin (jak w komentarzach na blogach i stronach internetowych). Ze względu na duży wzrost popularności tego typu grup, ilość zrzeszonych użytkowników umożliwia organizację manipulacji rynkowej *pump and dump*, czyli zmowy polegającej na sztucznym pompowaniu ceny wybranego instrumentu finansowego i sprzedaży, kiedy zostanie już wywindowana. Z powodu braku kontroli problem ten obecnie dotyczy głównie kryptowalut, np. Dogecoin, ale również zdarzają się przypadki manipulacji notowaniami spółek akcyjnych, np. ceną akcji spółki GameStop na przełomie 2020 i 2021 roku [114, 131].

— **Media społecznościowe** - Ze względu na coraz większą ilość użytkowników, w Internecie zaczęły powstawać serwisy znacznie ułatwiające dostęp i wymianę informacji. Media społecznościowe powstały jako usługi do utrzymywania kontaktów ze znajomymi, publikowanie zdjęć i opinii. Z powodu prostoty użycia przyciągnęły znacznie więcej użytkowników niż grupy dyskusyjne i obecnie prawie zupełnie je wyparły. Media społecznościowe również umożliwiają tworzenie wieloosobowych grup dyskusyjnych, więc to właśnie tam inwestorzy obecnie wymieniają się informacjami. Obserwując bieżące trendy, można zauważyć również wzrost popularności *copy tradingu*, czyli inwestowania bez zgłębiania potrzebnej wiedzy, lecz kopiowania wszystkich operacji internetowego autorytetu. Platformy takie jak eToro pozwalają prowadzić *copy trading* w całkowicie automatyczny sposób.

— ***Insider trading*** - Dość kontrowersyjną metodą inwestowania było wykorzystywanie informacji niedostępnych publicznie. Osoby zatrudnione w spółkach notowanych na giełdzie często posiadają dostęp do dokładniejszych informacji niż tylko te, które zostały opublikowanie w Internecie. Ze względu na umowne zakazy osobistego inwestowania w spółki, o których posiadają wiedzę, *insiderzy* przekazują swoje informacje rodzinie, znajomym, politykom. Wiele instytucji nadzoru finansowego bardzo starannie zwalcza wszelkie przejawy tego typu działalności, jednak istnieją argumenty za zezwoleniem na *insider trading*, np. [30].

Obecnie Internet jest najbardziej powszechnym i najchętniej stosowanym przez inwestorów instytucjonalnych i detalicznych źródłem wiedzy. Ilość, jakość i szybkość, z jaką mogą być przesyłane informacje, jest największa do tej pory w porównaniu z historycznymi jej źródłami. Poza tym, w łatwiejszy sposób można osiągnąć dostęp do historycznych danych. Efektem jest osiąganie coraz silniejszej efektywności przez rynki. Zwłaszcza rozwój mediów społecznościowych przyczynił się do rozpowszechnienia inwestowania giełdowego wśród młodego pokolenia. W 2017r. Agarwal S. i in. opublikowali pracę, która w zbiorczy sposób podsumowuje literaturę na temat wpływu internetowych źródeł informacji na zachowanie inwestorów na giełdach papierów wartościowych [2].

Szybkość przesyłu informacji jest porównywalna z szybkością światła, a więc o reakcji decyduje czas potrzebny na zinterpretowanie wiadomości i podjęcie decyzji. Czas reakcji zależy również od tego czy wiadomości były oczekiwane (np. zaplanowane z góry ogłoszenie wyników finansowych), czy nieoczekiwane (niespodziewane wydarzenia polityczne, wychodzące na jaw skandale) i może wynosić od 1 do 20 minut. Oprócz początkowych efektów propagacja informacji ma również skutki długofalowe, które powodują późniejsze zmiany ceny (np. jej powrót lub kontynuowanie trendu) przez kolejnych kilkadziesiąt minut do kilku dni.

1.3. Wpływ informacji na decyzje inwestorów

W niniejszym podrozdziale opisano, w jaki sposób pojedynczy inwestor interpretuje informacje i jak wpływają one na jego proces decyzyjny w inwestowaniu. Natomiast podrozdział 1.4 przedstawia wpływ informacji na zachowanie się całych rynków, a zatem efektu decyzji wielu agentów tworzących te rynki.

W dużym uproszczeniu - napływające wiadomości o instrumentach finansowych (lub ich emitentach) można podzielić na trzy kategorie: dobre, złe i neutralne. Dobre

powodują wzrost popytu i spadek podaży, złe odwrotnie, zaś neutralne nie wywołują zmian. Nie każda informacja ma jednakowo dużą wagę, zatem jej wpływ może być różny, a zróżnicowane horyzonty inwestycyjne podmiotów powodują, że ich reakcje na te same wiadomości są inne. Krótkotrwałe spadki lub problemy finansowe spółek mogą nie być wystarczające do sprzedaży udziałów przez podmioty lokujące kapitał na kilkanaście lub kilkadziesiąt lat. Cena rynkowa ustalana jest przez całkowity popyt i podaż, zatem decyduje o niej nastawienie ogółu inwestorów. Nie ma więc konieczności oceniania w jaki sposób dana informacja zostanie odebrana przez każdego inwestora z osobna, aby prognozować cenę. Wyjątkowo istotny jest również ogólny stan gospodarki, historyczne ceny aktywów oraz ostatniej wydarzenia. Surowe dane zawarte w wiadomościach zamieniają się w wartościowe informacje dopiero, gdy można podać ich kontekst [37, 128].

Dwie popularne kategorie metod podejmowania decyzji to: analiza fundamentalna i analiza techniczna. Podstawowym założeniem analizy technicznej jest brak konieczności uwzględniania jakichkolwiek informacji innych niż szereg czasowy ceny instrumentu finansowego, ponieważ wszystkie istotne informacje, które mają na nią wpływ, są już w niej uwzględnione. Na podstawie ceny wyliczane są tzw. wskaźniki analizy technicznej, w wielu przypadkach oparte o arytmetyczne i wykładnicze średnie ruchome różnej długości, wskaźniki siły trendu. Popularne jest również wykorzystanie różnego typu wykresów: liniowych i świecowych. Głównym celem analizy technicznej jest poszukiwanie powtarzających się wzorców, które pozwalają przewidzieć przyszłe wzrosty lub spadki. Zastosowanie odpowiednich wskaźników, formacji świec i innych wzorców mają pozwolić na algorytmiczne podejście do procesu decyzyjnego. Oznacza to nie poleganie na inwestowaniu "uznaniowym" oraz ma zapobiec poddawaniu się emocjom panującym na rynkach lub wynikających z bieżących wydarzeń, co użytkownicy analizy technicznej często zarzucają zwolennikom analizy fundamentalnej. Mimo to analiza techniczna wciąż pozostawia duże pole do własnych interpretacji, np. dobór odpowiednich parametrów liczbowych dla wskaźników lub wzrokowe rozpoznawanie formacji świec [136].

Analiza fundamentalna zwykle odnosi się do akcji spółek (rzadziej innych instrumentów finansowych) i zgodnie ze swoją nazwą, pozwala na zbadanie "fundamentów" aktywa, czyli całościowej sytuacji ekonomicznej emitentów. Stosując różne metody, dokonuje się też wyceny spółki na podstawie dostępnych bilansów księgowych. W analizie fundamentalnej uwzględnia się także wiele dodatkowych czynników: informacje zawarte w prospektach emisyjnych, raporty finansowe spółek, wypłacane w przeszłości dywi-

dendy, analizy rynkowe czy sytuacje makroekonomiczne. Wynikiem przeprowadzonej analizy jest porównanie czy wycena spółki jest niższa niż jej cena giełdowa (wówczas spółkę uznaje się za "niedowartościowaną" i w najbliższej przyszłości można spodziewać się wzrostu cen jej akcji), czy wyższa (wówczas pojawia się sytuacja odwrotna). Należy zauważyć, że sposób włączenia wszystkich dodatkowych informacji do modelu wyceny nie zawsze jest ścisły, więc mimo wykorzystania większej ilości informacji, niż w analizie fundamentalnej, otrzymane prognozy niekoniecznie są lepsze.

Analiza techniczna i fundamentalna często są postrzegane jako przeciwstawne podejścia i wielu inwestorów, zwłaszcza detalicznych, wykorzystuje tylko jedną z nich. Mimo to istnieją sposoby łączenia obu technik - np. przy pomocy analizy fundamentalnej dokonuje się wyboru spółek, zaś przy pomocy analizy technicznej generowany jest sygnał kupna i sprzedaży akcji.

Stosując systematyczne podejście analizy technicznej i fundamentalnej, inwestorom nie zawsze udaje się wytrwać w zamierzonej strategii. Ze względu na stale zmieniające się otoczenie gospodarcze, polityczne i rynkowe, do uczestników rynku cały czas napływają wiadomości o nowych wydarzeniach, które mogą zaważyć na wyniku inwestycji. Zwłaszcza negatywne wiadomości wywierają silny wpływ na podejmowane decyzje, w skrajnych przypadkach wywołując strach i panikę. Często silna reakcja na wyjątkowo negatywne (ale i pozytywne) wiadomości wywołuje zachowania stadne wśród uczestników rynków, tym samym pogłębiając spadki (lub wzrosty) do poziomów nieadekwatnych do ważności powodujących je wydarzeń. Dopiero po dłuższym czasie (godzinach, dniach), ceny instrumentów finansowych powracają do odpowiednich poziomów. Tego typu zjawiska opisują finanse behawioralne [35, 184].

Finanse behawioralne to dziedzina nauki, która bada zachowania inwestorów przy pomocy narzędzi stosowanych w psychologii. W przypadku próby opisu procesu decyzyjnego pojedynczego uczestnika rynku, nie zawsze można przyjąć, że jego zachowanie jest w pełni racjonalne. Przyczyną nieracjonalności są behawioralne aspekty podejmowania decyzji inwestycyjnych. Okoliczności (uwarunkowania) mające wpływ na proces decyzyjny można podzielić na poznawcze i motywacyjne. Uwarunkowania poznawcze to ograniczenia percepcji człowieka przy nadmiarze informacji. W takich sytuacjach ludzie bazują na uproszczonych modelach rzeczywistości, w których często posługują się doświadczeniami z przeszłości, niekoniecznie mającymi zastosowanie w zaistniałej sytuacji. Uwarunkowania motywacyjne to między innymi efekt pewności (wybieranie pewnego zysku, nawet jeśli jego wartość oczekiwana jest niższa niż zysku niepewnego),

awersja do ryzyka w czasie osiągania zysków i skłonność do ponoszenia nadmiernego ryzyka w czasie ponoszenia strat.

Przeciwdziałanie uleganiu uwarunkowaniom poznawczym jest możliwe, lecz wymaga doświadczenia i odpowiedniej edukacji. Z tego powodu uwarunkowaniom częściej (lecz nie jedynie) ulegają inwestorzy indywidualni. W przypadku inwestorów instytucjonalnych pojawiają się takie uwarunkowania jak: nadmierna pewność siebie (wynikająca z przeceniania swoich umiejętności), efekt stadny i podążanie za decyzjami innych profesjonalnych inwestorów.

Wszelkie nieracjonalne zachowania większych grup uczestników rynku prowadzą do anomalii rynkowych: efekty kalendarzowe (np. tzw. efekt stycznia, rajd Świętego Mikołaja), nadreaktywność i subreaktywność rynku w obliczu niektórych wydarzeń, możliwość wykorzystywania strategii momentum (istnienie długookresowych trendów). Nieracjonalność pojedynczego inwestora nie byłaby w stanie znacząco i długotrwale zmienić poziomu cen, ale podobne ograniczenia dotyczą (w różnym stopniu) wszystkich uczestników rynku, przez co ich efekty kumulują się. W rezultacie rynki w krótkich okresach mogą być nieefektywne.

Jedną z metod symulacji i przewidywania efektów zachowań pojedynczych inwestorów na zjawiska występujące na rynkach jest modelowanie agentowe. Podejście to polega na tworzeniu symulacji składających się z wielu autonomicznych agentów i interakcji, jakie między nimi zachodzą. Dzięki oddolnemu spojrzeniu na oddziaływania agentów na siebie, możliwe jest odtworzenie zjawisk rynkowych w skali makro, które nie muszą pomijać takich niuansów, jak na przykład nieracjonalne zachowania. Liczba inwestorów na giełdzie jest stosunkowo duża, zatem zbudowanie symulacji z odpowiednią liczbą agentów było możliwe dopiero przy użyciu komputerów. Potwierdzeniem poprawności zbudowanego modelu jest możliwość odtworzenia historycznych zdarzeń giełdowych oraz przebiegi cen (i ich statystyki) odpowiadające tym obserwowanym w rzeczywistości [182].

Pierwszym wieloagentowym modelem giełdy była symulacja zaproponowana przez Kima G. i Markowitza H. w 1989 roku [104]. Model ten zakłada istnienie dwóch typów uczestników giełdy, przy czym każdy z nich posiada część kapitału w gotówce, a część w akcjach. Typ pierwszy dąży do rebalansowania swojego portfela tak, aby zawsze połowa środków była ulokowana na giełdzie, zaś typ drugi utrzymuje środki w stosunku gwarantującym ustalony minimalny kapitał końcowy. Poza tym każdy agent rozpoczyna symulację z jednakowym kapitałem początkowym i nie otrzymuje żadnych

dodatkowych informacji poza aktualną ceną akcji. Mimo prostoty modelu, otrzymane wyniki pozwalają badać destabilizujący wpływ hedgingu portfolio.

Jedne z pierwszych najbardziej istotnych modeli zachowań giełdowych korzystających z podejścia agentowego to m.in.:

— wymieniony wcześniej model Kima-Markowitza,
— ekonofizyczne podejscie Levy'ego M., Levy'ego H. i Solomona S. zakładające interakcje agentów maksymalizujących swoje indywidualne funkcje użyteczności, które w efekcie dają okresowe zmiany cen przy pewnych konfiguracjach,
— kolejny model stworzony przez Solomona, Levy'ego i Huanga, czyli próba symulacji dystrybucji bogactwa między agentami, przypominający model drapieżnik-ofiara, stosunkowo dobrze odtwarza fluktuacje cen, ale nie do końca poprawnie samą dystrybucję bogactwa,
— dwa modele (Conta-Bouchauda i Solomona-Weisbucha) wykorzystujące teorię perkolacji kolejno w finansach i marketingu.
— model Luxa-Marchesi'ego - symulacja spekulacyjnego zachowania, w której udaje się wygenerować zjawisko bańki spekulacyjnej.

Modele te zostały zbiorczo opisane i porównane przez Samanidou E. i in. w 2007 roku [159]. Jednym z wniosków płynących z tej publikacji jest silna specjalizacja każdego z podejść. Ilość różnych zachowań rzeczywistych uczestników rynku jest niemal niemożliwa do uwzględnienia, a w efekcie otrzymane symulacje poprawnie odtwarzają niektóre zjawiska w skali makro, kosztem dokładności odtworzenia innych.

Rozwój komputerów pozwala na skuteczne tworzenie coraz bardziej złożonych agentów używanych w symulacjach. W 2018 roku Llacy B. i Peffer G. opublikowali pracę, w której opisali użycie agentów korzystających z "realistycznych strategii" [119]. W zaproponowanym modelu uwzględniają giełdę papierów wartościowych z wieloma różnymi instrumentami oraz dwa typy agentów: inwestujących na podstawie analizy fundamentalnej oraz inwestujących na podstawie analizy technicznej. Parametry strategii stosowanych przez agentów umożliwiają wieloparametrową kalibrację. Osiągnięte rezultaty to dobre odtworzenie przebiegów cen i ich parametrów, na przykład grube ogony rozkładów stóp zwrotów. Praca ma znaczący wkład w metody walidacji agentowych modeli rynków kapitałowych.

Modele agentowe umożliwiają "oddolne" symulowanie zachowania inwestorów, jednak ostatecznie to całościowy popyt i podaż ustalają cenę. Metodą zbiorczego porów-

nania nastawienia inwestorów do konkretnych aktywów lub ich grup, z pominięciem szczegółowego zgłębiania stosowanych przez nich indywidualnie strategii oraz nieracjonalnych zachowań, jest analiza sentymentu rynkowego. Sentyment inwestorów to ich ogólne nastawienie (pozytywne, negatywne lub neutralne), które może objawiać się zmianami cen, wzmożoną objętością handlu, a także treścią publikowanych treści w Internecie - głównie na blogach, grupach dyskusyjnych i w mediach społecznościowych. Dokładne zmierzenie sentymentu jest trudne, ponieważ wynika ono z subiektywnych odczuć inwestorów, a niekoniecznie obiektywnej oceny sytuacji.

Badania sentymentu inwestorów zostały zainspirowane pojawiającą się na rynkach anomalią nad- i subreaktywnością na pewne typy wiadomości. Barberis N. et al. w 1998 roku opracowali model cen aktywów uwzględniający sentyment uczestników rynku [6]. Ich podejście zostało zwalidowane symulacjami, w których z sukcesem generowali przebiegi cen o parametrach zbliżonych do rzeczywistych. Według ich modelu inwestorzy reagują zbyt silnie na wiadomości o zdarzeniach, które nie miały dotąd miejsca (wiadomości anormalne), a zbyt słabo na ten sam typ często pojawiających się wydarzeń (publikacja wyników finansowych, wypłaty dywidend).

Badanie sentymentu rynkowego w zautomatyzowanej formie przy użyciu narzędzi NLP do analizy wypowiedzi inwestorów w serwisach internetowych zaczęło się rozpowszechniać wraz z rozwojem technik przetwarzania języka naturalnego. Antweiler W. i Frank M. Z. w 2004 roku opublikowali pracę na temat ekstrakcji opinii i „wiedzy tłumu" z grup dyskusyjnych [4]. Podejścia te zostały dokładniej opisane w podrozdziale 3.5.

1.4. Ekonometria finansowa a informacje na rynkach finansowych

Ekonometria to wykorzystanie statystyki matematycznej do badania i rozwiązywania problemów ekonomicznych. W celu przedstawienia jakiegoś zagadnienia, ekonometrycy tworzą model matematyczny danego zjawiska, który jak najdokładniej odwzorowuje dane. Specyfiką ekonometrii finansowej jest modelowanie szeregów czasowych, zaś badane wartości opisane są najczęściej przez zmienne losowe. W wyniku powstaje zatem model stochastyczny zjawiska.

Ekonometria finansowa skupia się głównie na modelowaniu przebiegu cen aktywów w czasie. W ogólności można przyjąć, że celem modelowania jest znalezienie odwzorowania f danego wzorem 1.2

$$f(\mathbf{P}_{t+1}|\mathbf{\Omega}_t), \tag{1.2}$$

gdzie $\mathbf{P}_{t+1}$ jest ceną aktywa w kolejnym okresie $t+1$, a $\mathbf{\Omega}_t$ wszelkimi informacjami posiadanymi w bieżącym okresie t [143].

Ekonometria finansowa umożliwia modelowanie już na podstawie informacji wyłącznie o historycznych cenach, np. wykorzystując autoregresyjny zintegrowany model średniej ruchomej - ang. *Autoregressive Integrated Moving Average* (ARIMA) i uogólniony model autoregresyjny z warunkową heteroskedastycznością - ang. *Generalized Autoregressive Conditional Heteroscedascity* (GARCH). Rzadko jednak wykorzystywane są bezwzględne wartości cen instrumentów. Podstawowymi wielkościami używanymi w ekonometrii finansowej jest stopa zwrotu R_t opisana wzorem:

$$R_{t+1} = \frac{P_{t+1} - P_t}{P_t}, \tag{1.3}$$

lub, ze względu na wygodniejsze modelowanie, logarytmiczna stopa zwrotu r_t:

$$r_{t+1} = ln\frac{P_{t+1}}{P_t} = ln(1 + R_{t+1}). \tag{1.4}$$

Poza stopami zwrotu bardzo istotna jest miara ryzyka związanego z danym instrumentem. Najczęściej stosowane miary to wariancja σ^2 i odchylenie standardowe σ dane wzorami 1.5 i 1.6:

$$\sigma^2 = \frac{\sum_{t=1}^{T}(R_t - \bar{R})^2}{T-1}, \tag{1.5}$$

$$\sigma = \sqrt{\sigma^2}, \tag{1.6}$$

gdzie $\bar{R}$ jest średnią arytmetyczną wszystkich stóp zwrotu w szeregu, a T liczbą rozważanych stóp zwrotu.

Najprostsze modele (jednej zmiennej) bazują wyłącznie na historycznych obserwacjach szeregów czasowych, które starają się przewidzieć. Bardziej złożone podejścia pozwalają pracować z wieloma zmiennymi w ramach jednego modelu - mogą to być modele wielorównaniowe (zawierające kilka zmiennych objaśniających, w tym zmienne egzogeniczne), a także takie z wieloma zmiennymi objaśnianymi. Wówczas modele wykorzystują rachunek wektorowy.

Użycie informacji zawartej wyłącznie w szeregu czasowym historycznych cen in-

strumentów pozwala na estymowanie modeli prognozowania na wiele sposobów, przy czym zamiast cen, używane są logarytmiczne stopy zwrotu. Jednym z podstawowych podejść jest model ARIMA, czyli model procesu niestacjonarnego w zakresie wariancji oraz funkcji kowariancji lub jego odmiana uwzględniająca sezonowość: sezonowy autoregresyjny zintegrowany model średniej ruchomej - ang. *Seasonal Autoregressive Integrated Moving Average* (SARIMA). Szczególnym przypadkiem w tej klasie modeli, jest model błądzenia losowego [143].

Oprócz modeli prognozowania przyszłych wartości szeregów czasowych, mocno rozwinięta jest również klasa modeli zmienności. Zmienność ma bezpośrednie przełożenie na ryzyko inwestycji, dlatego jest to wyjątkowo istotna klasa modeli. Obserwowanie procesów finansowych z dużą częstotliwością pozwala zauważyć występowanie okresów z większą zmiennością cen, jak i z mniejszymi fluktuacjami. Wobec tego założenie stałej wariancji warunkowej procesu stochastycznego nie pozwoli na skuteczne modelowanie w krótkich okresach. Engle R. w 1982r. zaproponował model autoregresyjny z warunkową heteroskedastycznością - ang. *Autoregressive Conditional Heteroscedascity* (ARCH), który dobrze opisuje finansowe procesy stochastyczne uwzględniając: zmienną w czasie wariancję, skupianie danych oraz grube ogony rozkładów [51]. W 1986r. Bollerslev T. w swojej publikacji przedstawił uogólnioną wersję tego modelu: uogólniony model autoregresyjny z warunkową heteroskedastycznością - ang. *Generalized Autoregressive Conditional Heteroscedascity* (GARCH), która jeszcze lepiej opisuje procesy finansowe [12]. Bardziej szczegółowo modele ARIMA i GARCH zostały opisane w rozdziale 2.1.

Jednym z podejść ekonometrii finansowej do poprawienia skuteczności prognoz było włączenie wiadomości tekstowych, na których bazują inwestorzy w swoich decyzjach. Ze względu na całkowicie odmienną strukturę tekstu pisanego dla żywego odbiorcy, konieczne okazało się wstępne przetworzenie tekstu w sposób czytelny dla modeli matematycznych.

Zdecydowana większość modeli szeregów czasowych polega na próbkowaniu o stałej częstotliwości, przez co włączenie informacji pochodzących z wiadomości giełdowych jest utrudnione. Wiadomości tekstowe napisane dla żywych odbiorców cechują się niejednorodną długością i objętością zawartych informacji. Również ich waga dla modelowanego szeregu może być różna. Z tego powodu klasyczne metody ekonometryczne opierały się dotąd głównie na prostych metrykach, takich jak ilość wiadomości w czasie lub analiza sentymentu inwestorów z wykorzystaniem wyłącznie słów kluczowych w nagłówkach.

Jednym z podstawowych pytań związanych z analizą tekstowych wiadomości, na które może odpowiedzieć ekonometria finansowa, jest: jak szybko ceny aktywów reagują na pojawienie się nowych informacji? Istotnym czynnikiem jest zaskoczenie inwestorów. Jeżeli w określonym terminie ma pojawić się ważna informacja, np. publikacja wyników finansowych, uczestnicy rynków są przygotowani na reakcję. Nawet jeśli nie znają treści wiadomości wcześniej, mogą mieć przygotowane strategie na różne okoliczności. W takim przypadku wpływ wiadomości bardzo szybko znajduje swoje odzwierciedlenie w cenach. Czas propagacji znajduje się w zakresie 1-15 minut, co zostało zbadane w wielu publikacjach, między innymi przez Patell J. i Wolfson M. w 1984 dla giełd NYSE i AMEX [148], Jennings R. i Starks L. w 1985 dla giełdy NYSE [97], Louhichi W. w 2008r. dla giełdy Euronext Paris [121]. Powyższe prace skupiają się nie tylko na czasie reakcji inwestorów, ale również trajektorii ceny w czasie, ponieważ nie wszyscy reagują w jednakowym momencie, a kształt krzywej nie jest odpowiedzią skokową.

W przypadku niespodziewanych wiadomości reakcja uczestników giełdy może być znacząco dłuższa. Brooks R. et al. w 2003r. opisali, w jaki sposób ceny instrumentów finansowych notowanych na NYSE i AMEX reagują na wydarzenia, które nie były oczekiwane [15]. Do swojej analizy ręcznie wybrali wydarzenia, które nie mogły być z góry przewidziane, mające negatywny wpływ na spółki, takie jak wyciek ropy z tankowca Exxon Valdez w 1989, katastrofy lotnicze, wybuchy fabryk lub nagła śmierć CEO. W celu kontroli czy stopy zwrotów z akcji spółki dotkniętej wydarzeniem faktycznie są anormalne, autorzy porównują je ze stopami zwrotu z akcji bezpośrednich konkurentów w tym samym czasie. Według ich rezultatów czas zmian jest znacząco dłuższy - niespodziewane wydarzenia zaburzają trajektorie cen aktywów nawet przez 120 minut od ich wystąpienia. W ciągu pierwszych 3 minut badacze nie zaobserwowali zmian w zawieranych transakcjach, następnie cena drastycznie spada przez 15-20 minut. Kolejnych 100 minut różni się dla wybranych wydarzeń. W niektórych przypadkach następuje powrót do średniej, w innych dalsze (lecz wolniejsze) pogłębianie spadków. Informacje o tego typu wydarzeniach mają wpływ nie tylko na ceny, lecz również ilość zawieranych transakcji.

Istnieje duża, widoczna różnica w zachowaniu inwestorów w ciągu pierwszych godzin handlu po niespodziewanym ogłoszeniu wypłaty dywidend, względem sytuacji, gdy publikacja tych informacji jest w z góry określonym momencie. Powodem jest uwzględnienie oczekiwanych dywidend w cenach już wcześniej, nawet jeśli dokładne informacje jeszcze nie są znane. W 2006r. Graham et al. opisali w swojej publikacji po-

równanie reakcji uczestników rynku na oczekiwane i nieoczekiwane ogłoszenia wypłat dywidend [72]. W badaniach posługują się dwiema grupami spółek notowanych na giełdzie NYSE: pierwsza publikuje informacje na temat dywidend regularnie co kwartał, druga informuje inwestorów nieregularnie i bez wcześniejszych zapowiedzi. Praca skupia się głównie na ilości transakcji, zmienności i spreadach cen w pobliżu czasu ogłoszenia wiadomości o dywidendach. W przypadku znanej wcześniej daty ogłoszenia dywidend, zwiększona aktywność ma miejsce jeszcze przed właściwą publikacją informacji. Jest to wynik spójny z rezultatami pracy Barberisa N. et al z 1998 roku [6].

Na zachowanie cen ma jednak wpływ nie tylko moment publikacji wiadomości (spodziewany lub niespodziewany), lecz także informacje w nich zwarte, które mogą być normalne lub anormalne. Przy współczesnej ilości treści publikowanych w Internecie, większość to parafraza oryginalnych wiadomości, przetwarzana przez kolejne serwisy; mało istotne plotki o produktach lub zarządach firm, a czasem nawet celowa dezinformacja. Oryginalne wiadomości, które dostarczają inwestorom **nowe** informacje, zdarzają się stosunkowo rzadko. Praca Glassermana P. i Mamayskyego H. z 2019 roku analizuje sentyment zawarty w wiadomościach przy użyciu podstawowych technik NLP (n-gramów na poziomie całych słów) [65]. Autorzy testują jaki wkład do entropii mają n-gramy pojawiające się w wiadomościach finansowych, a przez to wzrost fluktuacji cen instrumentów na giełdzie NYSE. Estymowana jest m.in. realizowana zmienność indeksu S&P 500 oraz indeks VIX i sentyment wiadomości wykorzystując model wektorowej autoregresji - ang. *Vector Autoregression model* (VAR). Dzięki temu podejściu możliwe było stworzenie modelu oceniającego anormalność pojawiających się wiadomości.

Nieco odmienne podejście prezentuje Lee S. w publikacji z 2009 roku [111]. Jej badania procesu stochastycznego opisującego stopy zwrotu nie odnoszą się do konkretnej treści zawartej w wiadomościach o wydarzeniach dotyczących spółek, wiadomościach makroekonomicznych, wiadomościach o wypłatach dywidend lub wynikających z analizy technicznej, ale do samego procesu nadchodzących informacji, które kształtują ceny. Zaproponowany model umożliwia predykcję okresów ze zwiększonym prawdopodobieństwem skoków stóp zwrotów. Jest on opisywany przez proces dyfuzji skokowej składający się z procesu Browna i dodatkowego komponentu, opisującego skoki procesem Poissona. Otrzymany model pozwala na wykorzystanie go nawet na danych o częstotliwości większej niż dzienna.

Wykorzystanie wiadomości dotyczących partnerów i konkurencji spółki w prognozowaniu cen jej akcji nie podnosi znacząco skuteczności modeli. Chua C. i Tsiaplias S.

zbadali wpływ nadchodzących wiadomości na wyceny akcji spółek, uwzględniając jednak efekty nie tylko dla firm, których informacje dotyczą, lecz także ich konkurencji i partnerów [26]. Do opisu napływu anormalnych wiadomości użyty został wielowymiarowy model Poissona, a do opisu wiadomości normalnych (typowych) wielowymiarowy model GARCH.

Dużym wyzwaniem dla ekonometrii finansowej w ciągu ostatnich lat było opisywanie zjawisk posługując się danymi o coraz wyższej częstotliwości. Dla dziennych szeregów czasowych możliwe jest modelowanie stóp zwrotów. W przypadku danych *intraday* (o częstotliwości większej niż dzienna) większość podejść pozwalało na skuteczne prognozowanie jedynie zmienności stóp zwrotu, jednak zaczęły pojawiać się również metody dla samej wartości stóp zwrotu. Wiele z nich wykorzystuje informacje zawarte w wiadomościach jako zmienne egzogeniczne, jednak istnieją również rozwiązania wykorzystujące jedynie szeregi czasowe cen, np. analiza funkcyjna skumulowanych dziennych zwrotów Kokoszki P. i Zhanga X. [108]. Modele te zostały przedstawione w rozdziale 2.

Wykorzystanie najnowszych technik NLP do ekstrakcji informacji zawartych w wiadomościach pojawiających się w Internecie, a następnie użycie tych dodatkowych przesłanek do poprawy predykcji szeregów czasowych o wyższej częstotliwości to luka badawcza, którą niniejsza praca stara się wypełnić.

1.5. Podsumowanie

Większość współczesnych gospodarek to gospodarki rynkowe. Jednym ze szczególnych typów rynków jest rynek finansowy, którego nadrzędnym celem jest uniezależnienie konsumpcji od posiadanego w danym momencie kapitału. Przedmiotem handlu są ogólnie rozumiane instrumenty finansowe, zaś uczestnikami emitenci, inwestorzy oraz instytucje pośredniczące, regulujące i organizujące ten rynek.

W niniejszej pracy szczególnie istotny jest sposób, w jaki powstają i przepływają informacje na temat wydarzeń na rynkach finansowych. Najwcześniejsze metody propagowania wiadomości to przekaz ustny, tekst drukowany i telefon, jednak współcześnie najczęściej wykorzystywany jest Internet. Przekazywane treści dotyczą historycznych cen aktywów, raportów finansowych spółek, wiadomości o tematyce finansowej. Są to także wpisy w mediach społecznościowych i na blogach. Hipoteza rynku efektywnego, w zależności od przyjętej siły, mówi, że informacje natychmiast znajdują odzwiercie-

dlenie w cenach instrumentów, jednak fizyczna realizacja sprawia, że rynki bywają nieefektywne nawet przez kilkadziesiąt minut.

Nowe informacje wywierają wpływ na decyzje inwestorów - niespodziewane negatywne wiadomości często wywołują gwałtowne i szybkie spadki. W przypadku wiadomości pozytywnych zachodzi podobny efekt, lecz o mniejszej sile. Zachowania pojedynczych inwestorów są przedmiotem badań modelowania agentowego.

Szybka i automatyczna analiza pojawiających się na bieżąco wiadomości o tematyce finansowej może umożliwić wykorzystanie nowych informacji, zanim zrobią to inni uczestnicy rynku. Niniejsza praca skupia się na zbadaniu możliwości wykorzystania technik przetwarzania języka naturalnego (NLP) w celu zwiększenia skuteczności prognozowania notowań giełdowych poprzez włączenie do modeli informacji zawartych w wiadomościach, zanim zostaną one odzwierciedlone w poziomach cen.

W rozdziale 2 opisane zostały metody modelowania finansowych szeregów czasowych. Rozdział 3 przedstawia przegląd współczesnych technik NLP. Proponowana metodologia połączenia tych dwóch podejść opisana została w rozdziale 4, zaś otrzymane przy jej pomocy wyniki znajdują się w rozdziałach 5 i 6.

Rozdział 2

Modelowanie i prognozowanie jednowymiarowych finansowych szeregów czasowych

Możliwość przewidzenia zachowania się cen na giełdach papierów wartościowych jest jedną z najbardziej pożądanych przez inwestorów i badaczy umiejętności. Dokładna prognoza umożliwia świadome alokowanie swojego kapitału i stworzenie bardziej skutecznych strategii inwestycyjnych. Bardzo popularną klasą modeli służącą do prognoz stóp zwrotu w czasie jest autoregresyjny model średniej ruchomej - ang. *Autoregressive Moving Average* (ARMA). Jest on zwykle wykorzystywany w połączeniu z modelem zmienności klasy GARCH, dając w efekcie model klasy ARMA-GARCH opisany w podrozdziale 2.1. Współcześnie coraz częściej używane są również modele uczenia maszynowego przedstawione w podrozdziale 2.2.

Istnieją również wielowymiarowe modele finansowych szeregów czasowych, które jednak nie będą rozważane w niniejszej pracy ze względu na wyjątkowo dużą złożoność obliczeniową przy założonej częstotliwości próbkowania. Ich szczegółowy opis można znaleźć m.in. w publikacjach Engle R. [50] oraz Pajor A. [145].

2.1. Modele klasy ARMA-GARCH

Jednym z podstawowych i jednocześnie najbardziej popularnych narzędzi analizy finansowych szeregów czasowych w ekonometrii finansowej są modele klasy ARMA-GARCH. Modele tej klasy składają się ze struktury ARMA, która opisuje warunkową wartość oczekiwaną procesu stochastycznego oraz struktury GARCH opisującej wariancję warunkową. W zastosowaniu do danych giełdowych, najczęściej korzysta się z szeregu logarytmów stóp zwrotu $ln(r_t)$. Modele ARMA sprawdzają się najlepiej dla danych z długim horyzontem czasowym - rocznym, miesięcznym, dziennym.

2.1.1. Modele ARMA

Na początku omówiona zostanie grupa liniowych modeli ARMA(p, q), które łączą ze sobą model autoregresyjny - ang. *Autoregressive model* (AR) rzędu p oraz model średniej ruchomej - ang. *Moving Average model* (MA) rzędu q.

Model autoregresyjny rzędu p procesu stochastycznego X_t dany jest następującym wzorem:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t, \tag{2.1}$$

gdzie X_t jest wartością modelowanego procesu w chwili t, c jest stałą niezależną od czasu, $\varphi_0, ..., \varphi_p$ parametrami modelu, zaś ε_t ścisłym białym szumem z zerową średnią i stałą wariancją σ^2.

Model średniej ruchomej rzędu q procesu stochastycznego X_t dany jest wzorem:

$$X_t = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t, \tag{2.2}$$

gdzie μ jest średnią szeregu czasowego X_t (jego wartością oczekiwaną), $\theta_0, ..., \theta_q$ parametrami modelu, pozostałe oznaczenia jak w równaniu 2.1.

Model ARMA(p, q) można przedstawić jako kombinację modeli danych wzorami 2.1 i 2.2. Przyjęte zostaje założenie, że wartość oczekiwana procesu średniej ruchomej μ jest równa zero, by uniknąć wprowadzania sumy dwóch stałych. Ogólna postać modelu jest więc następująca:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t, \tag{2.3}$$

gdzie ε_t jest ścisłym białym szumem z zerową średnią i stałą wariancją σ^2, a p i q są nieujemnymi liczbami całkowitymi.

W przypadku procesów o długiej pamięci zdefiniować można model ARIMA(p, d, q), czyli uogólniony model ARMA rzędu (p, q) dla szeregów niestacjonarnych w zakresie wariancji i funkcji kowariancyjnej. Sprowadzenie procesu do formy stacjonarnej odbywa się przez d-krotne różnicowanie badanego szeregu czasowego. Wobec tego model ARIMA(p, 1, q) dla logarytmów cen $ln(P_t)$ jest równoważny modelowi ARMA(p, q) dla logarytmicznych stóp zwrotu r_t, ze względu na podstawowe właściwości logarytmów.

Do identyfikacja procesów ARMA(p, q) wykorzystywana jest funkcja autokorelacji -

ang. *Autocorrelation Function* (ACF) oraz funkcja autokorelacji cząstkowej - ang. *Partial Autocorrelation Function* (PACF). Funkcja ACF dana jest następującym wzorem:

$$R(\tau) = \frac{K(\tau)}{K(0)}, \tag{2.4}$$

gdzie τ jest odstępem między próbkami, $K(\tau)$ to funkcja kowariancyjna wybranego procesu, a $K(0)$ to jego wariancja. Funkcja autokorelacji mówi, czy proces ma właściwość pamięci, tzn. jak długo utrzymuje się wpływ impulsu pojawiającego sie w chwili t.

Funkcja autokorelacji cząstkowej dana jest równaniami:

$$\alpha(0) = 1, \tag{2.5}$$

$$\alpha(\tau) = \boldsymbol{\phi}_{\tau\tau}, \tau \geqslant 1, \tag{2.6}$$

gdzie $\boldsymbol{\phi}_{\tau\tau}$ to ostatni element rozwiązania poniższego układu równań dla $\tau = 0, 1, ..., p$

$$\boldsymbol{\phi}_{\tau} = \boldsymbol{\Gamma}_{\tau}^{-1} K(\tau), \tag{2.7}$$

$$\boldsymbol{\Gamma}_{\tau} = [K(i-j)]_{i,j=1}^{\tau}, \tag{2.8}$$

$$K(\tau) = [K(1), K(2), ..., K(\tau)], \tag{2.9}$$

gdzie $\alpha(\tau)$ to funkcja autokorelacji cząstkowej. Właściwości procesu mogą być badane na podstawie przebiegów (szybkości zanikania) jego funkcji autokorelacji cząstkowej.

Doman M. i Doman R. odradzają wykorzystanie funkcji ACF oraz PACF [45]. Zamiast nich do wyznaczenia rzędów p i q zalecają użycie jednego z kryteriów informacyjnych. Kryterium Informacyjne Akaikego - ang. *Akaike Information Criterion* (AIC) i Bayesowskie Kryterium Informacyjne Schwarza - ang. *Bayesian Information Criterion* (BIC) dane są następującymi wzorami:

$$AIC = -2\frac{\ln L(\hat{\theta})}{T} + 2\frac{a}{T}, \tag{2.10}$$

$$BIC = -2\frac{\ln L(\hat{\theta})}{T} + 2\frac{a\ \ln T}{T}, \tag{2.11}$$

gdzie $\hat{\theta}$ jest wektorem oszacowanych parametrów modelu, $L(\hat{\theta})$ funkcją wiarygodności

dla modelu z parametrami $\hat{\theta}$, T ilością modelowanych obserwacji, zaś a jest liczbą wszystkich parametrów modelu.

Procedura polega na przyjęciu wysokiej wartości parametrów p_{max} oraz q_{max}, a następnie badaniu wybranego kryterium informacyjnego dla modeli zbudowanych używając wszystkich kombinacji $0 \leqslant p \leqslant p_{max}$ i $0 \leqslant q \leqslant q_{max}$. Wybrany zostaje model dający najmniejszą wartość kryterium informacyjnego.

Estymacja parametrów modeli ARMA(p, q) dla danych empirycznych w ogólności jest możliwa przez wykorzystanie metody największej wiarygodności. Jest to zagadnienie czysto numeryczne i zostało zaimplementowane w wielu gotowych bibliotekach dla języków R, Python i innych. Prognozowanie wartości procesu w kolejnym okresie $t + 1$ przy pomocy modeli ARMA(p, q) polega na obliczeniu warunkowej wartości oczekiwanej procesu:

$$\tilde{x}_{t+1} = E[x_{t+1}|x_t, x_{t-1}, ...]. \tag{2.12}$$

Prognozy z dalszym horyzontem czasowym h wymagają iteracyjnego obliczenia prognoz dla okresów $t+1, ..., t+h-1$ i podstawienie ich w miejsce odpowiednich historycznych obserwacji, zatem wartość prognozy zależy od tego samego szeregu czasowego:

$$\tilde{x}_{t+h} = E[x_{t+h}|x_t, x_{t-1}, ...]. \tag{2.13}$$

2.1.2. Modele GARCH jako narzędzie analizy zmienności finansowych szeregów czasowych

W przypadku danych o dużej częstotliwości (dla cen instrumentów finansowych to dane dzienne, godzinowe, minutowe lub o jeszcze większej częstotliwości próbkowania) modele ARMA nie dają dobrych rezultatów, ponieważ trudno jest badać zależności przyczynowo-skutkowe. Zdarza się, że przyczyna i skutek nie są punktem w czasie, lecz procesami, których czas trwania może się pokrywać. Przykładem może być zmiana ceny w odpowiedzi na pewne wydarzenie - inwestorzy reagują dopiero po otrzymaniu informacji, co może następować w różnych chwilach dla każdego z nich.

Szeregi cen instrumentów finansowych obserwowane z dużą częstotliwością charakteryzują się zależną od czasu zmiennością oraz klastrowaniem zmienności, czyli występowaniem naprzemiennie okresów o mniejszych i większych fluktuacjach. W takim

przypadku podstawowym narzędziem ekonometrii finansowej są modele zmienności z rodziny GARCH.

Niech punktem wyjścia do dalszych rozważań będzie proces stochastyczny, opisujący logarytmiczne stopy zwrotu, który można przedstawić następująco (za Doman M. i Doman R. [45]):

$$r_t = \mu_t + \varepsilon_t, \tag{2.14}$$

gdzie μ_t jest średnią warunkową pod warunkiem zbioru informacji na temat procesu r_t dostępnych w chwili $t-1$ włącznie, a ε_t jest ciągiem niezależnych zmiennych losowych o tym samym rozkładzie ze średnią zero i zmienną wariancją σ_t^2.

Podstawowy model ARCH(q) (model autoregresyjny z warunkową heteroskedastycznością rzędu q) został zaproponowany przez Engle'a R. w artykule z 1982 roku [51]. Jest opisany za pomocą dwóch równań:

$$\varepsilon_t|\psi_{t-1} \sim \ N(0, h_t), \tag{2.15}$$

$$h_t = h(\varepsilon_{t-1}, \varepsilon_{t-2}, ..., \varepsilon_{t-p}, \alpha), \tag{2.16}$$

Najczęściej zakłada się też, że funkcja h jest dana wzorem:

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2, \tag{2.17}$$

gdzie $\alpha_0 > 0$ oraz $\alpha_i \geqslant 0$ dla $i > 0$.

Zwykle w zastosowaniu do szeregów czasowych logarytmicznych stóp zwrotów aktywów w miejsce równania 2.14 stosuje się stacjonarny model klasy ARMA.

Model ARCH(q) może zostać uogólniony do modelu GARCH(q, p) poprzez dodanie do wzoru 2.16 bezpośredniej zależność od przeszłych wartości h_{t-i}:

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j}, \tag{2.18}$$

gdzie $\alpha_0 > 0$, $\alpha_i \geqslant 0$ dla $i > 0$ oraz $\beta_i \geqslant 0$. Model ten został zaproponowany i zbadany przez Bollsersleva T. w 1986 [12]. Dla finansowych szeregów czasowych sprawdza się on lepiej niż model ARCH, ponieważ dokładniej odwzorowuje rozkłady o grubych ogonach, a procesy finansowe wymagają dużych rzędów opóźnień q.

W rzeczywistych zastosowaniach ekonometrycznych najczęściej ustala się $p = q = 1$

(w przypadku modelowania pojedynczych spółek, zdarza się, że lepiej sprawdzają się modele GARCH(1, 2) i GARCH(2, 1) dają lepsze wyniki [60]). Tak zdefiniowany model GARCH nie zawiera w sobie modułu opisującego autokorelacji stóp zwrotu, przez co konieczne jest równoległe wykorzystanie modelu ARMA do ich opisu.

Podczas budowy modelu GARCH należy ustalić typ rozkładu zmiennej losowej ε_t. Wykorzystane mogą być różne rozkłady dające w efekcie różny wpływ na zachowanie modelu. Zostały one opisane między innymi w publikacjach [60,151]. Poniżej dokładniej przedstawione są dwa rozkłady wykorzystane w niniejszej pracy.

Wzór 2.15 przedstawia najczęściej stosowany rozkład normalny o wartości oczekiwanej 0. W ogólności może to być rozkład normalny o dowolnych parametrach:

$$\varepsilon_t|\psi_{t-1} \sim \ N(\mu, \sigma^2), \tag{2.19}$$

gdzie μ jest wartością oczekiwaną, a σ^2 wariancją. Funkcja gęstości rozkładu prawdopodobieństwa dla rozkładu normalnego dana jest następującym wzorem:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \ \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{2.20}$$

Badania empiryczne wskazują, że finansowe szeregi czasowe posiadają istotnie grubsze ogony (większą kurtozę) w swoich rozkładach brzegowych niż zaproponowany w podstawowej wersji (wzór 2.15) rozkład normalny. Lepsze efekty daje zwykle wykorzystanie rozkładu t-Studenta zaproponowane już w oryginalnej pracy Bollersleva T. [12]:

$$\varepsilon_t|\psi_{t-1} \sim \ St(\mu, \sigma^2, \nu), \tag{2.21}$$

gdzie μ to modalna, σ to parametr skali (fizyczna interpretacja tych dwóch parametrów jest podobna do rozkładu normalnego), zaś ν jest liczbą stopni swobody. Funkcja gęstości prawdopodobieństwa tego rozkładu dana jest następującym wzorem:

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\sigma^2\nu\pi}\,\Gamma(\frac{\nu}{2})}\left(1+\frac{(x-\mu)^2}{\sigma^2\nu}\right)^{-(\frac{\nu+1}{2})}, \tag{2.22}$$

gdzie Γ to funkcja Gamma. Przy wzroście liczby stopni swobody ν rozkład t-Studenta zmierza do rozkładu normalnego.

Poza opisanymi wyżej rozkładami prawdopodobieństwa (wykorzystywanymi przez autora w badaniach) wymienić można również: skośny rozkład t-Studenta [142], uogólniony rozkład błędu GED [88,135], rozkład α-stabilny [117], mieszaniny rozkładów [87],

rozkład hiperboliczny [174]. Rozkłady te często mają zastosowanie w specyficznych sytuacjach, kiedy rozkład normalny i rozkład t-Studenta nie dają zadowalających wyników.

Modyfikacje modelu GARCH mogą obejmować nie tylko zmianę rozkładu zmiennej losowej ε_t, ale także jego specyfikacji (parametryzacji). Powstało wiele propozycji ulepszeń modelu, których celem było uwzględnienie zjawisk obserwowanych empirycznie w finansowych szeregach czasowych, a które nie były odzwierciedlone w modelu podstawowym. Ze względu na mnogość obecnie istniejących specyfikacji, opisane zostaną dwie z nich, które obok podstawowej wersji modelu zostały wykorzystane w niniejszej pracy:

W przypadku wszystkich specyfikacji modelowana jest zmieniająca się w czasie wariancja warunkowa oraz efekt skupiania się wariancji w pewnych okresach. Modyfikacje modelu GARCH mają na celu głównie uwzględnienie asymetrycznego wpływu dodatnich i ujemnych stóp zwrotu na wariancję.

Pierwszy z tych modeli - wykładniczy model GARCH - ang. *Exponential GARCH* (EGARCH) został zaproponowany przez Nelsona D. w 1991 roku [135]. Jedną z zalet tej specyfikacji jest zastosowanie logarytmicznej postaci wariancji warunkowej. Dzięki takiemu rozwiązaniu dodatniość wariancji warunkowej jest automatycznie zagwarantowana bez nakładania ograniczeń na parametry modelu. Równanie modelu EGARCH dane jest następującym wzorem:

$$\ln h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \left\{\theta z_{t-i} + \lambda(|z_{t-i}| - \mathbb{E}(|z_{t-i}|)\right\} + \sum_{j=1}^{p} \beta_j \ln h_{t-j}, \tag{2.23}$$

gdzie z_t jest zmienną standaryzowaną (co ogranicza wpływ wartości nietypowych na wariancje warunkową) zastępującą ε_t. Dla rozkładu normalnego $\mathbb{E}(|z_{t-i}|) = \sqrt{2/\pi}$, jednak można przyjąć inne typy rozkładów opisane szerzej np. w [45]. Dodatkowo zakłada się $\alpha \equiv 1$ ze względu na nadmiarowość jednego parametru.

Drugi model to GJR-GARCH, którego nazwa pochodzi od nazwisk twórców: Glostena L., Jagannathana R., i Runkle'a D. Zaproponowali tę specyfikację w 1993 roku z zamiarem oszacowania efektu dźwigni poprzez estymację dodatkowego modułu we wzorze wariancji warunkowej [67]. Model GJR-GARCH dany jest wzorami:

$$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{q} \omega_i I_{t-i} \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j}, \tag{2.24}$$

oraz

$$I_t = \begin{cases} 1, & \text{gdy } \varepsilon_t \leqslant 0, \\ 0, & \text{gdy } \varepsilon_t > 0, \end{cases} \tag{2.25}$$

gdzie ω_i jest nowym parametrem opisującym siłę efektu dźwigni - pozwala modelować poziom asymetrii wariancji warunkowej w odpowiedzi na dodatnie lub ujemne stopy zwrotu [45]. Dodatkowo konieczne są założenia gwarantujące dodatniość wariancji warunkowej: $\alpha_0 > 0$, $\alpha_i \geqslant 0$, $\alpha_i + \omega_i \geqslant 0$ dla $i = 1, ..., q$, $\beta_j \geqslant 0$ dla $j = 1, ..., p$.

Tabela 2.1. Specyfikacje modelu GARCH.

Nazwa modelu	**Specyfikacja modelu**
ARCH(q) Źródło: [51]	$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2$
GARCH(p, q) Źródło: [12]	$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j}$
EGARCH(p, q) Źródło: [135]	$\ln h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \left\{\theta z_{t-i} + \lambda(\lvert z_{t-i}\rvert - \mathbb{E}(\lvert z_{t-i}\rvert)\right\}) + \sum_{j=1}^{p} \beta_j \ln h_{t-j}$
GJR-GARCH(p, q) Źródło: [67]	$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{q} \omega_i I_{t-i} \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j}$
TGARCH(p, q) Źródło: [152]	$\sqrt{h_t} = \alpha_0 + \sum_{i=1}^{q} (\alpha_i^+ \varepsilon_{t-i}^+ - \alpha_i^- \varepsilon_{t-i}^-) + \sum_{j=1}^{p} \beta_j \sqrt{h_{t-j}}$
AGARCH(p, q) Źródło: [54]	$h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i (\varepsilon_{t-i} - \gamma_i \sqrt{h_{t-i}})^2 + \sum_{j=1}^{p} \beta_j h_{t-j}$
GARCH-M(p, q) Źródło: [53]	$r_t = \boldsymbol{x'}_t \boldsymbol{\xi} + \delta g(h_t) + \varepsilon_t$ $h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j}$
GARCH-MT(p, q) Źródło: [61]	$y_t = c + \delta^+ I_t h_t + \delta^- (1 - I_t) h_t + \varepsilon_t$ $h_t = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j h_{t-j}$
IGARCH(p, q) Źródło: [52]	$h_t = \alpha_0 + \varepsilon_{t-1}^2 + \sum_{i=1}^{q} \alpha_i (\varepsilon_{t-i}^2 - \varepsilon_{t-1}^2) + \sum_{j=1}^{p} \beta_j (h_{t-j} - \varepsilon_{t-1}^2)$
FIGARCH(p, q) Źródło: [5]	$\varphi(L)(1 - L)^d \varepsilon_t^2 = \alpha_0 + [1 - \beta(L)]\nu_t$
GARCHNet Źródło: [17]	$h_t = ln(1 + exp(W_{Vl} l_n + b_V))$

Źródło: Opracowanie własne na podstawie [60].

W tabeli 2.1 zostały zebrane opisane wyżej specyfikacje modelu GARCH, wykorzystywane w niniejszych badaniach, a także kilka bardziej popularnych wersji tego modelu wraz z ich równaniami i publikacjami opisującymi je bardziej szczegółowo. Parametry przedstawionych specyfikacji muszą podlegać pewnym ograniczeniom, aby zapewnić dodatniość h_t. Restrykcje te zawarte są w oryginalnych publikacjach wymienionych przy nazwie każdej ze specyfikacji.

Modele klasy GARCH mają bardzo szerokie zastosowanie. Poprzez modyfikację specyfikacji modelu i łączenie go z innymi członami, możliwe jest wykorzystanie go do badań opartych nie tylko o dane czysto ekonometryczne. Oloko T. et al. wykorzystując model GARCH-MIDAS dowiedli istnienia wpływu zmian klimatycznych na powiększenie zmienności stóp zwrotu na azjatyckich rynkach finansowych [138]. W tym przypadku dane dotyczące klimatu zostały włączone do modelu jako zmienna egzogeniczna.

Przykładem zastosowania modelu klasy GARCH całkowicie poza dziedziną ekonometrii finansowej jest publikacja Dai H. i in. [36] dotycząca modelowania zmienności zanieczyszczeń powietrza $PM_{2.5}$. W swoich badaniach Autorzy wykorzystali hybrydowy model XGBoost-GARCH-MLP.

Parametry modeli GARCH estymuje się zazwyczaj za pomocą metody największej wiarygodności lub metody quasi-największej wiarygodności [60]. Logarytm funkcji wiarygodności dla modeli klasy GARCH przyjmuje następującą postać:

$$\ln L(\boldsymbol{\theta}) = \sum_{t=1}^{T} \ln f(r_t|\boldsymbol{\psi}_{t-1}; \boldsymbol{\theta}), \tag{2.26}$$

gdzie $\boldsymbol{\theta}$ jest wektorem wszystkich parametrów modelu, $\boldsymbol{\psi}_{t-1}$ zbiorem wszystkich informacji dostępnych w chwili $t-1$, zaś f jest funkcją gęstości rozkładu warunkowego stopy zwrotu r_t. Estymacja modelu metodą największej wiarygodności polega więc na wyliczeniu $\hat{\theta}$ poprzez maksymalizację wzoru 2.26:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}}\ L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}}\ \ln L(\boldsymbol{\theta}). \tag{2.27}$$

Wartości pochodnych cząstkowych koniecznych do obliczeń są w ogólnym przypadku wyznaczane numerycznie, a w niektórych przypadkach szczególnych mogą zostać wyliczone analitycznie, aby przyspieszyć cały proces. W praktyce zwykle estymuje się jednocześnie strukturę ARMA oraz GARCH, ponieważ otrzymywane tak estymatory są dokładniejsze.

Długość szeregów wykorzystywanych do estymacji modeli powinna być ustalona eksperymentalnie. Z jednej strony długi szereg może zawierać wiele informacji i dać lepszy model (o ile jest on stabilny, a czas estymacji nie jest problemem) - wówczas można wykorzystać wszystkie dostępne dane. Z drugiej zaś strony, jeżeli właściwości szeregu są zmienne w czasie, należy wykorzystać szeregi krótsze. Badania nie wskazują jednoznacznie, które podejście jest lepsze [60].

Wybór rzędów dla łączonej struktury ARMA-GARCH również odbywa się poprzez wykorzystanie kryteriów informacyjnych AIC lub BIC dane wzorami 2.10 i 2.11. Kryteria te zostały opisane dokładniej w rozdziale 2.1.1.

Większość dostępnych bibliotek numerycznych pozwala na automatyczną estymację modeli GARCH oraz jednoczesną estymację modeli ARMA-GARCH. W niniejszej pracy do zbudowania wszystkich modeli ARMA-GARCH został wykorzystany pakiet *rugarch* dla języka R.

Prognozy modelu GARCH mogą zostać poprawione poprzez włączenie zmienności zrealizowanej jako dodatkowej zmiennej objaśniającej. Doman M. w 2004 roku opublikowała artykuł o modelowaniu zmienności polskich indeksów giełdowych przy użyciu modeli GARCH [44]. W swoich badaniach użyła dwóch typów danych: o rozdzielczości dziennej oraz 20-minutowej dla indeksów WIG20 oraz TechWIG z Giełdy Papierów Wartościowych w Warszawie, m. in. pokazując, że włączenie zmienności zrealizowanej podnosi skuteczność predykcji.

W publikacji z 2022 roku Vo N. i Ślepaczuk R. opisali możliwość zastosowania różnych wariantów modeli ARIMA-GARCH w strategiach inwestycyjnych na indeksie S&P 500 [185]. Użyte w badaniu dane miały rozdzielczość dzienną. Badaczom udało się osiągnąć strategie bardziej skuteczne niż strategia "kup i trzymaj" lub te oparte o modele ARIMA, wobec czego wnioskują, że modele ARIMA-GARCH pozwalają na skuteczniejsze prognozowanie.

Nieco odmienne podejście do prognozowania danych o wysokiej częstotliwości prezentują Kokoszka P. i Zhang X. w publikacji z 2012 roku [108]. Zamiast logarytmicznej stopy zwrotu, wyliczanej jako różnica logarytmów cen miedzy kolejnymi okresami, wykorzystują skumulowane dzienne stopy zwrotu $r_n(t_j)$:

$$r_n(t_j) = 100[lnP_n(t_j) - lnP_n(t_1)], \tag{2.28}$$

gdzie n jestem indeksem kolejnych dni, zaś t_j oznacza czas o indeksie j liczony od początku sesji, przy czym $t_{j+1} - t_j = 1$ minuta. Wykorzystywane dane mogą być interpretowane jako krzywe skumulowanych stóp zwrotu dla każdego dnia osobno. W swoim podejściu korzystają z analizy funkcjonalnej, modelując przebiegi skumulowanych stóp zwrotu jako funkcje rozłożone w szereg Fouriera. To podejście jest obliczeniowo bardziej złożone niż ARIMA-GARCH, jednak pozwala skutecznie tworzyć modele dla danych o wysokiej częstotliwości.

2.2. Modele uczenia maszynowego

Rozwój uczenia maszynowego (ML) w ostatnich dekadach pozwolił na tworzenie ateoretycznych modeli w wielu dziedzinach nauki. W przeciwieństwie do klasycznych programów komputerowych, dla których kod musi być napisany wprost, algorytmy uczenia maszynowego poprawiają swoją skuteczność automatycznie ucząc się danych.

Najbardziej popularna ogólna definicja uczenia maszynowego została podana przez Mitchella T. [129] w następującej formie (tłum. autora):

> Program komputerowy uczy się na podstawie doświadczenia E w odniesieniu do klasy zadań T i miary efektywności P, jeżeli jego efektywność wykonania zadania T mierzona za pomocą miary P rośnie wraz ze wzrostem doświadczenia E.

Ze względu na sposób reprezentacji danych wejściowych i wyjściowych oraz zadania T wyróżnia się 3 główne klasy uczenia maszynowego, które można opisać na podstawie powyższej definicji, stosując probabilistyczny punkt widzenia przedstawiony przez Murphy'ego K. [132]:

1. **Uczenie nadzorowane** - Jest to najczęściej stosowana forma ML. W tym przypadku zadaniem T jest znalezienie odwzorowania f danych wejściowych $\mathbf{x} \in \mathcal{X}$ na dane wyjściowe $\mathbf{y} \in \mathcal{Y}$. Aby umożliwić algorytmowi nauczenie się odwzorowania, doświadczenie E jest reprezentowane w formie N par $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$. Zbiór $\mathcal{D}$ nazywany jest **zbiorem treningowym**. Dane wyjściowe $\mathbf{y} \in \mathcal{Y}$ są traktowane jako znana z góry i poprawna informacja. Odwzorowanie f może być rozumiane jako model prawdopodobieństwa warunkowego $p(\mathbf{y}|\mathbf{x})$.
2. **Uczenie nienadzorowane** - W przeciwieństwie do uczenia nadzorowanego, nie istnieje zbiór znanych odpowiedzi dla zadanych danych wejściowych. Dostępny jest zbiór $\mathcal{D} = \{\mathbf{x}_n : n = 1, ..., N\}$. Zadaniem T uczenia nienadzorowanego jest stworzenie modelu rozkładu prawdopodobieństwa $p(\mathbf{x})$, który może posłużyć do generowania nowych próbek danych o tych samych parametrach.
3. **Uczenie przez wzmacnianie** - Najbardziej zbliżoną formą ML do rzeczywistego procesu uczenia się żywych organizmów jest uczenie przez wzmacnianie. Zadaniem T jest nauczenie się systemu (agenta), w jaki sposób powinien wchodzić w interakcje ze swoim otoczeniem. Agent podejmuje decyzje o wykonywanych akcjach $\mathbf{a}$ na podstawie obserwacji $\mathbf{x}$ (danych wejściowych) stanu $\mathbf{s}$, w jakim się znajduje w swoim otoczeniu. System nie dostaje wprost informacji, jak powinien się zachować w odpowiedzi na dane obserwacje. Zamiast tego agent sporadycznie otrzymuje sygnał

nagrody (lub kary). Dzięki temu uczy się polityki π - funkcji odpowiadającej za jego zachowanie: $\mathbf{a} = \pi(\mathbf{x})$.

W zastosowaniu do prognozowania szeregów czasowych najczęściej stosowane jest uczenie nadzorowane przy założeniu, że $\mathbf{y}_n$ to zbiór historycznych obserwacji modelowanej wielkości, zaś $\mathbf{x}_n$ to jej zmienne objaśniające - w tym opóźnione zmienne endogeniczne, a także wszelkie zmienne egzogeniczne (występujące w czasie nie później niż zmienna objaśniana). Łatwo zauważyć, że przy takim sformułowaniu modeli uczenia nadzorowanego, modele ekonometrii finansowej są ich podzbiorem.

Uczenie maszynowe zwykle wykorzystuje dane wejściowe wstępnie przetworzone nieliniowymi transformacjami, które przez rozwinięcie w szereg Taylora można przedstawić w formie wielomianu rzędu D. Wówczas odwzorowanie f ma postać:

$$f(x; \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_D x^D, \tag{2.29}$$

gdzie $\mathbf{w}$ jest wektorem parametrów wielomianu [172, 173].

Stosując skróconą notację $\phi(x) = [1, x, x^2, ..., x^D]$ wzór 2.29 można zapisać jako:

$$f(x; \mathbf{w}) = w^T \phi(x). \tag{2.30}$$

W przypadku, gdy dane wejściowe są wielowymiarowe $\mathbf{x} \in \mathcal{R}^M$, funkcja ϕ przyjmuje postać:

$$\phi(\mathbf{x}) = [1, x_1, x_2, ..., x_1^2, x_2^2, ...], \tag{2.31}$$

a funkcja f:

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_{M+1} x_1^2 + w_{M+2} x_2^2 + ..., \tag{2.32}$$

W początkowych latach rozwoju ML, eksperymentowanie z różnorodnymi transformacjami danych wejściowych zajmowało badaczom wiele czasu. Dokonywanie tych nieliniowych przekształceń w sposób automatyczny jest obecnie możliwe dzięki zastosowaniu głębokich sieci neuronowych. Jeżeli funkcja $\phi(\mathbf{x})$ będzie posiadać zestaw dodatkowych parametrów $\mathbf{V}$, wówczas funkcję f daną wzorem 2.32 można zapisać w postaci:

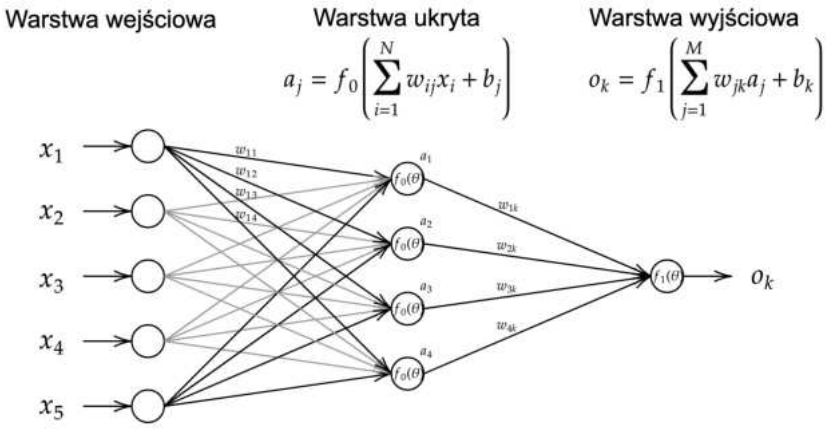
$$f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x}; \mathbf{V}), \tag{2.33}$$

która jest tożsama z siecią neuronową o głębokości L. Można przepisać powyższy wzór w formie, która wprost obrazuje przekształcenia dokonywane przez kolejne jej warstwy [129]:

$$f(\mathbf{x}; \boldsymbol{\theta}) = f_L(f_{L-1}(...(f_1(\mathbf{x}))...)), \tag{2.34}$$

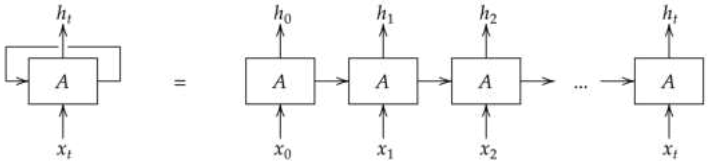
gdzie $f_l(\mathbf{x}) = f(\mathbf{x}; \boldsymbol{\theta}_l)$ dla $l = 1, ..., L-1$, zaś ostatnia warstwa jest funkcją liniową postaci $f_L(\mathbf{x}) = \mathbf{w}^T f_{1:L-1}(\mathbf{x})$. Wszystkie typy sieci neuronowych (gęsto połączone, rekurencyjne, konwolucyjne) można zdekomponować w wyżej opisany sposób.

Wygodną formą przedstawienia sieci neuronowych jest schemat ich architektury, na którym zaznaczone są przepływy informacji między kolejnymi warstwami neuronów. Przykładowa sieć z dwiema warstwami ($L = 2$) została przedstawiona na rysunku 2.1.



Rysunek 2.1. Schemat sieci neuronowej z jedną ukrytą warstwą. Dane wejściowe to wektor dowolnego wymiaru. Warstwy ukryte przeprowadzają kolejno dwie operacje - mnożenie przez macierz z przesunięciem o stały wektor, a następnie zaaplikowanie funkcji f_l. Zwykle jako f_l zostaje wybrana funkcja nieliniowa, aby umożliwić sieci nauczenie się potencjalnych nieliniowych zależności w danych.
Źródło: Opracowanie własne.

W modelowaniu szeregów czasowych szczególnie przydatną klasą sieci neuronowych jest Rekurencyjna Sieć Neuronowa - ang. *Recurrent Neural Network* (RNN), która w naturalny sposób uwzględnia ciągi przyczynowo - skutkowe. Oprócz zwykłych danych wchodzących do sieci, dodatkowym wejściem są dane wyjściowe wygenerowane w poprzednim kroku. Może to być zobrazowane jako pętla sprzężenia zwrotnego lub iteracyjne rozwinięcie całej sieci na poszczególne kroki (rysunek 2.2).

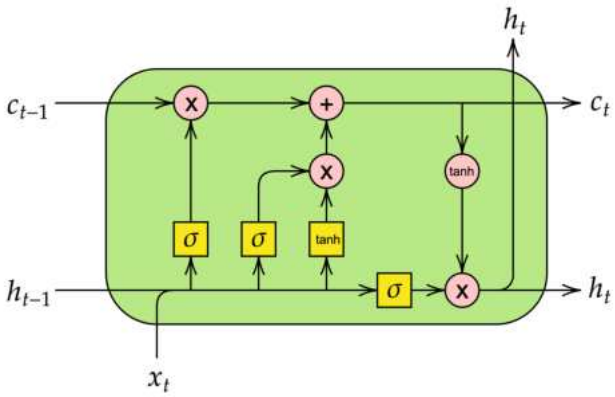


Rysunek 2.2. Schemat przepływu danych w rekurencyjnej sieci neuronowej. Sieć A na wejściu przyjmuje kolejne wektory wejściowe oraz wektor stanu z chwili $t-1$. W ten sposób możliwe jest nauczenie sieci, aby pamiętała informacje przetwarzane w przeszłości i wykorzystywała je do wnioskowania na podstawie nowych danych.
Źródło: Opracowanie własne.

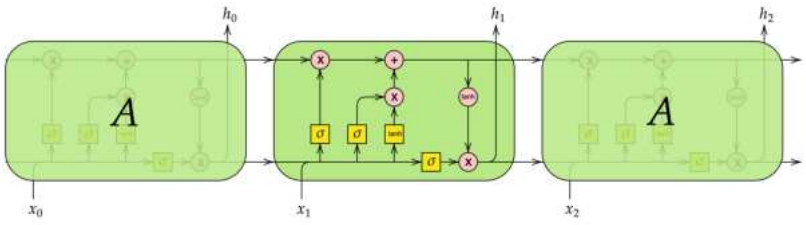
Działanie sieci RNN polega na iteracyjnym przetwarzaniu szeregu danych wejściowych, stąd możliwość zastosowania do szeregów czasowych i danych sekwencyjnych, np. tekstu. Poza zwykłymi połączeniami kolejnych warstw w RNN występują również sprzężenia zwrotne - rezultat obliczeń wybranych neuronów dla próbki danych w kroku t zostaje zapamiętany i przekazany jako jedno z wejść tych samych lub innych (w zależności od typu sieci) neuronów w kroku $t+1$. W czasie procesu trenowania wagi połączeń zwrotnych są również optymalizowane, przez co sieć uczy się zapamiętywać pewne informacje (staje się systemem stanowym) [99, 158].

Dużym ograniczeniem sieci RNN jest stopniowe zanikanie informacji, które pojawiły się na początku przetwarzanego szeregu czasowego, przez co sieci te nie są w stanie wyciągać długoterminowych wniosków. W 1997 roku Hochreiter S. i Schmidhuber J. zaproponowali rozwiązanie tego problemu przez modyfikację budowy pojedynczego neuronu [84]. Powstały w ten sposób nowy typ sieci wyróżniała Długa Pamięć Krótkotrwała - ang. *Long Short-Term Memory* (LSTM). Neuron LSTM oprócz przekazywania dalej informacji posiada dodatkowy parametr odpowiedzialny za decyzję, czy daną informację należy zapamiętać w długim terminie, czy zapomnieć. Schemat neuronu jest przedstawiony na rysunku 2.3, zaś sposób przepływu informacji przez sieć typu LSTM został przedstawiony na rysunku 2.4.

Ważną zaletą sieci neuronowych jest olbrzymia elastyczność co do typu modelowanych danych, zwłaszcza w porównaniu do klasycznych modeli ekonometrycznych. Wykorzystanie wielowymiarowych danych jest wspierane przez większość pakietów do trenowania sieci neuronowych (np. TensorFlow [1]). Wadą natomiast jest duży rozmiar otrzymywanych modeli (duża liczba parametrów), przez co konieczne jest wykorzystanie znacznie większej ilości danych, a proces trenowania (estymowania) jest bardziej wymagający pod względem potrzebnych zasobów. Mimo to, dzięki rozwojowi kompu-



Rysunek 2.3. Pojedynczy neuron typu LSTM. Oprócz wewnętrznego stanu h_t do kolejnego kroku przekazywana jest również zmienna c_t kontrolująca, czy obecne dane wejściowe powinny zostać zapamiętane czy zapomniane. Kontrolowanie zapominania podlega treningowi. Mechanizm ten zapobiega zastępowania starszych (lecz istotnych) informacji nowymi.
Źródło: Opracowanie własne.



Rysunek 2.4. Przepływ danych przez sieć typu LSTM. Przepływ danych jest analogiczny do przepływu zwykłej sieci RNN (rys. 2.2), jednak oprócz wewnętrznego stanu h_t przekazana zostaje również zmienna c_t.
Źródło: Opracowanie własne.

terów w ostatnich dekadach, sieci neuronowe stały się powszechne używane w wielu dziedzinach, w tym również w ekonometrii finansowej.

Stan rozwoju różnych typów ML (nie tylko sieci neuronowych) oraz ich porównania między sobą i z modelami klasycznej ekonometrii został opisany w dwóch pracach: publikacji Obthonga M. i in. z 2020 roku [137] i artykule Soni P. i in. z 2022 roku [166].

Wczesne udane próby zastosowania sieci neuronowych do prognozowania finansowych szeregów czasowych sięgają 2005 roku. Ich podsumowanie oraz bardziej szczegółowe porównanie z modelami GARCH opublikował w 2011 roku Guresen E. i in. [76]. Osiągnięte wyniki potwierdzają większą skuteczność predykcji indeksów giełdowych niż przy wykorzystaniu modeli GARCH.

Sieci neuronowe mogą być również wykorzystane do predykcji akcji konkretnych spółek. Chlebus M. i in. w 2021 roku opublikowali artykuł, w którym wykorzystywali modele hybrydowe do predykcji akcji Nvidia Corporation osiagając wyniki lepsze niż przy użyciu modeli typu ARIMA [24].

2.3. Podsumowanie

W ekonometrii finansowej zostały podjęte próby uwzględnienia strumienia nowych informacji w modelach stóp zwrotu. Najczęściej wykorzystywane są historyczne szeregi czasowe, np. w modelach ARIMA-GARCH, jednak współcześnie coraz częściej do modeli dodawane są dane egzogeniczne, które przybliżają efekty związane z reakcją inwestorów na nowe wiadomości.

Oprócz klasycznych modeli ekonometrii finansowej coraz częściej wykorzystywane są modele uczenia maszynowego, zwłaszcza rekurencyjne sieci neuronowe.

Rozdział 3

Współczesne techniki NLP oraz możliwości ich zastosowania do analizy informacji finansowych

Przetwarznie Języka Naturalnego - ang. *Natural Language Processing* (NLP) staje się coraz bardziej popularne w wielu dziedzinach nauki i zastosowaniach komercyjnych. Historia analizy języka naturalnego sięga IX wieku i z czasem jej użycie stało się wyjątkowo powszechne. Ze względu na mnogość celów, jakie można stawiać przed systemem NLP, wyróżnia się odrębne *zadania*, które zwykle są realizowane przez niezależne modele.

Najbardziej współczesne modele zwykle oparte są o głębokie sieci neuronowe, do których stworzenia wykorzystano bardzo duże ilości danych tekstowych. Następnie części tych modeli są wykorzystywane wielokrotnie w procesie *fine-tuningu*. W szczególności tak stworzone sieci można wykorzystać w analizach ekonometrycznych.

3.1. Historia rozwoju NLP

3.1.1. Początki - tłumaczenie maszynowe i systemy regułowe

Pierwsze próby automatycznego przetwarzania języka naturalnego zostały odnotowane już w IX wieku. Najwcześniej napisane manuskrypty, których autorem był arabski uczony al-Kindi, dotyczyły kryptografii, kryptoanalizy, tłumaczenia maszynowego, analizy statystycznej oraz częstotliwościowej tekstów pisanych. Jednym z najszerzej znanych jego dzieł jest traktat „O odczytywaniu zaszyfrowanych listów", w którym autor kładzie nacisk na wykorzystywanie tekstów na tyle długich, aby posiadały wystarczająco dużą statystykę. Traktat ten został zrekonstruowany i wydany ponownie w 2003 roku [95].

Zainteresowanie tłumaczeniem maszynowym pojawiło się ponownie w XVII wieku wśród europejskich filozofów i naukowców. W 1629 roku René Descartes zaproponował stworzenie uniwersalnego języka, który będzie składał się z jednakowych symboli przed-

stawiających te same idee w różnych językach. Myśl ta była kontynuowana przez wielu znanych uczonych, między innymi Newtona czy Leibniza. Jednym z podstawowych celów stworzenia takiego języka było zastąpienie łaciny i umożliwienie sprawniejszego komunikowania się oraz tłumaczenia tekstów [107].

W latach 30. XX wieku zaczęły pojawiać się patenty maszyn tłumaczących. Jedne z pierwszych prototypów takiej maszyny zostały zaprojektowane przez Artsrouni'ego G.. Wykorzystywał w nich taśmę perforowaną do automatycznego (mechanicznego) tłumaczenia pomiędzy dwoma językami. Dalsze próby (między innymi Troyanskiego P.) pozwalały na tłumaczenie nie tylko słów między językami, ale także ich gramatyki, bazując na zasadach języka esperanto. Tłumaczenie odbywało się w trzech krokach: najpierw rodzimy użytkownik języka źródłowego normalizował tekst w postaci form logicznych, następnie maszyna tłumaczyła je do języka docelowego, a na końcu rodzimy użytkownik języka docelowego korygował otrzymany tekst [90].

Pierwsze projekty maszyny tłumaczącej, będącej komputerem, pochodzą z 1949 roku, a ich pomysłodawcą był Weaver W. Proponowana przez niego metodologia opierała się o teorię informacji i osiągnięcia w kryptografii pochodzące z czasów II Wojny Światowej [190].

Jednym z najbardziej znanych testów poziomu rozwoju systemów sztucznej inteligencji jest Test Turinga, który opiera się właśnie o zaawansowane metody NLP. W 1950 roku A. Turing opublikował artykuł, w którym opisał kryterium inteligencji systemu bazujące na tym, czy w konwersacji tekstowej z człowiekiem jest w stanie wystarczająco dobrze naśladować żywą osobę. W przypadku, gdy człowiek nie jest w stanie rozstrzygnąć czy rozmawia z komputerem, czy innym człowiekiem, można powiedzieć, że pomyślnie przeszła test Turinga [179].

Kilka lat po publikacji Weavera W. [190] na wielu uniwersytetach w Stanach Zjednoczonych wzrosło zainteresowanie tłumaczeniem maszynowym, zaś nowe osiągnięcia pozwoliły na pozyskanie środków na badania. Jednym z najbardziej znanych wydarzeń tego okresu jest Eksperyment Georgetown-IBM, który odbył się 7 stycznia 1954 roku. Eksperyment ten był prezentacją całkowicie zautomatyzowanej maszyny tłumaczącej zdania z języka rosyjskiego na angielski. W prezentacji wykorzystano sześćdziesiąt wyselekcjonowanych zdań. System wykorzystywał jedynie sześć reguł gramatycznych i 250 pozycji w słowniku, zaś domeną, w której pracował, była chemia organiczna. Do wykonywania obliczeń użyto komputera IBM 701, a zdania musiały być wprowadzone

na taśmie perforowanej. Jednym z głównych celów eksperymentu było przyciągnięcie inwestorów i zebranie środków na prowadzenie dalszych badań [91, 92].

W 1957 roku Chomsky N. (lingwista), przedstawił ideę gramatyki generatywnej, która tłumaczy zjawisko polegające na tym, że każdy mówiący w danym języku jest zdolny do tworzenia zdań, których nigdy dotąd nie konstruował, zaś słuchający może zrozumieć zdania, których nigdy dotąd nie słyszał. Gramatyka generatywna jest skończonym zbiorem reguł, które pozwalają budować zrozumiałe zdania [25]. Późniejsze badania wykazały, że nie wszystkie wnioski Chomsky'ego N. są w mocy, jednak przez kilka kolejnych dekad miały silny wpływ na rozwój lingwistyki, a także NLP.

W latach 60. i 70. powstało wiele praktycznych zastosowań NLP, głównie systemów dialogowych. Jednym z nich była ELIZA (symulator psychoterapeuty powstały na Massachusetts Institute of Technology), która udzielała bardzo ogólnych odpowiedzi na zadawane pytania [193]. Bardziej zaawansowanym podejściem był PARRY (symulator osoby chorej na schizofrenię stworzony na Stanford University) [29].

Wykorzystanie systemów regułowych było kontynuowane do końca lat 80. W tym czasie powstał m.in. algorytm Leska (służący do ujednoznaczniania słów), wiele implementacji gramatyki generatywnej, a także komercyjnych systemów dialogowych.

3.1.2. Podejście statystyczne

U schyłku lat 80. wpływ teorii Chomsky'ego N. osłabł, zaś wzrosła popularność podejścia statystycznego. Modelowanie języka naturalnego coraz częściej opierano na dużych zbiorach tekstów (korpusach językowych), których dostępność rosła. Również przetwarzanie tak znacznych ilości tekstu stało się możliwe dzięki coraz większej mocy obliczeniowej komputerów.

Tłumaczenie maszynowe nadal było jedną z wiodących dziedzin NLP ze względu na oczywistą możliwość wykorzystania praktycznego. Jedna z grup badawczych IBM pod kierownictwem M. Nagao rozwijała tłumaczenie maszynowe oparte o przykłady. Metoda ta całkowicie ignorowała zasady gramatyczne (syntaktyczne i semantyczne) i polegała wyłącznie na odpowiedniej liczbie przykładów [134].

Wszelkie podejścia statystyczne do tłumaczenia wymagały dużych równoległych korpusów w kilku językach. W latach 90. Parlament Kanady oraz Unia Europejska zarządziły, że wszelkie oficjalne teksty muszą być dostępne we wszystkich językach urzędowych. Dzięki temu powstała idealna okazja do rozwijania systemów tłumacze-

nia. Mimo iż treść korpusów była bardzo specjalistyczna, ich dostępność przeważyła o użyteczności.

Statystyczne metody NLP miały istotną zaletę względem podejścia regułowego - były znacznie bardziej odporne na niedoskonałe dane wejściowe. Język używany na co dzień jest dużo bardziej zanieczyszczony niż perfekcyjnie przygotowane zdania w lingwistyce regułowej np. pojawiają się powtórzenia, błędy gramatyczne wprowadzone przez człowieka, cofnięcia, przerwy w trakcie mówienia. W dziedzinie rozpoznawania mowy dodatkowo pojawiają się zakłócenia dźwiękowe, np. przez odgłosy otoczenia, zmianę dystansu od osoby mówiącej czy jednoczesną mowę kilku osób. W przypadku podejścia statystycznego można już na etapie trenowania modeli wprowadzać tego typu zakłócenia, aby uodpornić na nie system.

Po roku 2000 Internet był już powszechnie dostępny, zaś ilość ogólnodostępnego tekstu była tak duża, że niemożliwe było opisywanie go ręcznie, a tym samym uczenie nadzorowane. Spowodowało to zwiększenie nacisku na metody nienadzorowane, wykorzystujące ogólne właściwości tekstu, np. uzupełnianie luk w tekstach, z których zostały usunięte losowe słowa.

W tym czasie pojawiło się wiele komercyjnych projektów, z których część (po modyfikacjach, a często zaadaptowaniu podejścia wykorzystującego sieci neuronowe) funkcjonuje do dzisiaj:

— **Google Translate** (dawniej część Google Language Tools) - darmowy serwis umożliwiający tłumaczenie wybranych par języków,
— **AltaVista Babel Fish** - konkurencyjny serwis do tłumaczenia maszynowego, obecnie Microsoft Translator,
— **Siri** - wirtualny osobisty asystent stworzony przez firmę Apple, wbudowany w systemy operacyjne MacOS, iOS, iPadOS,
— **Cortana** - wirtualny asystent firmy Microsoft,
— **Alexa** - wirtualny asystent firmy Amazon,
— **Google Assistant** - wirtualny asystent firmy Google.

Wymienione serwisy, zwłaszcza w przypadku wirtualnych asystentów, wykorzystują jednocześnie wiele nurtów w NLP, bowiem składają się z systemu rozpoznawania mowy, tłumaczy maszynowych, systemów dialogowych, itp.

3.1.3. Sieci neuronowe

Jednym z przełomów w skutecznym przetwarzaniu języka naturalnego przez sieci neuronowe było opublikowanie nowego typu sieci LSTM przez Hochreitera S. i Schmidhubera J. w 1997 [84]. Zaproponowane rozwiązanie polegało na użyciu szczególnego typu neuronu w sieciach RNN, który w trakcie treningu uczył się, które informacje należy zapisać na dłużej, w przeciwieństwie do klasycznych sieci RNN posiadających stały współczynnik zastępowania starych informacji nowymi. Sieci RNN i LSTM zostały opisane dokładniej w rozdziale 2.2.

Już na początku XXI wieku sieci LSTM były szeroko wykorzystywane do analizy szeregów czasowych, a także języka naturalnego. Ich wadą była konieczność sekwencyjnego przetwarzania (co dawało efekt podobny do trenowania sieci z bardzo dużą liczbą warstw), a przede wszystkim problemy ze wstępnym uczeniem sieci (ang *pre-training*) na dużym, łatwo dostępnym zbiorze danych, a następnie użycia takiej sieci do nauczenia się na docelowym zbiorze danych (uczenie transferowe - ang. *Transfer Learning* (TL)). Taka technika wyjątkowo dobrze sprawdza się w sieciach splotowych (konwolucyjnych), służących do rozpoznawania obrazu.

Kilka lat później w uczeniu maszynowym zaczęły przeważać *głębokie* sieci neuronowe, tj. sieci o dużej liczbie warstw. Wykorzystanie ich w dziedzinie NLP przyniosło wiele skutecznych modeli językowych, a także metod parsowania tekstu [68, 69, 100]. W tym czasie rozwijały się również najbardziej popularne obecnie narzędzia do pracy z przetwarzaniem języka - NLTK [120] oraz spaCy [85].

Wprowadzenie modelu BERT (model oparty o dwukierunkowe enkodery z siecią typu transformer - ang. *Bidirectional Encoder Representations from Transformers*) z mechanizmem uwagi przez zespół badawczy Google Brain w 2017 [183] pozwoliło na wstępne trenowanie modeli językowych na ogromnych ilościach dostępnych tekstów, a następnie „douczenie" ich na mniejszych, lekko różniących się zbiorach danych. Sieci BERT są współcześnie używane do najbardziej zaawansowanych zadań NLP – automatyczne streszczanie, odpowiadanie na otwarte pytania na podstawie danego tekstu, a także maszynowe tłumaczenie dowolnej pary języków. Mechanizm uwagi pozwala na dobre zrozumienie zdania nawet, gdy semantycznie powiązane ze sobą wyrażenia położone są daleko od siebie. Dokładniejszy opis działania mechanizmu uwagi znajduje się w rozdziale 4.3.

Obecnie modele z mechanizmem uwagi są powszechne w większości zastosowań NLP. Jeden z najnowszych trendów polega na wykorzystaniu bardzo dużych gene-

ratywnych sieci z mechanizmem uwagi do stworzenia modelu ogólnego zastosowania, którego dane wejściowe i wyjściowe składają się z tekstu zrozumiałego przez człowieka. Jednym z takich modeli jest GPT-3 opublikowany w 2020 roku przez badaczy z grupy OpenAI [16]. Model ten składa się z około 175 miliardów parametrów i został wytrenowany na tekście zebranym automatycznie z olbrzymiej ilości stron internetowych. Kolejne wersje tego modelu to ChatGPT [139], posiadający 1,5 miliarda parametrów (zmniejszenie modelu było możliwe dzięki zastosowaniu nowej architektury) oraz GPT-4 [140, 141] zbudowany z 170 bilionów parametrów.

Większość bieżących badań prowadzonych jest przez komercyjne spółki technologiczne, jednak często ich wyniki publikowane są na licencjach typu *Open Source*. Poza samą treścią publikacji zwykle dostępny jest kod źródłowy potrzebny do odtworzenia wyników, a także pliki z wagami wstępnie wytrenowanych modeli. Dzięki takiemu podejściu badania mogą postępować znacznie szybciej, a trenowanie własnych modeli językowych ogólnego przeznaczenia, które jest wyjątkowo kosztowne, nie jest koniecznym krokiem dla każdego badacza.

3.2. Zadania NLP

Zastosowania NLP można skategoryzować ze względu na realizowane *zadanie* T, gdzie *zadanie* oznacza odpowiedni sposób przewidywania lub wnioskowania na podstawie danych. Typy *zadań* zostały w formalny sposób opisane w rozdziale 2.2. Niektóre z *zadań* mogą wydawać się naturalne dla człowieka, jednak w przypadku komputerowej implementacji muszą zostać rozbite na mniejsze części. Na przykład na *ekstrakcję informacji z tekstu pisanego* może składać się:

— klasyfikacja zdań ze względu na znaczenie,
— rozpoznawanie jednostek referencyjnych - ang. *Named Entity Recognition* (NER),
— analiza sentymentu.

Wybór wszystkich pod-zadań realizowanych przez system jest zawsze uzależniony od celu, w jakim jest on tworzony, a także od dostępnych danych wejściowych. Gdyby posiadany zbiór danych składał się z zeskanowanych dokumentów (obrazów), dodatkowo konieczne byłoby wstępne optyczne rozpoznawanie pisma - ang. *Optical Character Recognition* (OCR).

Zadania NLP w ogólności nie zmieniły się od początków rozwoju tej dziedziny. Nadrzędnym celem wciąż jest szeroko pojęte rozumienie tekstu - ekstrakcja informacji, tłumaczenie tekstu na inny język czy synteza mowy. Jednak szczegółowe podejście do rozwiązania tych problemów (rozbicie na konkretne pod-zadania) zmieniało się wraz z głównym nurtem badań. W przypadku systemów regułowych z lat 60. i 70. kluczowymi etapami były: oznaczanie części mowy, analiza leksykalna, wykorzystywanie słów kluczowych. Obecne rozwiązania skupiają się na przetwarzaniu całościowo dłuższych fragmentów tekstu i istotniejsze są neuronowe reprezentacje języka, NER oraz wykorzystanie mechanizmu uwagi.

3.2.1. Reprezentacja tekstu

Podstawowym zagadnieniem, bez którego nie jest możliwa dalsza praca z NLP, jest wybór sposobu reprezentacji języka. Procesory komputerowe są zdolne do przetwarzania jedynie liczb binarnych, lecz dzięki zastosowaniu pewnych abstrakcji, pośrednio również liczb całkowitych i zmiennoprzecinkowych. Wykonywanie operacji arytmetycznych we współczesnych komputerach zostało silnie zrównoleglone, zatem możliwe jest prowadzenie obliczeń na wektorach, macierzach i tensorach wyższych rzędów.

Architektura komputerów była rozwijana, umożliwiając coraz szybsze obliczenia. Najbardziej powszechne układy do obliczeń to:

— (centralna jednostka przetwarzająca - ang. *Central Processing Unit* (CPU) - układ pozwalający na sekwencyjne wykonywanie operacji,
— (procesor graficzny - ang. *Graphics Processing Unit* (GPU) - układ zaprojektowany do szybkiego (równoległego) mnożenia i dodawania macierzy, szczególnie przydatnego w generowaniu trójwymiarowej grafiki,
— (procesor tensorowy - ang. *Tensor Processing Unit* (TPU) - układ zaprojektowany w 2016 roku przez firmę Google, specjalnie w celu równoległego mnożenia i dodawania tensorów.

Mimo tak zaawansowanego rozwoju, komputery wciąż nie są zdolne do przetwarzania języka (w tym tekstu pisanego) w taki sposób, jak ludzie. Znaki (litery, cyfry, znaki interpunkcyjne i specjalne) są kodowane w postaci binarnej, co pozwala na przechowywanie ich w pamięci komputera. Jednak nawet umiejętność zapisu nie oznacza, że komputer „rozumie" zapamiętany tekst lub chociaż potrafi go w jakiś sposób przetworzyć.

W zdecydowanej większości zastosowań pierwszym krokiem jest wybór słów, które trafią do modelu języka. Zwykle słowa błędnie zapisane (z błędami ortograficznymi czy literówkami) w ogóle nie trafiają do modelu. W podejściach statystycznych podczas tworzenia reprezentacji języków fleksyjnych najczęściej konieczna jest lematyzacja słów - pozbycie się ich części fleksyjnej. Np. dla języka polskiego wszystkie rzeczowniki są sprowadzane do mianownika. Dzięki tym operacjom, kosztem utraty części informacji zawartych w gramatyce, możliwe jest znaczne zredukowanie liczby wszystkich słów, które należy zapamiętać. Każde pozostałe unikalne słowo to *token*, zaś operacja przetwarzania nieobrobionego tekstu na ciąg tokenów to *tokenizacja*. W niektórych zastosowaniach nazwy własne lub innego typu frazy mogą również być traktowane jako indywidualne tokeny. Tokenizacja stanowi interesujący i szeroki problem, którego rozwiązanie jest zależne od języka, w którym napisany jest tekst oraz nadrzędnego *zadania* T i została opisana w licznych publikacjach, np. [74, 191].

W celu umożliwienia komputerowi pracy na tekście (już jako zbiorze słów, a nie znaków), każdemu tokenowi można przypisać zmienną liczbową. Wykorzystanie liczb porządkowych nie ma większego sensu ze względu na brak naturalnego znaczeniowego porządku słów. Z tego powodu zaczęły rozwijać się metody reprezentacji wektorowej.

Jedna z najbardziej uproszczonych metod reprezentacji tekstu jako wektora to Worek Słów - ang. *Bag of Words* (BoW). Metoda ta polega na zliczaniu wystąpień tokenów w dokumencie. Nazwa została pierwszy raz użyta przez Harrisa Z. w artykule, w którym krytykował tego typu spojrzenie na język naturalny, ponieważ podejście to nie uwzględnia struktury tekstu [80].

Stworzenie reprezentacji BoW w pierwszym kroku polega na wylistowaniu wszystkich indywidualnych tokenów występujących we wszystkich dokumentach. Każdy z tokenów stanowi wymiar wektora tej reprezentacji. Następnie wektory przedstawiające każdy z dokumentów zostają utworzone poprzez zanotowanie liczby wystąpień każdego z tokenów.

Dla przykładu mając trzy dokumenty:

1. „Anna ma nowy rower.”,
2. „Bartek ma samochód.”,
3. „Bartek ma samochód. Samochód jest nowy.”,

można stworzyć ich reprezentację wektorową BoW zaczynając od utworzenia listy wszystkich tokenów: [*”Anna”, ”Bartek”, ”jest”, ”ma”, ”nowy”, ”rower”, ”samochód”,*

"."]. W kolejnym kroku dokonywane są zliczenia wystąpień, jak to przedstawiono w Tabeli 3.1.

Tabela 3.1. Przykład zliczania tokenów w dokumentach.

	Anna	Bartek	jest	ma	nowy	rower	samochód	.
Dokument 1.	1	0	0	1	1	1	0	1
Dokument 2.	0	1	0	1	0	0	1	1
Dokument 3.	0	1	1	1	1	0	2	2

Źródło: Opracowanie własne.

Wówczas dokumenty są reprezentowane przez następujące wektory:

$$\begin{aligned} \mathbf{v_1} &= [1,0,0,1,1,1,0,1], \\ \mathbf{v_2} &= [0,1,0,1,0,0,1,1], \\ \mathbf{v_3} &= [0,1,1,1,1,0,2,2]. \end{aligned} \tag{3.1}$$

Metoda BoW posiada wady, które utrudniają wykorzystanie jej w modelach ML. Wymiarowość wektorów wzrasta wraz z ilością wykorzystanych w dokumentach tokenów - dla większości współczesnych języków zapisanie zdania czy dokumentu dawałoby w efekcie bardzo rzadki wektor o wymiarze ponad 100 000. Metodę BoW można postrzegać jako zagregowane (zsumowane) wektory *one-hot* (kod 1 z n) reprezentujące każdy z tokenów z osobna.

Dodatkową wadą jest ortogonalność wszystkich tokenów bez względu na ich znaczenie. Reprezentacja wyrazów bliskoznacznych nie daje żadnej informacji o ich powiązaniu, co dodatkowo utrudnia interpretację dłuższych tekstów. Jedną z metod rozwiązywania tych problemów były próby redukcji wymiarowości wektorów otrzymanych przy użyciu BoW. Mimo zredukowania liczby wymiarów, wektory wciąż nie odwzorowywały w odpowiedni sposób relacji między tokenami.

Przełomem było zastosowanie *word embeddings*. Technika ta polega na ustaleniu z góry wymiarowości N reprezentacji języka, przy czym zwykle $N \in [10, 1000]$, a następnie przypisanie każdemu tokenowi wektora o wymiarze N z **losowymi** współrzędnymi. Kolejny krok to stworzenie sieci neuronowej uczonej w sposób nienadzorowany na dużym zbiorze łatwo dostępnych danych. Najczęściej wykorzystuje się zbiór artykułów Wikipedii lub ogólnie zbiór treści stron internetowych, np. CommonCrawl, zaś zadaniem sieci neuronowej jest uzupełnianie wyciętych z tekstu słów [31, 198]. Zadanie uzupełniania tekstu zostało dokładniej opisane w podrozdziale 3.2.2. Podczas treningu

współrzędne wektorów przypisanych do każdego tokenu traktowane są jako wagi sieci, dzięki czemu mogą się zmieniać tak, aby model osiągnął jak największe dopasowanie do danych [101, 126].

Przez pracę w takim reżimie współrzędne wektorów osiągają wartości, które pozwalają na reprezentację tokenów poprzez **kontekst**, w jakim występują. Otrzymane w ten sposób wektory mają wiele istotnych właściwości:

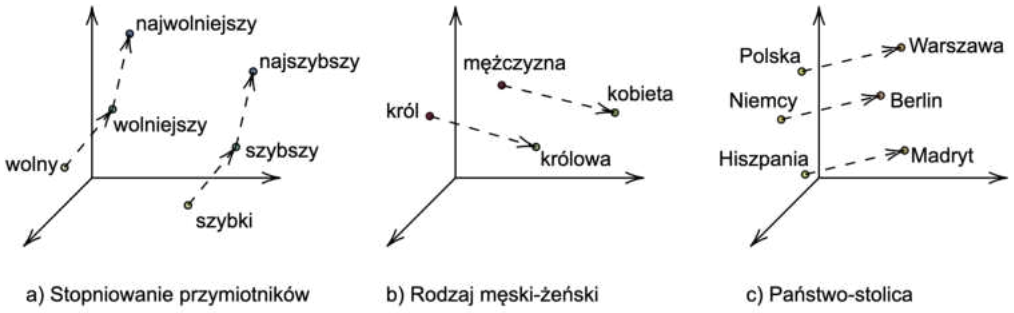
1. Wymiarowość jest znacznie ograniczona i może być z góry ustalona,
2. Tokeny o podobnym znaczeniu grupują się - dystans liczony przy pomocy klasycznych metryk wewnątrz stworzonej N-wymiarowej przestrzeni jest mniejszy dla wyrazów bliskoznacznych lub o podobnych funkcjach,
3. Możliwe jest wykonywanie operacji algebraicznych na otrzymanych wektorach, które zachowują znaczenie reprezentowanych pojęć.

Trzecia właściwość tych wektorów jest szczególnie interesująca z lingwistycznego punktu widzenia. W pełni wytrenowane *word embeddings* pozwalają na sprawdzanie relacji między słowami. Możliwe jest np. zaobserwowanie następującej relacji: token „wolny" ma się do tokenu „wolniejszy", tak jak token „szybki" ma się do tokenu „szybszy". W przestrzeni wektorowej relacja ta polega na sumowaniu i odejmowaniu wektorów. Następnie należy znaleźć token znajdujący się w przestrzeni najbliżej wynikowemu wektorowi. Operacje te można przedstawić w formie następującej zależności:

$$\mathbf{v_{we}}(wolniejszy) - \mathbf{v_{we}}(wolny) + \mathbf{v_{we}}(szybki) \approx \mathbf{v_{we}}(szybszy) \tag{3.2}$$

Reprezentacja *word embeddings* pozwala zaobserwować zależności takie jak stopniowanie przymiotników, rodzaj gramatyczny, relacje państwo-stolica i wiele innych [125]. Przykładowe relacje przedstawia Rysunek 3.1.

Reprezentacja *word embeddings* może być rozszerzona o informację o pozycji danego tokenu w dokumencie, co jest konieczne w celu wykorzystania mechanizmu uwagi w najnowszych implementacjach sieci neuronowych typu Transformer [183]. Zwykle informacja o pozycji tokenu wewnątrz dokumentu jest kodowana przy pomocy funkcji harmonicznych. Otrzymane w ten sposób wektory *positional embeddings* pozwalają na przetwarzanie wszystkich tokenów jednocześnie, co znacznie przyspiesza obliczenia względem sieci rekurencyjnych.



Rysunek 3.1. Wizualizacja przykładowych relacji wektorów *word embeddings* zrzutowanych na przestrzeń trójwymiarową. Rzutowanie na trójwymiarową przestrzeń pomaga przedstawić relacje w uproszczeniu. Właściwe wartości są wielowymiarowymi wektorami. a) Stopniowanie przymiotników zostało odwzorowane jako translacja o wektor o tym samym zwrocie, kierunku i wartości. Dodanie tego wektora do reprezentacji dowolnego przymiotnika w stopniu równym daje w efekcie reprezentację tego przymiotnika w stopniu wyższym. b) Dla rzeczowników określających osoby, zawody, itp. przejście między rodzajem męskim i żeńskim jest reprezentowane jako translacja o taki sam wektor. c) Odnalezienie stolicy danego państwa w przestrzeni *word embeddings* również polega na translacji o taki sam wektor.
Źródło: Opracowanie własne.

3.2.2. Uzupełnianie tekstu

Uzupełnianie tekstu to zadanie, w którym model stara się dopasować słowo lub słowa usunięte z tekstu na podstawie pozostałej jego części. Mimo niewielu praktycznych zastosowań tego typu modeli, okazały się one bardzo przydatne jako krok pośredni w tworzeniu reprezentacji językowych.

Największa zaletą tej metody jest wyjątkowo duża dostępność danych treningowych. Uczenie modelu odbywa się w sposób nadzorowany, jednak zbiór danych może zostać wygenerowany automatycznie. Podstawą w tym przypadku jest duży korpus w danym języku, który można łatwo pozyskać poprzez *crawling* stron internetowych - automatyczne pobranie i zapisanie ich treści. Ze względu na praktycznie dowolny układ stron internetowych, konieczne może być wstępne oczyszczenie danych, jednak wiele gotowych zbiorów jest udostępnianych za darmo np. przez fundację Common Crawl Foundation [31].

Kolejnym krokiem jest usunięcie z tekstu pojedynczych tokenów i pozostawienie w ich miejscu luki. Wówczas, korzystając z definicji uczenia nadzorowanego w rozdziale 2.2:

1. Usunięte tokeny $\mathbf{y} \in \mathcal{Y}$ są traktowane jako oczekiwane dane wyjściowe z modelu,

2. Tokeny otaczające pozostałą w tekście lukę $\mathbf{x} \in \mathcal{X}$ (kontekst) to dane wejściowe modelu,
3. Zadaniem T jest przewidzenie tokenu $\mathbf{y}$ na podstawie jego kontekstu $\mathbf{x}$.

Wektory $\mathbf{x}$ i odpowiadające im wektory $\mathbf{y}$ tworzone są poprzez okienkowanie całego dostępnego korpusu, aby zapewnić jak najlepsze wykorzystanie dostępnych danych. Przykład tworzenia zbioru danych treningowych z długością okna równą 3 jest przedstawiony w tabeli 3.2. Model wytrenowany w ten sposób to Ciągły Worek Słów - ang. *Continuous Bag of Words* (CBOW) i został zaproponowany przez Mikolova T. i in. w celu tworzenia ciągłych reprezentacji języka [125, 126].

Tabela 3.2. Przykład tworzenia zbioru danych uczących dla modelu CBOW.

Tekst	**x**	**y**
"**Anna wczoraj kupiła** nowy rower."	["Anna", "kupiła"]	"wczoraj"
"Anna **wczoraj kupiła nowy** rower."	["wczoraj", "nowy"]	"kupiła"
"Anna wczoraj **kupiła nowy rower**."	["kupiła", "rower"]	"nowy"
"Anna wczoraj kupiła **nowy rower**."	["nowy", "."]	"rower"

Źródło: Opracowanie własne.

Implementacja modelu CBOW zakłada przypisanie każdemu tokenowi wektora *word embeddings* inicjalizowanego losowo, którego wagi dopasowywane zostają w procesie trenowania. Jeżeli wykorzystany korpus jest wystarczająco duży, otrzymana reprezentacja *word embeddings* może zostać „zamrożona" i wykorzystana jako pierwsza warstwa innego modelu NLP realizującego zadanie T', dzięki czemu proces treningu będzie krótszy. Jest to szczególnie przydatne w przypadku, gdy dane posiadane dla zadania T' są trudniej dostępne.

Istnieją również inne metody trenowania modeli w celu otrzymania reprezentacji języka, np. wykorzystujące informacje zawarte w układzie znaków wewnątrz słów lub korzystające z wektorów $\mathbf{x}$ i $\mathbf{y}$ odwrotnie do modelu CBOW (model *skip-gram*). Każda z tych metod daje nieco odmienne wyniki, jednak zasada, na której bazują, jest jednakowa [11, 73].

Wektory *word embeddings* wstępnie wytrenowane metodą uzupełniania tekstu są udostępniane przez wiele zespołów badawczych w wersji *Open Source* i można je pobrać np. z repozytoriów TensorFlow Hub lub Hugging Face [1, 89].

3.2.3. Ekstrakcja informacji zawartych w tekście

Ekstrakcja informacji jest wysokopoziomowym zadaniem NLP, które polega na odnalezieniu w danym tekście *ważnych* informacji, a następnie przekazanie ich dalej w *skróconej formie*. To, które informacje są *ważne*, jest arbitralnie ustalone w procesie uczenia modelu, np. podsumowanie całego tekstu lub położenie jednostek referencyjnych. Również forma, w jakiej wyekstrahowane informacje są przekazane, może zostać ustalona na wiele sposobów. W przypadku, gdy odbiorcą jest człowiek, może to być tekst (podsumowanie) lub tabela (jednostki referencyjne i ich pozycja), zaś gdy informacje mają podlegać dalszej obróbce, bardziej korzystne może być przedstawienie ich w postaci wektora liczbowego. Ekstrakcja informacji może być rozumiana bardzo szeroko i została opisana w licznych publikacjach, np. [98, 164].

Enkodowanie informacji w postaci wektora może być szczególnie przydatne w przypadku wykorzystania uczenia transferowego opisanego w rozdziale 4.4.2. Większość modeli ekonometrycznych oraz ogólnie modeli ML wymaga danych wejściowych o określonej wymiarowości. Informacje przekazywane w formie tekstu pisanego mogą mieć zmienną długość i być niemożliwe do zinterpretowania przez modele działające wyłącznie na danych liczbowych.

Metody ekstrakcji informacji zawartych w tekście wiadomości o tematyce finansowej i wykorzystanie ich do prognozowania cen aktywów emitowanych przez podmioty w nich opisane są głównym przedmiotem badań w niniejszej pracy.

3.2.4. Analiza sentymentu

Analiza sentymentu (lub *opinion mining*) to szczególny przypadek ekstrakcji informacji zawartych w tekście. Polega na ustaleniu emocji lub subiektywnego nastawienia autora tekstu do omawianej przez niego tematyki. Przykładowe zastosowania to badanie satysfakcji klientów na podstawie ich recenzji, ocena poparcia partii politycznych na podstawie treści publikowanych w mediach społecznościowych, a także analiza nastawienia panującego wśród inwestorów do różnego typu inwestycji.

Pierwsze próby analizy sentymentu przeprowadzano na MIT już w 1969 roku. Stone P. wraz z zespołem badali emocje zawarte w tekstach [170]. Późniejsze zastosowania objęły między innymi analizę recenzji produktów w celu zbadania zadowolenia klientów [180]. Na początkowych etapach wykorzystywano słowa kluczowe i proste modele ML (klasyfikatory Bayesowskie, maszyny wektorów nośnych), jednak współcześnie naj-

częściej wykorzystywane są sieci neuronowe typu LSTM (patrz rozdział 2.2) i sieci z mechanizmem uwagi.

Analiza sentymentu może być również zastosowana do predykcji notowań giełdowych. W sytuacji, gdy inwestorzy są optymistycznie nastawieni, zwykle ceny instrumentów rosną, zaś gdy inwestorzy sę negatywnie nastawieni do sytuacji na rynku, wyprzedają aktywa i ceny spadają. Analiza sentymentu w finansach bazuje na bahawioralnym aspekcie decyzji inwestycyjnych, tzn. poddawaniu się emocjom przez uczestników rynku. Prognozowanie kierunku zmian cen na podstawie opinii wyrażanych przez inwestorów daje pozytywne wyniki [130, 144]. Nastawienie do konkretnych spółek może być badane na podstawie publikacji w mediach społecznościowych, na blogach lub oficjalnych serwisach informacyjnych o tematyce finansowej i gospodarczej.

3.2.5. Pozostałe zadania NLP

Istnieje wiele innych zadań realizowanych w dziedzinie przetwarzania języka naturalnego. Skuteczną metodą odnajdywania informacji w tekście jest identyfikacja jednostek referencyjnych. W przypadku tekstów o tematyce finansowej jednostkami referencyjnymi mogą być nazwy spółek, symbole giełdowe, nazwiska członków zarządu i znanych pracowników.

Question answering to zadanie, w którym dane wejściowe i dane wyjściowe to tekst zrozumiały przez człowieka, a model uczy się odpowiadać na zadane mu pytania. Również w tym przypadku wymagane jest dobre zrozumienie tekstu przez model, jednak w przypadku modeli ekonometrycznych, odpowiedź w postaci tekstu nie jest wygodna w celu dalszego jej przetwarzania.

Dosyć podobne w działaniu jest zadanie streszczania tekstu - dane wyjściowe również są w postaci tekstu zrozumiałego przez człowieka, a celem modelu jest napisanie streszczenia otrzymanego na wejściu dłuższego tekstu, nie gubiąc przy tym jego sensu i najistotniejszych informacji.

W wielu praktycznych zastosowaniach ten sam tekst podlega analizie przeprowadzanej przez kilka modeli wytrenowanych pod kątem różnych zadań - np. klasyfikacja całego dokumentu oraz oznaczanie jednostek referencyjnych. Otrzymane w ten sposób dane mogą być łączone i wykorzystywane dalej przez odpowiednią obróbkę otrzymanych informacji.

W przypadku, gdy tekst jest jednym z wielu typów danych wejściowych modelu (lub

większego algorytmu), zwykle analizowany jest całościowo przy użyciu sieci neuronowej enkodującej informacje zawarte w tekście.

3.3. Współczesne osiągnięcia w przetwarzaniu języka naturalnego

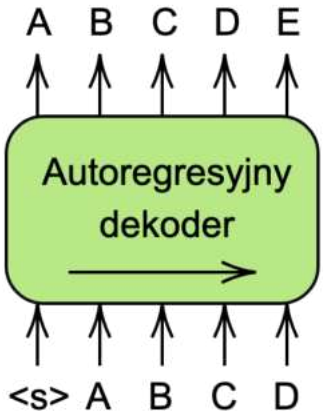
Obecnie modele oparte o NLP coraz częściej bazują na wykorzystaniu uczenia transferowego. W tym celu wykorzystuje się dostępne, wstępnie wytrenowane (i zwykle duże) modele NLP, które były uczone do rozwiązywania zadania T_1, a następnie modyfikuje się lub wzbogaca o dodatkową strukturę i trenuje na zadaniu T_2. Uczenie transferowe, mimo coraz częstszego zastosowania w modelach ML, a w szczególności w komputerowym przetwarzaniu obrazów, w modelach NLP zyskało popularność dopiero wraz z wprowadzeniem mechanizmu uwagi [183], opisanego w rozdziale 4.3. Sieci neuronowe mogą składać się z wielu równoległych członów wykorzystujących mechanizm uwagi, dzięki czemu stworzone za ich pomocą reprezentacje językowe są bardziej ekspresywne.

3.3.1. Budowa modeli językowych

Jako bazę do tworzenia modeli NLP wykorzystuje się reprezentacje językowe oparte o duże sieci neuronowe z mechanizmem uwagi wytrenowane w sposób nienadzorowany na możliwie jak największych korpusach. Otrzymana w ten sposób architektura to model językowy - ang. *Language Model* (LM). Istnieją obecnie trzy najbardziej powszechne klasy modeli językowych różniące się zadaniem T_1 wykorzystywanym do ich wytrenowania [127]:

— autoregresyjne modele językowe,
— model językowy z maskowaniem,
— model językowy typu enkoder-dekoder.

Modele autoregresyjne są trenowane w celu przewidywania kolejnego słowa x_i na podstawie wszystkich poprzednich słów $x_0, x_1, ..., x_{i-1}$, a tym samym są jednokierunkowe. Schemat działania modelu autoregresyjnego został przedstawiony na rysunku 3.2 - sekwencja na dole rysunku to szereg kolejnych tokenów pojawiających się na wejściu modelu, zaś powyżej przedstawiona jest oczekiwana odpowiedź. Zadaniem T_1 modelu jest znalezienie parametrów dekodera typu Transformer θ_T metodą największej wiarygodności $\Sigma_i log(P(x_i|x_0, x_1, ..., x_{i-1}); \theta_T)$. Przykładem zastosowania tego podej-



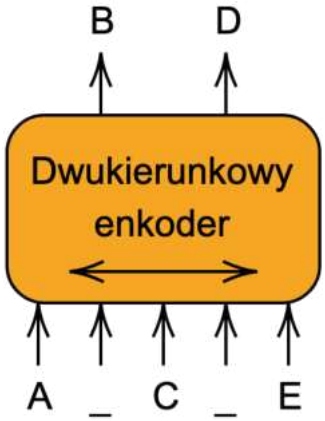
Rysunek 3.2. Autoregresyjny model językowy. Działanie tego typu modelu polega na przewidywaniu kolejnego słowa (tokenu) w sekwencji. Dużą zaletą modeli autoregresyjnych jest możliwość wykorzystania łatwo dostępnych danych do treningu. Duży korpus tekstu jest dzielony w taki sposób, aby danymi wejściowymi były sekwencje słów, a danymi wyjściowymi reprezentacja kolejnego słowa. Metodą przesuwającego się okna można automatycznie stworzyć tyle sekwencji uczących, ile wszystkich tokenów w korpusie.
Źródło: Opracowanie własne na podstawie [112].

ścia mogą być kolejne modele publikowane przez grupę badawczą OpenAI: GPT [153], GPT-2 [154] i GPT-3 [16]. Pierwszy model GPT osiągnął wyniki znacznie przewyższające wyniki obecnego wówczas czempiona w większości *benchmarków* (GLUE, CoLA).

Modele językowe z maskowaniem (rysunek 3.3) wykorzystują informacje z całego otoczenia słowa. Trening tych modeli polega na zamaskowaniu niektórych (zwykle losowo wybranych) słów w tekście, a następnie próbie "odgadnięcia" ich. Informacja dostępna dla modelu, to całe otoczenie zamaskowanego słowa, a zatem architektura musi być dwukierunkowa. Zadanie to również może być przedstawione jako odnalezienie parametrów $\hat{\boldsymbol{\theta}}_T$ metodą największej wiarygodności:

$$\hat{\boldsymbol{\theta}}_T = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}} \, \Sigma_i m_i log(P(x_i|x_0, ..., x_{i-1}, x_{i+1}, ..., x_n); \boldsymbol{\theta}_T), \tag{3.3}$$

gdzie $m_i \in 0, 1$ oznacza, czy słowo (token) x_i jest zamaskowane. Jednym z pierwszych modeli tego typu był BERT, z dodatkowym założeniem, że wiele słów jednocześnie może być zamaskowanych w celu zrównoleglenia treningu [39]. Dużą zaletą modeli tego typu jest brak autoregresyjności, dzięki czemu większa część obliczeń może być wykonywana

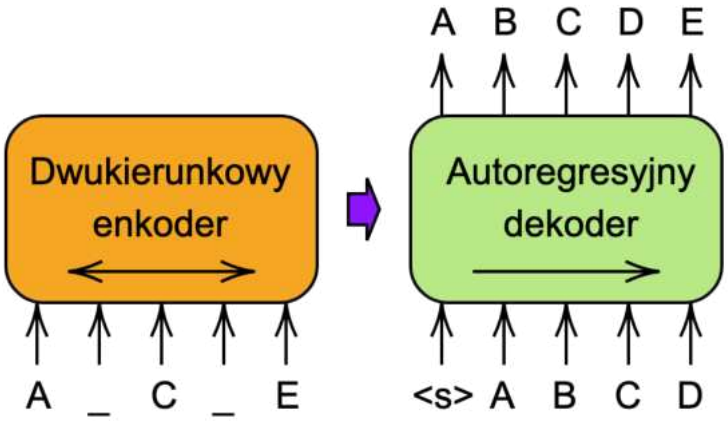


Rysunek 3.3. Model językowy z maskowaniem. Zadaniem tego typu modelu jest odgadywanie zamaskowanego słowa w tekście na podstawie sekwencji znajdującej się przed nim i po nim. Możliwe jest automatyczne stworzenie dużej ilości danych treningowych na podstawie korpusu tekstu poprzez maskowanie kolejnych słów i przesuwanie okna określającego sekwencje przed i po zamaskowanym słowie.
Źródło: Opracowanie własne na podstawie [112].

równolegle, na przykład z użyciem układów GPU lub TPU, a także rozproszonych klastrów. Na podstawie modelu BERT powstało wiele jego wersji z mniejszymi lub większymi zmianami w architekturze lub sposobie treningu, np. RoBERTa [118] czy AL-BERT [109].

Modele typu enkoder-dekoder, przedstawione na rysunku 3.4, uczą się generować całą sekwencję tokenów $y_0, ..., y_n$ w odpowiedzi na podaną sekwencję tokenów $x_0, ..., x_m$ na wejściu. Zadaniem modelu jest predykcja kolejności słów w zdaniu na podstawie słów podanych w zaburzonym porządku. Trening również odbywa się metodą największej wiarygodności dla $log(P(y_0, ..., y_n|x_0, ..., x_m); \theta_T)$, gdzie θ_T jest poszukiwanym wektorem parametrów. Zbiór danych treningowych może zostać stworzony na wiele sposobów - zwykle oczekiwana sekwencja wyjściowa jest tworzona poprzez różne zniekształcenia sekwencji wejściowej. Przykładowymi modelami powstałymi w ten sposób są BART [112] oraz T5 [155].

Wyjątkowo istotnym czynnikiem na etapie tworzenia modelu językowego jest wybór korpusu, na którym będzie trenowany. Zwykle w tym celu wybierany jest jak największy dostępny zbiór danych, pokrywający jak najogólniejszą tematykę. Kryterium to spełnia na przykład zbiór wszystkich artykułów z Wikipedii [198] lub zbiór powstały przez po-

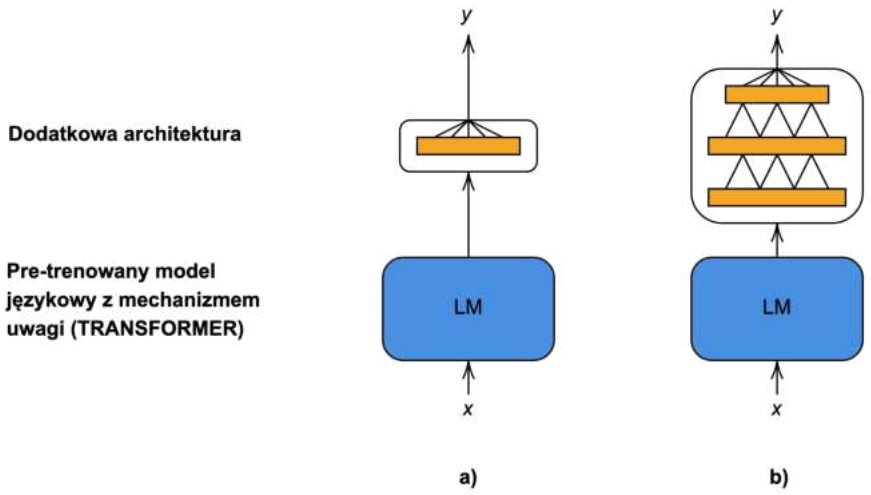


Rysunek 3.4. Model typu enkoder-dekoder. Zadaniem tego modelu jest generowanie całej sekwencji tokenów w odpowiedzi na sekwencję wejściową. Model musi przewidzieć kolejność słów w sekwencji podanej w zaburzonej kolejności. Dane treningowe mogą być wygenerowane automatycznie na podstawie dużego korpusu danych przez wprowadzanie różnych zaburzeń. Źródło: Opracowanie własne na podstawie [112].

branie jak największej liczby stron internetowych, np. CommonCrawl [31]. Dodatkowo należy uwzględnić jakość takiego korpusu (jak najmniej słów z literówkami, niegramatycznych zdań, kawałków kodu źródłowego stron internetowych, czy nic nieznaczących zlepek liter). Istotne jest odpowiednie oczyszczenie i wstępne przygotowanie danych, jednak Kaplan et al. pokazują, że przy wystarczająco dużej skali, zyski pochodzące ze zwiększania korpusu przewyższają zyski z dokładniejszego oczyszczania go [103]. Wielkość opisywanych korpusów mieści się w granicach $10^8 - 10^{13}$ tokenów.

3.3.2. *Fine-tuning* modeli NLP

W wielu przypadkach wykorzystanie dużych, wstępnie wytrenowanych modeli językowych uzasadnione jest niewystarczającą ilością danych odpowiadających docelowemu zadaniu T_2. Gdy jednak jest to możliwe, korzystne jest użycie modeli językowych wytrenowanych na korpusie pochodzącym z domeny przypominającej tematyką domenę docelową, np. SciBERT (wytrenowany na treściach artykułów naukowych [8]), BioBERT (wytrenowany na treściach publikacji biomedycznych [110]), LEGAL-BERT (wytrenowany na treściach dokumentów prawnych [21]), FinBERT (wytrenowany na treściach dokumentów o tematyce finansowej [201]).



Rysunek 3.5. Architektura modeli NLP opartych o bazowy model LM. W zależności od złożoności zadania konieczne jest dodanie mniejszej lub większej ilości warstw sieci. a) Jeżeli nowe zadanie modelu przypomina zadanie pierwotne, np. nieznacznie zmienione zadanie klasyfikacji, najczęściej wystarczające jest dodanie jednej warstwy odpowiadającej nowym kategoriom. b) Dla nowych zadań znacznie odbiegających od zadania pierwotnego, wymagane może być dodanie większej ilości warstw przetwarzających wyjście modelu LM w odpowiedni sposób. Źródło: Opracowanie własne na podstawie [127].

Posiadając model językowy, kolejnym krokiem jest TL, czyli wykorzystanie tego modelu jako bazy, do której dobudowywane są kolejne moduły (patrz rozdział 4.4.2). Głównym zadaniem modelu językowego jest przetworzenie informacji zawartej w tekście, a następnie przekazanie jej do nowej części modelu, która realizuje zadanie T_2. W przypadku, gdy zadaniem jest klasyfikacja całych zdań (lub dokumentów), wystarczające może być dodanie pojedynczej, gęsto połączonej warstwy sieci neuronowej (rysunek 3.5a). Podejście to sprawdza się szczególnie, gdy zadanie T_2 w znacznym stopniu przypomina zadanie T_1. Jeżeli realizowane zadanie T_2 jest bardziej skomplikowane (lub w znacznym stopniu odbiega od pierwotnego zadanie T_1), konieczne może okazać się zbudowanie większego modelu połączonego z modelem językowym (rysunek 3.5b).

Ogólny proces *fine-tuningu* modeli uczenia maszynowego i stosowane w nim techniki zostały opisane dokładniej w rozdziale 4.4.3

3.3.3. Benchmarki modeli językowych

Jakość coraz nowszych modeli NLP proponowanych przez badaczy można rzetelnie ocenić jedynie poprzez porównanie ich do siebie przy pomocy odpowiednich miar i testów. Każdemu z zadań T odpowiadają inne sposoby pomiaru jakości otrzymanych modeli.

Jedną z metod weryfikacji, czy model LM jest w stanie dobrze reprezentować język, jest zbudowanie na jego podstawie klasyfikatora, który odróżnia zdania poprawne gramatycznie od niepoprawnych. W tym celu został stworzony zbiór danych *Corpus of Language Acceptability* (Korpus Językowej Akceptowalności), na którego podstawie możliwa jest weryfikacja dowolnego modelu językowego [189]. Dostępna jest strona internetowa, na której znajduje się uaktualniany na bieżąco ranking modeli z najlepszymi wynikami [146]. Test językowej akceptowalności jako wynik podaje jedną wartość - dokładność.

Bardziej złożonym narzędziem do testowania jakości modeli LM jest benchmark *General Language Understanding Evaluation* (Ogólna Ocena Zrozumienia Języka), czyli GLUE. Jest to cały zestaw zbiorów danych testowych i określonych na nim metryk podzielonych na 9 różnych zadań NLP, dla których modele zbudowane są na wspólnym modelu językowym [188]. Ze względu na bardzo dynamiczny rozwój dziedziny, konieczne było uaktualnienie testu o trudniejsze przykłady, dzięki czemu powstała jego druga wersja - SuperGLUE [187]. Twórcy tych testów stworzyli również oficjalną stronę internetową z rankingiem, dzięki czemu możliwe jest porównanie jakości kolejnych modeli.

W przypadku modeli dających w wyniku predykcji sekwencje (tekst), ewaluacja musi odbywać się w nieco bardziej złożony sposób. W tym celu stworzony został zbiór testowy *Stanford Question Answering Dataset* (Zbiór odpowiedzi na pytania Uniwersytetu Stanford), czyli SQuAD [157]. Zbiór ten służy do badania jakości modeli odpowiadających na pytania w tekstowej formie. W wersji 2.0 test został utrudniony o dodatkowe pytania dotyczące przetwarzanego tekstu, na które odpowiedź może być niemożliwa - wówczas model powinien zaznaczyć, że nie da się udzielić odpowiedzi [156]. W tym wypadku autorzy również udostępniają oficjalny ranking modeli.

Istnieje wiele innych testów i sposobów ewaluacji najnowocześniejszych modeli językowych, jednak te opisane powyżej są obecnie najczęściej stosowane, a ogólnodostępne rankingi pozwalają badaczom na skuteczne porównanie wyników swojej pracy.

3.4. Narzędzia i modele stosowane obecnie w NLP

3.4.1. Język programowania i biblioteki

Współcześnie badania w dziedzinie NLP prowadzone są głównie z wykorzystaniem języka Python [181]. Python to język programowania ogólnego przeznaczenia, który

powstał we wczesnych latach 90. jako projekt *Open Source*, zarządzany przez Python Software Foundation. Jedne z najważniejszych cech, które spowodowały, że stał się popularny w naukowych zastosowaniach, to przejrzystość i zwięzłość kodu, duża dostępność otwartoźródłowych bibliotek, a także brak dużych wymagań co do środowiska sprzętowego. Python jest językiem skryptowym - jego kod nie jest kompilowany, przez co obliczenia mogą być wolniejsze niż na przykład w języku C. Wiele bibliotek numerycznych zostało jednak stworzonych w językach niższego poziomu, udostępniając tylko interfejs dla programisty w języku Python, dzięki czemu osiągają porównywalną szybkość przetwarzania danych.

Ze względu na zastosowanie sieci neuronowych w budowie modeli NLP, jedną z najważniejszych bibliotek w tej dziedzinie jest TensorFlow (wraz z biblioteką Keras) [1]. TensorFlow został stworzony w 2015 roku przez Google Brain Team w celu prowadzenia obliczeń na tensorach - koniecznych w budowaniu sieci neuronowych. Biblioteka składa się z kilku modułów, z których najistotniejszym jest silnik wykonawczy przeprowadzający obliczenia, napisany w języku C++ i umożliwiający wykonywanie obliczeń na różnych typach maszyn - CPU, GPU czy TPU. W celu ułatwienia użycia udostępnione zostały interfejsy (m.in. Keras), które ułatwiają budowanie kolejnych warstw sieci neuronowej bez konieczności zagłębiania się w szczegóły numeryczne i pozostawiają stworzenie grafu obliczeń odpowiednim komponentom.

Konkurencyjną biblioteką umożliwiającą trenowanie sieci neuronowych jest PyTorch napisany przez zespół badaczy Facebooka w 2016 roku [147]. Możliwości zastosowania oraz sposób działania jest bardzo zbliżony do TensorFlow - jedną z największych różnic jest mniejsza czytelność kodu PyTorch, co jednak równoważy łatwiejsze debuggowanie go.

Oprócz bibliotek do tworzenia sieci neuronowych ogólnego przeznaczenia, dostępne są również biblioteki wspomagające prace z danymi tekstowymi. Jedną z pierwszych takich bibliotek był stworzony w 2001 roku na Uniwersytecie Pensylwanii NLTK (Natural Language Toolkit) umożliwiający statystyczną i głównie symboliczną analizę tekstu [120]. Jednym z zastosowań było wykorzystanie go do edukacji w zakresie lingwistyki komputerowej, a także prototypowania algorytmów NLP.

W 2009 roku została opublikowana biblioteka Gensim, której główną funkcją jest statystyczne NLP - m.in. modelowanie tematu, wyszukiwanie informacji, indeksowanie dokumentów. Gensim pozwala na parsowanie strumienia tekstu (tekst nie musi być w całości wczytany do pamięci komputera), dzięki czemu umożliwia pracę z bardzo duży-

mi korpusami. Biblioteka ta jest wykorzystywana w wielu komercyjnych i akademickich projektach.

Obecnie najbardziej zaawansowaną biblioteką do przetwarzania tekstu jest spaCy [85]. Biblioteka ta powstała w 2015 roku jako rozwiązanie głównie komercyjne, wspierające wykorzystanie głębokich sieci neuronowych poprzez integrację m.in. z TensorFlow i PyTorch. SpaCy umożliwia pracę z kilkudziesięcioma językami, w tym polskim. Autorzy udostępniają wiele gotowych szablonów projektów NLP, które znacznie przyspieszają tworzenie nowych aplikacji.

3.4.2. Infrastruktura obliczeniowa

Wykorzystanie sieci neuronowych i dużych korpusów w NLP wiąże się ze zwiększonym zapotrzebowaniem na moc obliczeniową. Wraz z rozwojem sieci neuronowych i wzrostem dostępności dużych ilości danych, konieczne stało się użycie sprzętu komputerowego wspierającego równoległe przetwarzanie wektorów, macierzy i tensorów. Urządzeniami spełniającym te kryteria są m.in. karty graficzne (GPU). Dalszy wzrost liczby parametrów w proponowanych modelach, a także wielkości korpusów, wymusiły rozwój metod estymacji modeli na rozproszonej infrastrukturze - większość komercyjnych i akademickich grup badawczych zaczęło wykorzystywać w swoich placówkach maszyny połączone w klastry.

Na samym początku XXI wieku coraz większą popularność zaczęło zyskiwać prowadzenie obliczeń w chmurze. Firmy Amazon, Google i Microsoft rozpoczęły udostępnianie odpłatnie swojej infrastruktury obliczeniowej, dzięki czemu badacze mogli skupić się na swoich projektach bez konieczności stałego utrzymywania serwerów w odpowiednim stanie. Z czasem usługi przechowywania danych i prowadzenia obliczeń stawały się coraz tańsze, a dostępne maszyny coraz nowocześniejsze. W 2018 roku Google zaczął udostępniać użytkownikom komputery z układami TPU dedykowanymi właśnie do tworzenia dużych modeli ML.

Obecnie dostawcy rozwiązań chmurowych oferują duże zniżki dla środowisk akademickich i start-upów. Firma Google udostępnia środowisko Colab, w którym całkowicie nieodpłatnie można używać układów TPU z pewnymi limitami czasowymi.

3.4.3. Zbiory danych i pre-trenowane modele

Wielu badaczy w dziedzinie ML wraz z publikacją udostępnia zbiór danych, kod źródłowy, a także zapisane parametry wytrenowanych modeli na licencjach typu *Open*

Source. Dzielenie się najnowszymi wynikami i pracą umożliwiają dedykowane w tym celu serwisy. Ma to szczególne znaczenie w przypadku budowania modeli NLP opartych o pre-trenowane modele językowe. Modele LM można pobrać przez API odpowiednich serwisów, wykorzystać do swoich badań jako gotowe komponenty i opcjonalnie dokonać *fine-tuningu*, jak to opisano w podrozdziale 3.3.2.

Najbardziej znanym portalem internetowym zrzeszającym społeczność badaczy ML jest Kaggle [102]. Platforma Kaggle funkcjonuje od 2010 roku. Poza możliwością publikacji (oraz pobierania) oczyszczonych, gotowych do użycia zbiorów danych i wytrenowanych modeli, Kaggle jest platformą, na której firmy ogłaszają konkursy na najlepszy model zbudowany na udostępnionych przez nie danych. Ze względu na dużą popularność platformy, dużą konkurencję i nagrody pieniężne, otrzymane rozwiązania często pomagają posuwać naprzód bieżący stan wiedzy.

W ramach biblioteki TensorFlow od 2018 roku dostępny jest moduł TensorFlow Hub, który działa na zasadzie repozytorium gotowych modeli [1]. Dużą zaletą tego rozwiązania jest możliwość bezpośredniego wczytania modelu z wewnątrz kodu źródłowego Pythona przez odpowiednie wywołanie w API biblioteki. Modele dostępne w TensorFlow Hub muszą zostać zaakceptowane zanim będą mogły być udostępnione innym użytkownikom, co zapewnia ich lepszą jakość.

Platformą, która kładzie szczególnie duży nacisk na modelowanie NLP, jest Hugging Face [89]. Hugging Face to firma, założona w 2016 roku, tworzącą czatboty i narzędzia służące do ich tworzenia. Ze względu na potrzebę pracy z najbardziej zaawansowanymi modelami językowymi, założyciele stworzyli również platformę wymiany modeli i zbiorów danych z dziedziny NLP. Poza udostępnionymi zasobami Hugging Face dostarcza również bibliotekę i narzędzia do pracy z tekstem.

Bardzo istotnym narzędziem w modelowaniu są zbiory testowe, dzięki którym można zweryfikować jakość otrzymanych modeli. Zostały one opisane w podrozdziale 3.3.3.

3.5. Automatyczna analiza wiadomości finansowych

Inwestorzy w podejmowaniu decyzji zwykle kierują się informacjami. Zmiana w ramach konkretnej spółki (np. jej zarządu lub połączenie z inną) albo jej otoczenia rynkowego mogą spowodować, że uczestnicy rynku postanowią dokonać zmian w swojej pozycji. Jednak żeby wiedzieć o takich zdarzeniach, zmuszeni są stale monitorować wiele źródeł informacji, które w dobie Internetu stale się powiększają - pojedynczy

inwestor nie jest w stanie ręcznie przeanalizować pełnego strumienia wiadomości, które do niego spływają. Wobec tych zależności oczywistym rozwiązaniem wydaje się być próba zautomatyzowania procesu analizy wiadomości.

Język naturalny może przybierać różne struktury i style, przez co ekstrakcja informacji w nim zawartych może przebiegać na wiele sposobów. Klasyczna analiza szeregów czasowych (w tym ludzkiej mowy lub notowań giełdowych), a także analiza z zastosowaniem technik uczenia maszynowego jest od wielu lat opisywana w licznych publikacjach naukowych. Jednak dopiero rozwój infrastruktury obliczeniowej, głównie GPU i TPU, umożliwił w ostatnich latach znaczny postęp w cyfrowym przetwarzaniu języka naturalnego - pierwsze zastosowanie sieci LSTM miało miejsce w 1997 [84], a sieci neuronowych z mechanizmem uwagi w 2017 [183]. Dzięki tym technikom tekst jest reprezentowany nie tylko jako zbiór niepowiązanych słów kluczowych (Bag-of-words), ale uwzględniana jest też m.in. gramatyka, kontekst czy nacechowanie emocjonalne. Zastosowanie najnowszych technik NLP do optymalizacji strategii inwestycyjnych lub prognozowania finansowych szeregów czasowych jest obecnie opisane w stosunkowo niewielkiej liczbie publikacji m. in.: [42, 43, 59, 83, 169].

3.5.1. Statystyczna analiza tekstu wiadomości

Statystyczna analiza tekstu pozwala na bezpośrednią predykcję ruchu notowań na podstawie treści wiadomości. Zwykle wykorzystywane są w niej parametry opisujące tekst, takie jak np. częstotliwość występowania konkretnych słów lub ilości nowych artykułów. Istota tego podejścia polega na wykorzystaniu (poza historycznymi szeregami czasowymi stóp zwrotu) dodatkowych informacji w postaci przetworzonych wiadomości kierowanych do ludzi. Tego typu wiadomości zwykle opisują wydarzenia, które mogą mieć wpływ na zachowanie inwestorów, a ich analiza może dostarczyć **nowych** informacji jeszcze niezawartych w historycznych szeregach czasowych.

Zachowania notowań giełdowych spółek można analizować w oparciu o wiadomości publikowane na ich temat. Istnieją znaczące różnice w zależności od wydźwięku opublikowanych wiadomości – szczególnie duże zmiany następują w przypadku złych wiadomości. Posiadają one również nieco inne charakterystyki powrotu do średniej w przypadku, gdy żadne wiadomości się nie pojawiają. Efekty te zostały opisane przez Chana W. w 2003 roku na podstawie danych zebranych na przestrzeni 20 lat [22].

Liczba publikacji na temat wykorzystania NLP w celu analizy wpływu wydarzeń

na świecie na rynki finansowe jest stosunkowo nieduża. Wynika to głównie z faktu, że NLP jest dosyć nową gałęzią nauki.

Pierwsze próby automatycznej analizy wiadomości publikowanych w Internecie w celu przewidzenia notowań giełdowych polegały na zliczaniu ręcznie wybranych słów kluczowych m. in. w *The Wall Street Journal*. W 1998 Wuthrich B. et al. opublikowali swoje pierwsze wyniki bazujące na kilku różnych modelach (systemie regułowym, K Najbliższych Sąsiadów i sieci neuronowej). Strategia oparta o predykcje systemu dla indeksów giełdowych daje wyniki lepsze niż inwestycja typu "Kup i trzymaj" [200].

Kolejnym podejściem było zbadanie możliwość wykorzystania algorytmów genetycznych i statystycznych klasyfikatorów tekstu w celu predykcji na rynkach finansowych opracowane przez Thomas J. D. i Sycara K. w 2000 roku. W badaniach wykorzystano artykuły z różnego rodzaju biuletynów internetowych. Zaproponowana metoda pozwoliła osiągać zwroty z inwestycji o 30% wyższe niż proste inwestowanie w cały indeks [177].

Kierunek ruchów cen akcji, a także kwartalnego przychodu można próbować przewidzieć na podstawie liczby słów o negatywnym wydźwięku pojawiających się w wiadomościach. Tetlock P. i jego zespół badali teksty publikowane w *Dow Jones Stock News* oraz *The Wall Street Journal* oraz ich długoterminowy wpływ na skumulowane osiągi finansowe. Zastosowanie na tyle prostego podejścia pozwala istotnie na przewidzenie kierunku ruchu, jednak w przypadku rzeczywistego środowiska inwestycyjnego (uwzględniającego prowizje i podatki) nie pozwalałoby osiągać pozytywnych wyników [175].

Schumaker R. P. i Chen H. w 2009 stworzyli hybrydowy system, który wykorzystywał uczenie maszynowe, metody statystyczne oraz analizę tekstu do przewidywania kierunku ruchów notowań spółek na giełdzie NYSE z ponad 71% skutecznością. Symulowana stopa zwrotu była o 2,88 punktu procentowego wyższa niż przy strategii "Kup i trzymaj" realizowanej na całym indeksie S&P 500. Badacze do modelowania wykorzystali maszynę wektorów nośnych działającą w oparciu między innymi o statystyczną analizę teksu wiadomości [162].

W 2014 Ding X. et al przedstawili porównanie kilku modeli NLP [42], które na wejściu otrzymywały jedynie nagłówki wiadomości finansowych i przewidywały kierunek ruchu indeksu S&P 500 w handlu intra-day. Model dokonywał predykcji na podstawie wydarzeń dotyczących spółek, o których informacja pochodziła bezpośrednio z tekstu.

Średnia skuteczność na danych testowych najlepszego z proponowanych modeli osiągała wartość 58,94%.

3.5.2. Predykcja oparta o wydarzenia

Jedną z metod predykcji notowań giełdowych na podstawie tekstu jest próba ekstrakcji informacji na temat wydarzeń, które mogą wywołać zmiany w otoczeniu rynkowym danej spółki. Podejście to polega na wykryciu w tekście informacji o wszelkich wydarzeniach, jak miały miejsce, a następnie predykcji potencjalnych zmian ceny w zależności od typu wydarzenia.

W 2016 Ding X. et al poprawili osiągnięty wynik przez zastosowanie zewnętrznego grafu wiedzy, dzięki któremu sieć neuronowa nie musiała uczyć się od początku zależności pomiędzy faktami. W dodatkowym grafie wiedzy przechowywane były między innymi informacje o powiązaniach miedzy nazwiskami osób i spółek, a także ich stanowiska. Skuteczność modelu była większa o ponad 5 punktów procentowych dla większości spółek w porównaniu z wynikami badaczy sprzed 2 lat [43].

Kolejnym krokiem było zbadanie wpływu włączenia informacji upublicznianych przez spółki z powodu prawnych obowiązków (regulatory disclosures) na jakość predykcji indeksów giełdowych. W 2018 Feuerriegel S. i Gordon J. w zaproponowanym przez siebie modelu z pozytywnym skutkiem wykorzystali te dane do podniesienia skuteczności predykcji indeksów GAX, CDAX i STOXX Europe 600, zwłaszcza w długim horyzoncie czasowym [59].

Wpływ światowych wydarzeń na wyniki akcji może być różny dla gospodarek o różnych profilach. Maqsood H. i in. stworzyli model prognozowania cen w oparciu o sentyment do konkretnych wydarzeń na podstawie ponad 11 milionów wiadomości publikowanych na Twitterze dla spółek w Stanach Zjednoczonych, Hong Kongu, Turcji i Pakistanie (rynki rozwinięte, wschodzące i słabo rozwinięte). Jednym z wniosków autorów był fakt, że lokalne wydarzenia zwykle mają większy wpływ niż globalne, a także że wydarzenia lokalne w bardziej rozwiniętych gospodarkach (np. wybory prezydenckie w USA) mogą również mieć znaczący wpływ na zachowania cen akcji w innych krajach [124].

Jedną z metod ekstrakcji informacji o wydarzeniach opisanych w wiadomościach jest zastosowanie sieci z mechanizmem uwagi, np. BERT lub AL-BERT ([39, 109]). Podejście to zostało wykorzystane przez X. Wu do stworzenia systemu, który najpierw ekstrahował informacje o wydarzeniach, a następnie dokonywał predykcji trendu

cen. Badania zostały przeprowadzone na indeksach S&P 500, Dow Jones i Nasdaq, zaś wiadomości pochodziły z publikacji Reutersa i Bloomberga. Otrzymane rezultaty potwierdziły, że to podejście daje wyniki lepsze niż inwestowanie z indeksy [199].

Jednym z najbardziej istotnych wydarzeń w analizie inwestycyjnej jest publikacja rocznych raportów finansowych. W 2023r. Glodd A. i Hristova D. zaproponowali dwuetpaowe podejście do ekstrakcji z tego typu publikacji [66]. Ich rozwiązanie polegało na wykorzystaniu jednej z odmian modelu BERT (DistilBERT [160]) do rozpoznawania fragmentów raportów mogących mieć wpływ na dalsze przebiegi cen, a następnie użyciu innego modelu typu BERT (FinBERT [201]) do analizy sentymentu tych fragmentów. Wyniki otrzymane przy pomocy takiego systemu są lepsze niż wykorzystanie do prognozowania wyłącznie historycznych cen.

3.5.3. Analiza sentymentu

Bardzo istotnym nurtem w wykorzystaniu NLP w ekonometrii stała się analiza sentymentu inwestorów. W przeciwieństwie do próby bezpośredniego przełożenia treści wiadomości giełdowych na zachowania notowań, analiza sentymentu dodaje krok pośredni, w którym na podstawie treści oceniane jest nastawienie inwestorów do danego aktywa, a w konsekwencji kierunek zmian notowań.

W 2004 Antweiler W. i Frank M. Z. rozpoczęli nowy trend – analizę treści stworzonych przez użytkowników grup dyskusyjnych (jak dotąd analizowano wyłącznie artykuły oficjalnie publikowane). Była to zatem pierwsza próba ekstrakcji opinii i „wiedzy tłumu". Do analizy wykorzystano ponad 1,5 miliona wiadomości opublikowanych na Yahoo! Finance i Raging Bull. Wiadomości były klasyfikowane na pozytywne i negatywne, zatem podejście to było bardzo zbliżone do analizy sentymentu. Autorzy potwierdzili pozytywny wpływ włączenia informacji pochodzących z forów dyskusyjnych do modelowania zmienności z częstotliwością dzienną, a także większą [4].

Trend ten był kontynuowany m. in. przez Kim Y. i in. podczas ich prób analizy opinii na dużą skalę (*opinion mining*), w której użyli ponad 78 000 wypowiedzi na temat spółek wchodzących w skład Korea Composite Stock Price Index [105]. Celem badań była klasyfikacja wiadomości na pozytywne, negatywne i neutralne, w celu przewidzenia, czy cena aktywa wzrośnie, spadnie, czy pozostanie na tym samym poziomie. Skuteczność otrzymanego modelu znajdowała się na poziomie 60-65%.

Mimo iż większość publikacji dotyczy akcji spółek, ze względu na mnogość zjawisk, które mogą ich dotyczyć (przejęcia, nowe produkty, skandale z osobami powiązanymi),

analizę sentymentu można również zastosować do prognozowania cen surowców. Li J. i in. przeprowadzili analizę wykorzystując artykuły publikowane przez Reutersa oraz ceny ropy (West Texas Intermediate) do stworzenia kilku modeli - maszyny wektorów nośnych, regresji logistycznej i sieci neuronowej. Za pomocą przyczynowości Grangera dowiedli, że sentyment wiadomości rzeczywiście ma wpływ na ceny ropy, a także zaproponowali modele, które dawały poprawne predykcje [113].

Thársis S. i Tomaso A. zaproponowali istnienie dwóch równoległych, powiązanych struktur – finansowej i społecznej. Przez odwzorowanie relacji między notowaniami różnych spółek na obu tych strukturach, możliwe było dokładniejsze przewidzenie ich wzajemnego wpływu na siebie i podniesienie skuteczności predykcji [169].

Po publikacji Devlina J. i in. ([39]) zaczęło pojawiać się wiele prac na temat wykorzystania architektury z mechanizmem uwagi (BERT) do różnych zadań NLP. Jednym z nich była właśnie ogólnie rozumiana analiza sentymentu (niekoniecznie rynkowego). Li X. i in. zaproponowali prosty model oparty o model językowy BERT z dodatkową warstwą do klasyfikacji sentymentu. Otrzymany w ten sposób model może służyć jako *benchmark* jakości tego typu modeli [115].

W 2019 Hiew J. i in. dokonali analizy możliwości przewidywania notowań giełdowych trzech popularnych firm na giełdzie w Hongkongu poprzez badanie sentymentu inwestorów w dyskusjach na Weibo.com przy użyciu sieci BERT [83]. Wynikiem ich pracy było zestawienie różnych modeli i ich skuteczności w zależności od horyzontu czasowego.

Rozszerzeniem analizy sentymentu jest analiza emocji inwestorów. Charalampos L. i Sotiris K. stworzyli model oparty o sieci konwolucyjne (zamiast sieci typu LSTM) połączone z modułem typu BERT w celu analizowania postów z serwisu Twitter (obecnie X) [116]. W swojej pracy z 2023r. wykorzystali 28 różnych emocji, które model starał się przypisać odpowiednim postom użytkowników, aby następnie wykorzystać tę informację do lepszego prognozowania dziennej ceny zamknięcia. Wyniki otrzymane przy pomocy tego modelu przewyższały wyniki otrzymywane przy użyciu modeli nieuwzględniających emocji inwestorów pod względem wielkości błędów prognoz.

3.5.4. Pozostałe podejścia

Dzięki wykorzystaniu modelu językowego BERT Chen Q. zbudował model, którego celem było przewidzenie trendów cen akcji wchodzących w skład Europe STOXX 600 Index bez pośredniego przejścia przez ekstrakcję z tekstu wydarzeń lub sentymentu.

Otrzymany model pozwolił stworzyć strategię inwestycyjną, która osiąga bardzo dobre wyniki (12,6% CAGR) nawet po uwzględnieniu kosztów transakcji [23].

Analizę sentymentu można połączyć z generatywnymi modelami w celu tworzenia predykcji w postaci oczekiwanego szeregu czasowego. Takie podejście wykorzystali P. Muthukumar i J. Zhong tworząc dwustopniowy model. W pierwszym kroku wiadomości są klasyfikowane na pozytywne i negatywne. W drugim kroku otrzymane klasy wykorzystywane są wraz z samym szeregiem czasowym cen do prognozowania dalszego ich przebiegu (w horyzoncie 1-, 15- i 30-dniowym). Autorzy bazowali na akcjach spółek branży lotniczej ze względu na silne powiązanie jej wyników ze światowymi wydarzeniami. Błąd predykcji mierzony jako pierwiastek błędu średniokwadratowego - ang. *Root Mean Squared Error* (RMSE) przewyższał inne stosowane do tego momentu metody, w tym głębokie sieci neuronowe [133].

Należy zauważyć, że stworzenie modelu przewidującego sentyment na podstawie tekstu wymaga dużej ilości poprawnie oznaczonych wiadomości. Najprostszym, jednak najbardziej czasochłonnym rozwiązaniem, jest przeczytanie i ręczne oznaczenie każdej wiadomości. Birbeck E. i Cliff D. zaproponowali wykorzystanie przebiegów cen akcji, aby automatycznie oznaczać sentyment wiadomości. Jest to zadanie odwrotne do zwykle realizowanego, jednak pozwala łatwiej pozyskać duże ilości danych treningowych [10].

Praktyczne użycie modeli NLP często wymaga nie tylko samego modelowania, ale również całej procedury odpowiedniej obróbki danych - czyszczenia wiadomości i odpowiedniego umiejscowienia ich w czasie względem szeregu czasowego cen. Dong Y. i jego zespół zaproponowali rozwiązanie pomagające wykorzystać dane z Twittera do usprawnienia predykcji cen akcji notowanych na NYSE w odporny na zakłócenia sposób. Publikacja ma raczej inżynieryjną charakterystykę, jednak wskazuje szereg istotnych szczegółów, na które inni badacze powinni zwracać uwagę w swoich systemach predykcji notowań giełdowych wykorzystujących dane tekstowe [46].

W 2022 roku zespół Google Research zbadał, jakie informacje na temat rynków mogą posiadać duże modele językowe [38]. Badania opierały się na generowaniu tekstu o tematyce inwestycyjnej i sprawdzaniu, czy model potrafi utrzymywać kontekst o znaczeniu finansowym. W swojej publikacji wykorzystywali dane pochodzące z serwisu Reddit. Następnie model generujący tekst został zaadaptowany na model klasyfikujący sentyment. Wyniki otrzymywane przy jego pomocy były porównywalne z modelami uczenia nadzorowanego, gdzie wszystkie dane musiały być opisane przez człowieka. Te-

go typu system może znacznie uprościć tworzenie nowych modeli poprzez przyspieszenie wstępnej obróbki danych.

Przedstawione podejścia do automatycznej analizy wiadomości giełdowych pozwalają na osiąganie dobrych wyników predykcji w dziennych horyzontach czasowych. Jak dotąd nie ma jednak podejść, które pozwalałyby na wykorzystanie treści wiadomości na poziomie *intra-day*. Jednym z celów niniejszej pracy jest wypełnienie tej luki badawczej.

3.6. Podsumowanie

Przetwarzanie języka naturalnego nie jest całkiem nową dziedziną nauki. Pierwsze osiągnięcia NLP w kryptografii pojawiły się już w IX wieku. Na przestrzeni kolejnych wieków metody te były rozwijane, a także stosowane w tłumaczeniu maszynowym. Pod koniec XX wieku systemy regułowe zostały zastąpione podejściem statystycznym, co umożliwiło wykorzystanie komputerów i znaczne przyspieszenie obliczeń. Jednak dopiero zastosowanie sieci neuronowych i dużych korpusów umożliwiło osiągnięcie rezultatów zbliżonych do rozumienia tekstu na poziomie człowieka.

Algorytmy NLP często składają się z wielu komponentów, które analizują tekst pod różnymi kątami. Podstawowym problemem jest odpowiednie reprezentowanie tekstu w pamięci komputera tak, aby był on możliwy do przetworzenia przez modele. Zwykle reprezentacja tekstu polega na stworzeniu słownika, który każdemu słowu przypisuje odpowiedni wektor liczb. Wektory te powstają w procesie trenowania *modeli językowych*, czyli sieci neuronowych, których zadanie jest dobrane w taki sposób, aby umożliwić nienadzorowane uczenie na łatwo dostępnym, dużym korpusie, np. uzupełnianie tekstu. Otrzymane w ten sposób modele mogą służyć jako baza do wykonywania innych zadań - ekstrakcji informacji, analizy sentymentu czy NER.

Współcześnie jako *model językowy* najczęściej wykorzystywane są głębokie sieci neuronowe z mechanizmem uwagi. Wielu badaczy udostępnia wytrenowane przez siebie modele, których architekturę można następnie zmodyfikować lub rozszerzyć tak, aby realizowały inne zadania. W ten sposób znaczna część modelu jest już wstępnie wytrenowana na korpusach ogólnego przeznaczenia i nawet stosunkowo niewielkie ilości danych odpowiadających docelowemu zadaniu pozwalają na skuteczny *fine-tuning*.

Większość najbardziej zaawansowanych modeli jest rozwijana i publikowana na licencjach Open Source, dzięki czemu w łatwy sposób można pobrać wyczyszczone zbiory danych tekstowych, pre-trenowane modele językowe (i ogólnie modele NLP), a

także porównać swoje wyniki używając ogólnodostępnych rankingów. Rozwijanie kodu źródłowego ułatwiają biblioteki języka Python, przygotowane do pracy z głębokimi sieciami neuronowymi i dużymi korpusami.

Modele NLP są z powodzeniem stosowane w automatycznej analizie wiadomości finansowych. Powstało wiele podejść bazujących na: analizie statystycznej słów pojawiających się w wiadomościach finansowych i raportach publikowanych przez spółki, ekstrakcji informacji o wydarzeniach opisywanych w wiadomościach, a także na analizie sentymentu inwestorów i treściach publikowanych na portalach społecznościowych. Mimo tak wielu podejść, żadne nie przedstawia metod wykorzystania treści wiadomości do prognozowania cen na poziomie *intra-day*. Celem niniejszej pracy jest m.in. uzupełnienie tej luki badawczej.

Rozdział 4

Proponowana metodologia badań wykorzystania technik NLP w analizie treści wiadomości finansowych

Celem prowadzonych badań jest sprawdzenie, czy możliwe jest wykorzystanie zautomatyzowanej analizy treści wiadomości finansowych do poprawy jakości prognoz na rynkach kapitałowych. Techniki NLP są dziedziną ściśle powiązaną z metodami numerycznymi i inżynierią oprogramowania. Wobec tego głównym narzędziem badawczym będą analizy i symulacje komputerowe wykorzystujące dane w postaci historycznych szeregów stóp zwrotów pojedynczych spółek, a także wiadomości opublikowane na ich temat w serwisach informacyjnych o tematyce finansowej.

W celu przeprowadzenia badań, stworzony został autorski system, na który składa się komponent gromadzący dane (historyczne przebiegi cen oraz pełną treść opublikowanych wiadomości), programy umożliwiające stworzenie modeli bazowych typu ARMA-GARCH oraz głębokich sieci neuronowych - sieci typu LSTM, do której dołaczony jest autorski moduł: FENE. Dodatkowo stworzone zostały programy pozwalające na porównanie otrzymanych modeli wykorzystując wachlarz mierników trafności prognoz oraz testów statystycznych.

4.1. Metodyka weryfikacji hipotez badawczych

We wstępie niniejszej pracy zostały postawione hipotezy badawcze. Ich weryfikacja jest oparta o obliczenia wykonane przy pomocy zbudowanych modeli. Wykorzystane zostały różne metody weryfikacji dopasowane do każdej z hipotez.

***Hipoteza*: Możliwe jest efektywne zautomatyzowanie analizy finansowych wiadomości oraz postów w mediach społecznościowych poprzez użycie współczesnych metod przetwarzania języka naturalnego – głównie głębokich sieci neuronowych.**

Metoda weryfikacji: Weryfikacja powyższej hipotezy opiera się na porównaniu modeli LSTM-FENE wykorzystujących analizę NLP z modelami bazowymi, które wiedzę o wydarzeniach mogą czerpać wyłącznie z historycznych szeregów czasowych oraz wzorców sezonowości (będących pośrednią metodą obserwacji zwiększonej ilości nowych informacji pojawiających się wśród uczestników rynku). Porównanie wykorzystuje skonfrontowanie mierników trafności prognoz (opisanych w rozdziale 4.5.3) osiągniętych przy pomocy badanych modeli. Oprócz tego, wykorzystany został statystyczny test Diebolda-Mariano [40], który umożliwia porównanie modeli różnego typu. Efektywne zautomatyzowanie analizy wiadomości powinno przełożyć się na lepsze wyniki osiągane przez model.

***Hipoteza*: Wykorzystanie treści całej wiadomości w analizie pozwala na osiągnięcie lepszych wyników prognozowania niż w przypadku wykorzystania wyłącznie treści nagłówków**.

Metoda weryfikacji: Nagłówki wiadomości finansowych są krótkim podsumowaniem ich treści. Informacje zawarte w pełnym tekście tych wiadomości mogą jednak dostarczać dodatkowych przesłanek dla modelu, dzięki którym jakość prognoz będzie lepsza. W celu zweryfikowania, czy faktycznie tak się dzieje, porównane zostały ze sobą modele o takich samych architekturach, jednak w procesie treningu użyte zostały pełne teksty wiadomości lub wyłącznie nagłówki. Prognozy z tych modeli zostały porównane ze sobą przy wykorzystaniu odpowiednich mierników jakości prognoz oraz testów statystycznych.

***Hipoteza*: Modele oparte na głębokich sieciach neuronowych z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z modelami wykorzystującymi klasyczne metody sztucznej inteligencji.**

Metoda weryfikacji: Weryfikacja tej hipotezy polega na porównaniu modeli LSTM-FENE wykorzystujących analizę NLP z modelami opartymi o sieć neuronową typu LSTM, która nie otrzymuje informacji pochodzących z wiadomości finansowych w bezpośredni sposób. Porównanie jakości modeli odbywa się poprzez analizę mierników trafności prognoz oraz wyników testu Diebolda-Mariano [40].

***Hipoteza*: Wykorzystanie szeregu czasowego notowań o 5 minutowej czę-**

stotliwości jest wystarczające, aby zaobserwować zmienioną charakterystykę zwrotów spowodowaną publikacją wiadomości.

Metoda weryfikacji: Momenty pojawienia się nowych informacji (publikacji wiadomości dotyczących spółek) to punkty w czasie, gdy inwestorzy mogą podjąć nowe decyzje i zwykle są początkiem wytrącenia rynku ze stanu ekwilibrium efektywności. Ponowne osiągnięcie stanu efektywności rynkowej to krótki okres, w którym zaproponowany model może sprawdzać się lepiej. Weryfikacja, czy agregacja danych do 5-minutowych interwałów jest wystarczająca polega na sprawdzeniu, czy efektywność prognoz jest większa w okresach po pojawieniu się nowych informacji w porównaniu z okresami bez publikacji żadnych wiadomości.

***Hipoteza*: Zautomatyzowana analiza finansowych wiadomości i postów w mediach społecznościowych za pomocą algorytmów NLP w połączeniu z obecnie stosowanymi metodami prognozowania finansowych szeregów czasowych pozwala na poprawienie efektywności predykcji.**

Metoda weryfikacji: Możliwość połączenia modeli ekonometrycznych z sieciami neuronowymi poprzez włączenie wektora enkodującego treść wiadomości pozwala na stworzenie dodatkowego typu modelu. Skuteczność predykcji tej metody została przetestowana przez porównanie prognoz modelu ARMA-GARCH-FENE z prognozami modeli ARMA-GARCH, wykorzystujących wzorce wolumenu będące pośrednią informacją o pojawianiu się nowych informacji, a także z prognozami modeli LSTM-FENE.

***Hipoteza*: Klasyczne modele ekonometrii finansowej typu ARMA-GARCH z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z klasycznymi modelami typu ARMA-GARCH bez tego modułu.**

Metoda weryfikacji: Weryfikacja tej hipotezy jest oparta na porównaniu modeli ARMA-GARCH-FENE wykorzystujących analizę NLP z modelami ARMA-GARCH, które nie otrzymują informacji pochodzących z wiadomości finansowych. Porównanie jakości modeli polega na analizie mierników trafności prognoz oraz wyników testu Diebolda-Mariano [40].

***Hipoteza*: Klasyczne modele ekonometrii finansowej typu ARMA-GARCH z autorskim modułem FENE generują lepsze (dokładniejsze)**

punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z modelami opartymi na głębokich sieciach neuronowych z autorskim modułem FENE.

Weryfikacja powyższej hipotezy opiera się na porównaniu modeli ARMA-GARCH-FENE wykorzystujących analizę treści wiadomości finansowych z modelami LSTM-FENE, które również otrzymują informacje zawarte w wiadomościach. Porównanie wykorzystuje analizę mierników trafności prognoz osiągniętych przy pomocy badanych modeli oraz przeprowadzenie testu Diebolda-Mariano [40].

4.2. Architektura autorskiego systemu prognozowania notowań giełdowych

Programy komputerowe i komponenty, stworzone do przeprowadzenia badań w niniejszej pracy, mogą być traktowane jako wszechstronny system. Dzięki modułowej budowie łatwo jest dokonywać zmian w poszczególnych etapach, a także porównywać wyniki różnych wariantów modeli.Programy tworzą jednolitą całość, w której dane przepływają zawsze w takim sam sposób, dzięki czemu osiągane wyniki są powtarzalne. W dużym uproszczeniu działanie autorskiego systemu można podzielić na dwa etapy. Architektura systemu została przedstawiona na rysunku 4.1.

Pierwszym krokiem jest wczytanie danych. Wszystkie konieczne zmienne są obliczone i zapisane w lokalnej bazie danych przez moduł wstępnego przetwarzania danych, opisany w rozdziale 4.2.1 (na rysunku 4.1 zaznaczony kolorem zielonym). Biblioteki wykorzystywane do estymacji modeli mogą wymagać podania całego zakresu historycznych danych (np. pakiet *rugarch* w języku R) lub podzielenia ich na krótkie okresy zawierające wszystkie opóźnione wektory (np. biblioteka *TensorFlow* w języku Python). Ostatni etap przygotowania danych różni się w zależności od wykorzystywanych narzędzi, dlatego został on włączony w etap estymacji modeli.

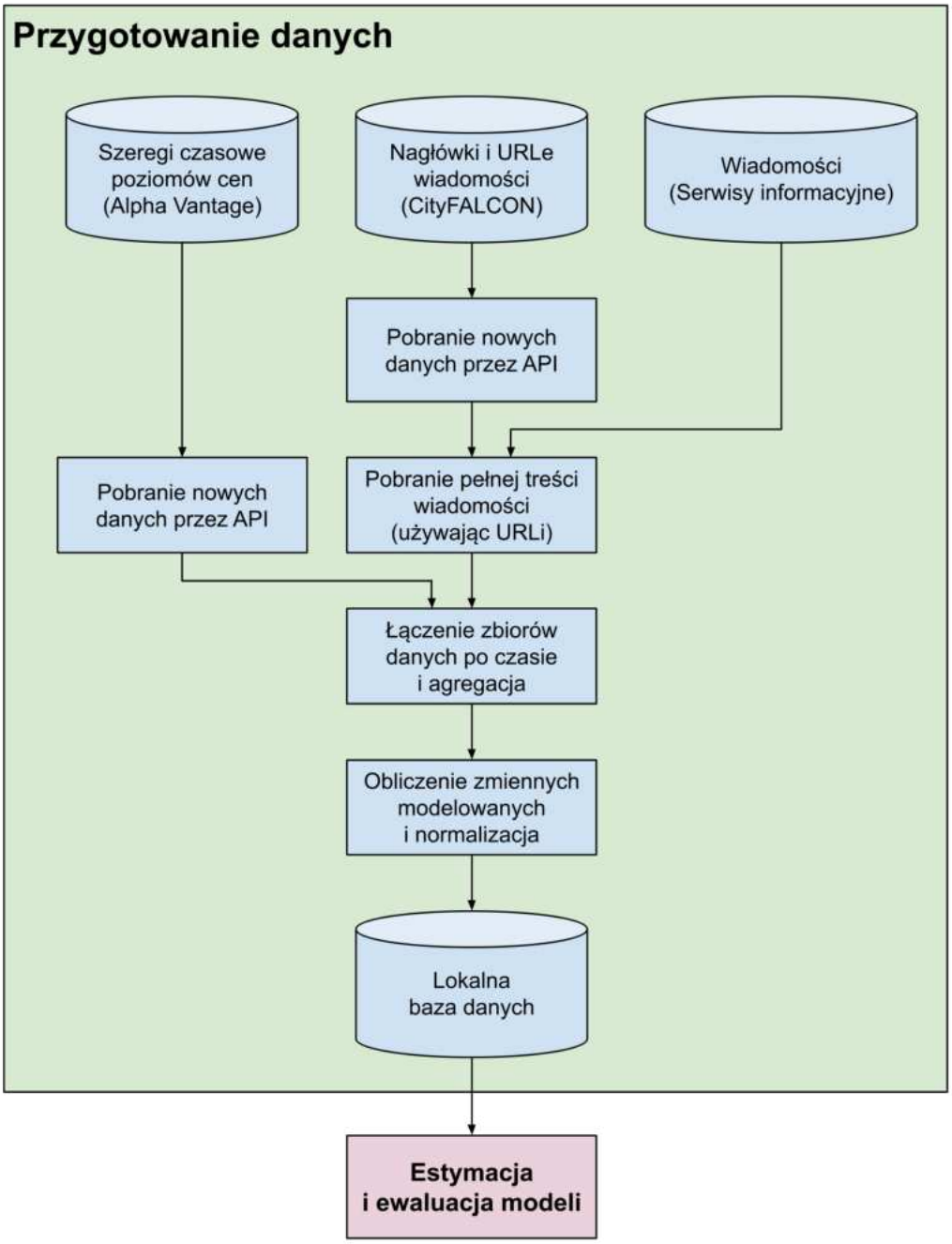
Drugi etap działania realizuje moduł estymacji i ewaluacji wszystkich modeli wykorzystywanych w badaniach: bazowych modeli ARMA-GARCH, autorskiego modelu LSTM-FENE oraz hybrydowych modeli ARMA-GARCH-FENE. Szczegółowy opis tej części systemu znajduje się w rozdziale 4.2.2

Rysunek 4.1. Ogólny schemat budowy autorskiego systemu prognozowania notowań giełdowych. Przygotowanie danych składa się z pobrania danych ze wszystkich źródeł, przetworzenia ich do wspólnej formy oraz obliczenie zmiennych wymaganych do modelowania. Estymacja i ewaluacja modeli to wygenerowanie modeli dla wszystkich badanych instrumentów, obliczenie prognoz oraz ich miar trafności.
Źródło: Opracowanie własne.

4.2.1. Przygotowanie danych

Autorski system prognozowania wykorzystuje dwa podstawowe typy danych: szeregi czasowe poziomów cen akcji spółek i wiadomości na temat tych spółek opublikowane w internetowych serwisach informacyjnych. Oba typy mają odmienną specyfikę oraz źródła i muszą być połączone w sposób zapewniający odpowiednie wzajemne umiejscowienie w czasie. Następnie na podstawie surowych danych obliczane są zmienne modelowane, które ostatecznie podlegają normalizacji i w takiej formie zostają zapisane w lokalnej bazie danych, skąd mogą być bezpośrednio wczytane i wykorzystane do modelowania. Proces ten został przedstawiony na rysunku 4.2.

Szeregi czasowe cen aktywów są pobierane przez API serwisu Alpha Vantage [3]. Dane są w postaci plików tekstowych CSV o **1-minutowej rozdzielczości** i dla każdego dnia, kiedy otwarta jest giełda, pokrywają rozszerzone godziny handlu. Dla każdego aktywa zdefiniowane są metadane zawierające między innymi strefę czasową giełdy, na której instrument jest notowany i datę ostatniego odświeżania danych na serwerze. Szeregi czasowe dla każdego okresu zawierają cenę otwarcia, cenę zamknięcia, najwyższą cenę, najniższą cenę oraz wolumen obrotu. System pobiera te informacje dla listy wszystkich spółek zdefiniowanych przez użytkownika.



Rysunek 4.2. Schemat algorytmu przygotowującego dane. Szeregi czasowe poziomów cen są wzbogacane informacjami z serwisów informacyjnych. Aby odnaleźć wszystkie pierwotne publikacje wykorzystywany jest serwis CityFALCON. Dane agregowane są do interwałów 5-minutowych, a następnie obliczone zostają zmienne objaśniające.
Źródło: Opracowanie własne.

Wiadomości giełdowe używane w systemie powinny być zdeduplikowane i składać się wyłącznie z oryginalnych wiadomości, tj. najwcześniejszych doniesień na temat danego wydarzenia. Tak określone wiadomości mogą pochodzić z wielu różnych źródeł, które należałoby stale monitorować. Dodatkowo wiadomości mogą być edytowane i zmieniane już po ich publikacji, często zaburzając czas podany przy artykule. Serwis CityFALCON zajmuje się m.in. dostarczaniem nagłówków i URLi (linków) wszystkich oryginalnych wiadomości dla wybranych przez użytkownika spółek [27]. Wykorzystując API CityFALCON system pobiera URLe do wiadomości na temat każdej z wybranych przez użytkownika spółek wraz z dokładnym czasem pierwszej publikacji artykułu, która jest zaokrąglana **do pełnej minuty**.

Kolejnym krokiem jest pobranie pełnej treści wiadomości dla wszystkich URLi otrzymanych z serwisu CityFALCON. Większość współczesnych serwisów informacyjnych wykorzystuje zaawansowane techniki tworzenia stron internetowych. Artykuły bardzo często poza tekstem zawierają obrazy, tabele, wykresy czy interaktywne elementy wykorzystujące *JavaScript*. Przetworzenie wszystkich tych treści stanowi ogromne wyzwanie, wobec czego niniejsza praca skupia się wyłącznie na części tekstowej artykułów. W celu połączenia ze stroną i wydobycia z niej kodu źródłowego HTML system wykorzystuje bibliotekę *requests* w języku Python. Kod HTML jest następnie oczyszczany przy pomocy wyrażeń regularnych, pozostawiając jedynie tekst zawierający treść artykułu. Dokładność ekstrakcji tekstu byłaby większa, gdyby wykorzystać dedykowane *crawlery*, jednak ze względu na mnogość używanych stron konieczne jest zastosowanie bardziej uniwersalnego rozwiązania. Ostatecznie system otrzymuje zbiór danych składający się z nagłówków, pełnego tekstu wiadomości finansowych, a także dokładnego czasu ich pierwszej publikacji.

Do skonstruowania połączonego zbioru danych, wykorzystywane są biblioteki języka Python *pandas* [176] oraz NumPy [79] służące do manipulacji danymi tabelarycznymi. Szeregi czasowe z serwisu Alpha Vantage zostają uzupełnione w przypadku braków, poprzez skopiowanie wartości z poprzedniego okresu. Następnie każda wiadomość zostaje przypisana do odpowiadającego jej w czasie notowania. Tak zbudowana tabela w zdecydowanej większości wierszy nie posiada żadnych wiadomości tekstowych ze względu na częstotliwość pojawiania się nowych artykułów. Dane są następnie agregowane do arbitralnie wybranej częstotliwości (**5 minut** - chyba, że zaznaczono inaczej) poprzez wybranie z każdego okresu:

— pierwszej ceny otwarcia,
— ostatniej ceny zamknięcia,
— minimum najniższej ceny,
— maksimum najwyższej ceny,
— sumę wolumenu,
— listę wszystkich wiadomości w tym okresie.

Ostatecznie odrzucane są obserwacje z okresów poza podstawowymi godzinami handlu.

Dla tak otrzymanych danych utworzone zostają dodatkowe zmienne używane w modelowaniu. Podstawową modelowaną wielkością jest logarytmiczna stopa zwrotu obliczona na podstawie cen zamknięcia. Wolumen jest znormalizowany poprzez obliczenie logarytmu jego wartości. Dodane zostają stosowane w sieciach neuronowych sztuczne zmienne czasowe. Są one utworzone poprzez obliczenie harmonicznych (funkcji sinus i cosinus) z wartości czasu dla każdego wiersza. Wykorzystuje się zwykle okresy pokrywające oczywistą sezonowość - rok i tydzień. Tak dodane zmienne pozwalają sieciom łatwiej modelować zmiany w danych, wynikające z sezonowości.

Utworzone zostają również wzorce sezonowości (wolumenu) wykorzystywane w modelach ARMA-GARCH obliczone zgodnie z zaleceniami w publikacji Będowskiej-Sójki B. [106]. W niniejszej pracy użyta została metoda uśrednionych obserwacji śróddziennych [7, 64]. W danych śróddziennych widoczne są regularne okresowe wahania, m.in. w zmienności, wolumenie i spreadzie bid-ask. Poprzez uśrednienie tych wahań dla wielu sesji otrzymywane są dzienne wzorce sezonowości wyznaczone dla każdego dnia tygodnia osobno.

W celu oszacowania estymatora składowej periodycznej śróddziennej obliczone zostają zwroty standaryzowane przy pomocy następującego wzoru:

$$\hat{r}_{t,n} = \frac{r_t}{\hat{\sigma}_t}, \tag{4.1}$$

gdzie $\hat{\sigma}_t$ jest estymatorem zmienności dziennej, który w niniejszej pracy został przyjęty jako dzienną zmienność zrealizowaną [13]. Następnie dla każdego interwału n wyznaczony może zostać estymator zmienności śróddziennej $\hat{s}_n^2$:

$$\hat{s}_n^2 = \frac{1}{T} \sum_{t=1}^{T} \hat{r}_{t,n}. \tag{4.2}$$

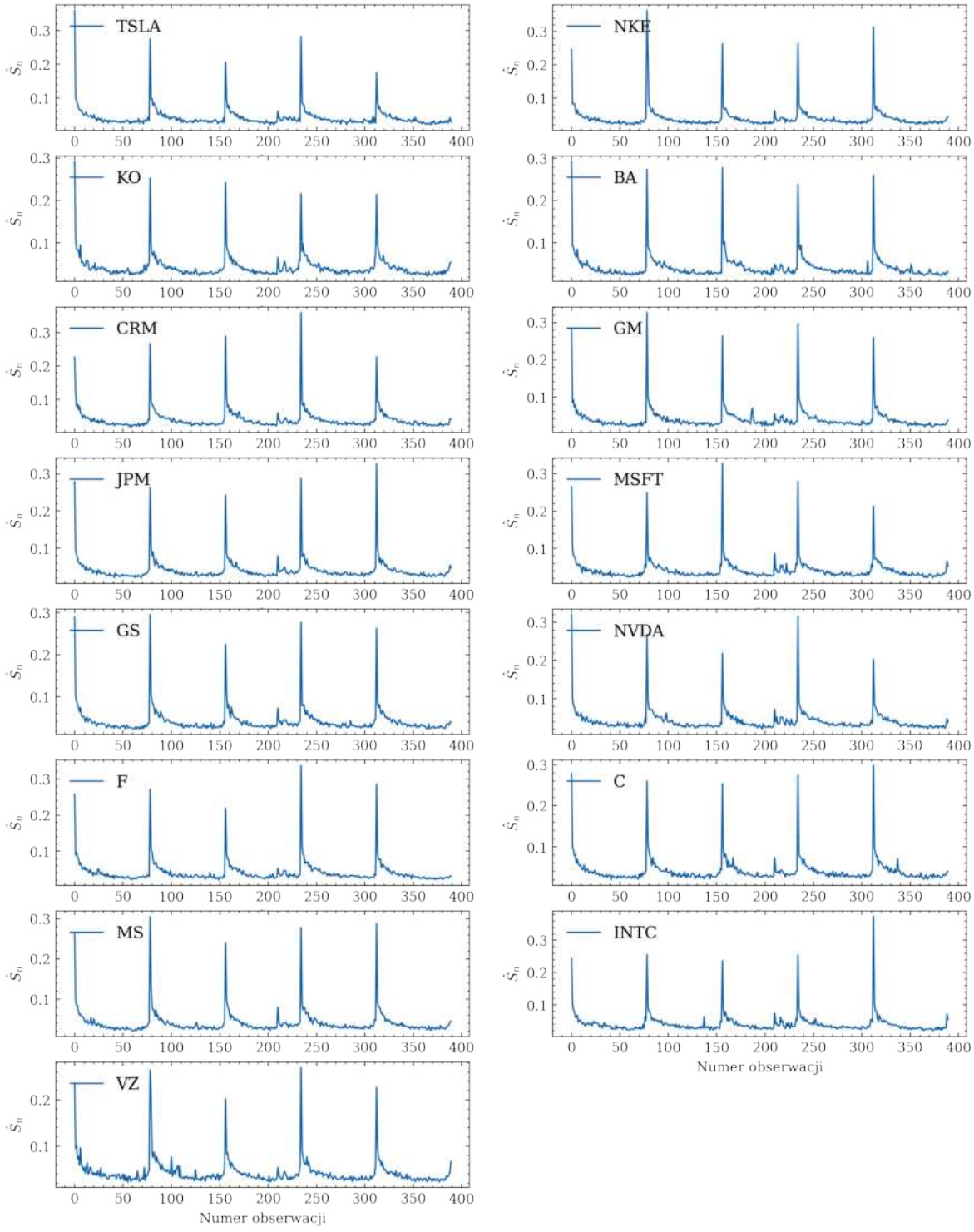
Wzorce sezonowości (stóp zwrotu i wolumenu) dla spółek wybranych do badania

są przedstawione na rysunkach 4.3 i 4.4. Wzorzec dla stóp zwrotu posiada kształt litery "L" - duże wartości występują na początku każdej sesji, a następnie gwałtownie spadają i pozostają niskie aż do końca. Kształt tej krzywej jest zgodny z opisem przebiegów w literaturze [106]. W przypadku wzorca wolumenu dla każdej ze spółek można zaobserwować ewidentnie wyższe wartości na początku i na końcu każdej sesji (kształt litery "U", również zgodny z tymi opisywanymi w literaturze). Dla większości spółek widać też pewną symetrię przebiegu dnia - po okresie bardzo dużych wartości na początku sesji wolumen powoli opada, po czym gwałtownie rośnie na koniec dnia. Jest to szczególnie widoczne w przypadku Tesli (TSLA), Boeinga (BA) i Forda (F). Przebiegi dla różnych dni różnią się dość nieznacznie między sobą, z pewnymi wyjątkami: wolumen Microsoft Corporation (MSFT) i The Coca-Cola Company (KO) jest widocznie większy w poniedziałki i wtorki, zaś dla Verizon Communications mniejszy w poniedziałki i wtorki.

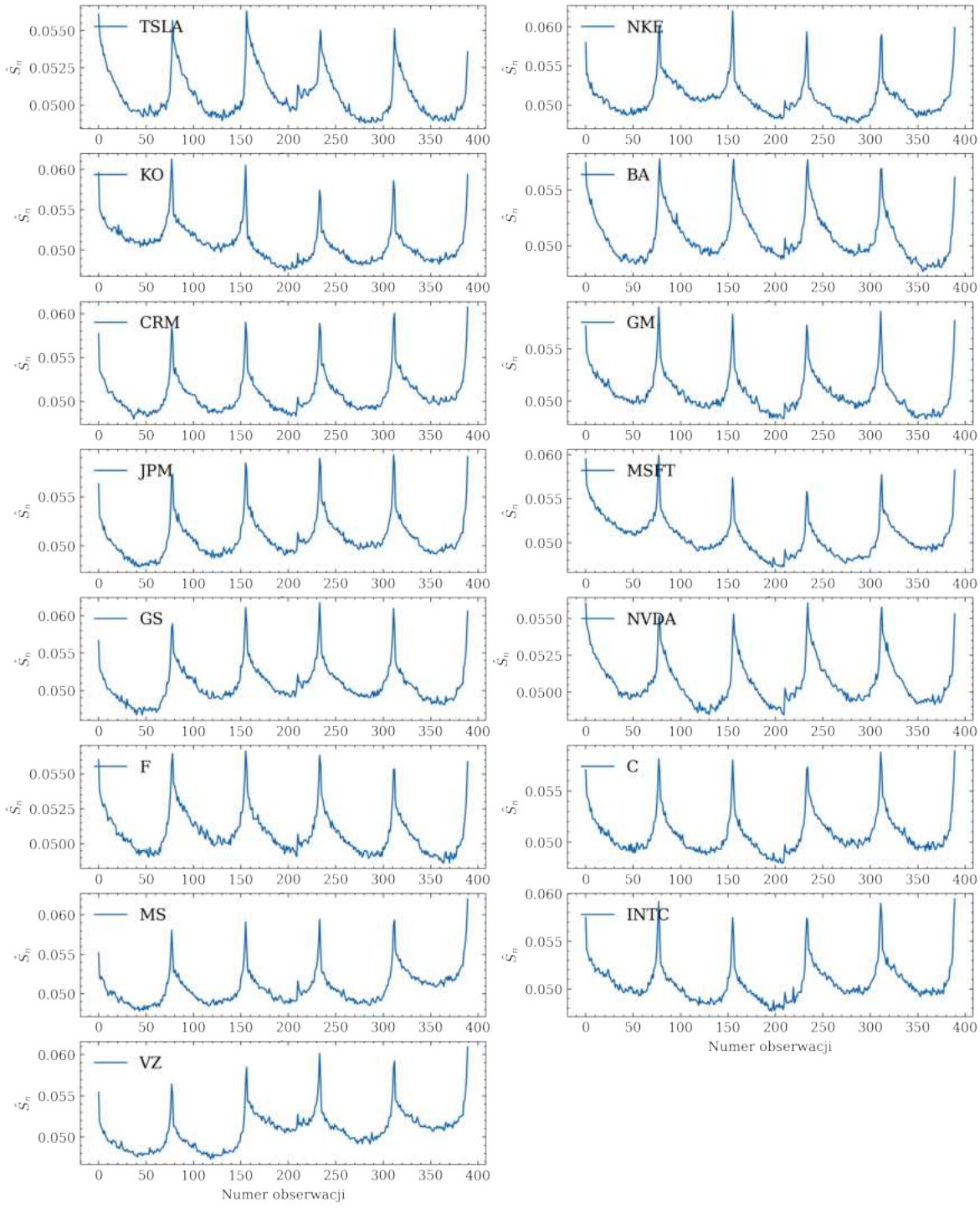
Analogiczna analiza została przeprowadzona dla kryptowalut. Wzorce sezonowości dla stóp zwrotu zostały przedstawione na rysunku 4.5. Można zaobserwować okresowość w wartościach zależną od pory dnia. Wzorce sezonowości dla wolumenu dla kryptowalut wybranych do badania są przedstawione na rysunku 4.6. Handel kryptowalutami odbywa się również w weekendy, dlatego te wzorce zostały obliczone dla 7 dni tygodnia. Na wykresach wzorców dla Bitcoina, Ethereum i Cardano (jedne z najbardziej popularnych kryptowalut) można zauważyć zdecydowanie większy wolumen na sesjach od poniedziałku do piątku niż w weekendy. Dla Litecoina i Dogecoina różnice nie są tak zauważalne. W przypadku kryptowalut nie występują bardzo duże skoki, takie jak dla spółek na początku i końcu sesji, ponieważ handel nigdy nie jest przerywany.

Wzorce sezonowości dla stóp zwrotu dla metali szlachetnych zostały przedstawione na rysunku 4.7. Na wykresach widoczna jest okresowość zależna od pory dnia. Handel walutami odbywa się na całym świecie i kształt tych wzorców odpowiada aktywności inwestorów na różnych rynkach świata. Rzetelne dane na temat wolumenu obrotu dla metali szlachetnych nie były dostępne dla rozdzielczości śróddziennej. Z tego powodu, wzorce sezonowości (wolumenu) dla metali szlachetnych nie były używane przy estymowaniu bazowych modeli ARMA-GARCH dla złota i srebra.

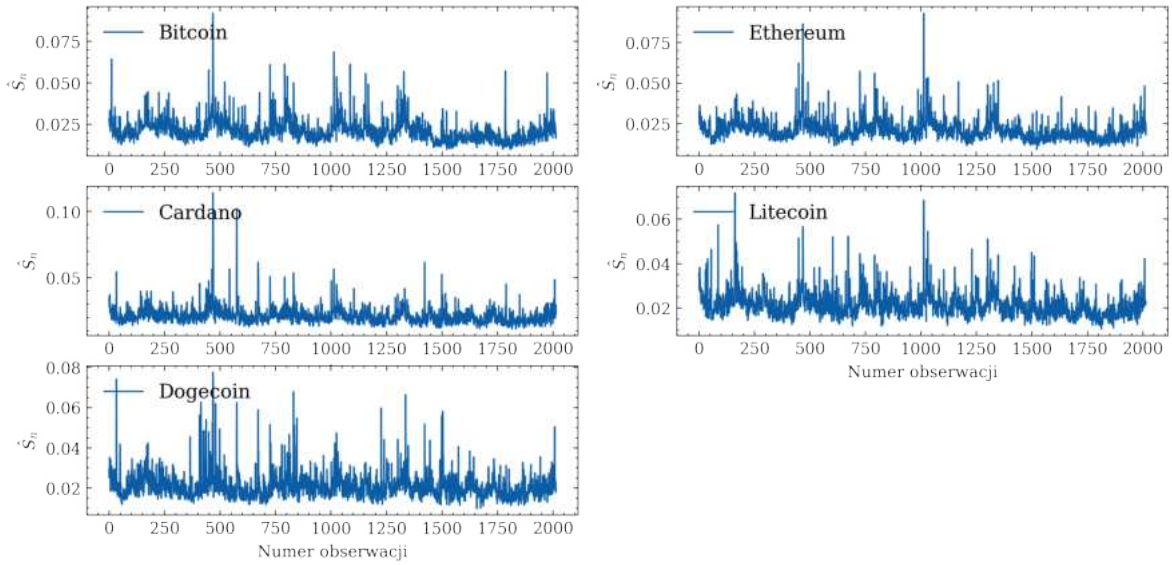
Bazowe modele ARMA-GARCH w niniejszej pracy estymowane są z wykorzystaniem stóp zwrotu, z których usunięta została składowa sezonowa poprzez podzielenie ich przez obliczony estymator śróddziennej składowej periodycznej. Obliczanie prognoz przy pomocy tak zbudowanych modeli polega na wyznaczeniu prognozy, a następnie



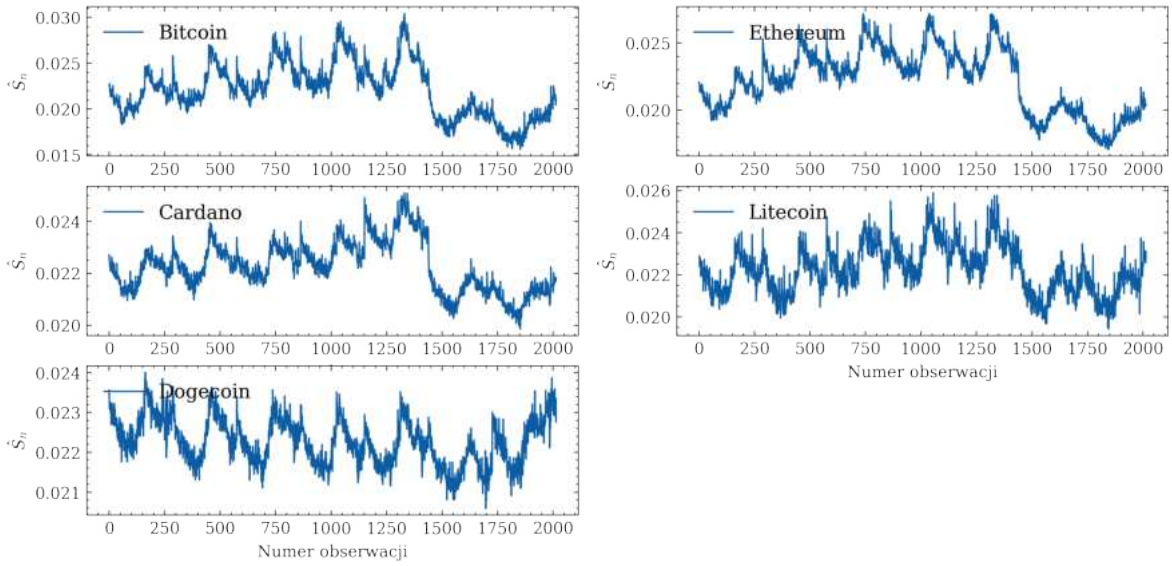
Rysunek 4.3. Wzorce sezonowości dla stóp zwrotu dla notowań spółek wybranych do badania dla każdego dnia tygodnia. Na wykresie można zaobserwować, że istotnie wyższe stopy zwrotu występują na początku każdej sesji giełdowej.
Źródło: Opracowanie własne na podstawie [106].



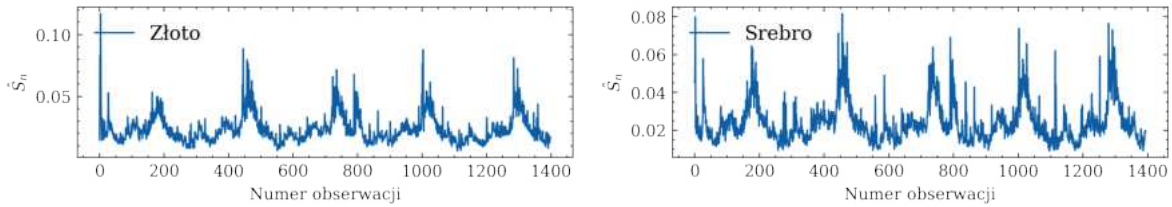
Rysunek 4.4. Wzorce sezonowości dla wolumenu dla notowań spółek wybranych do badania dla każdego dnia tygodnia. Na wykresie można zaobserwować, że istotnie wyższy wolumen występuje na początku i na końcu każdej sesji giełdowej.
Źródło: Opracowanie własne na podstawie [106].



Rysunek 4.5. Wzorce sezonowości dla stóp zwrotu dla kursów kryptowalut wybranych do badania dla każdego dnia tygodnia. Na wykresie można zaobserwować okresowość w wielkościach stóp zwrotu w zależności od pory dnia.
Źródło: Opracowanie własne na podstawie [106].



Rysunek 4.6. Wzorce sezonowości dla wolumenu dla kursów kryptowalut wybranych do badania dla każdego dnia tygodnia. Na wykresie można zaobserwować okresowość silniejszą niż w przypadku stóp zwrotu (Rys. 4.5).
Źródło: Opracowanie własne na podstawie [106].



Rysunek 4.7. Wzorce sezonowości dla stóp zwrotu dla kursów metali szlachetnych wybranych do badania dla każdego dnia tygodnia. Na wykresie można zaobserwować okresowość w wielkościach stóp zwrotu w zależności od pory dnia.
Źródło: Opracowanie własne na podstawie [106].

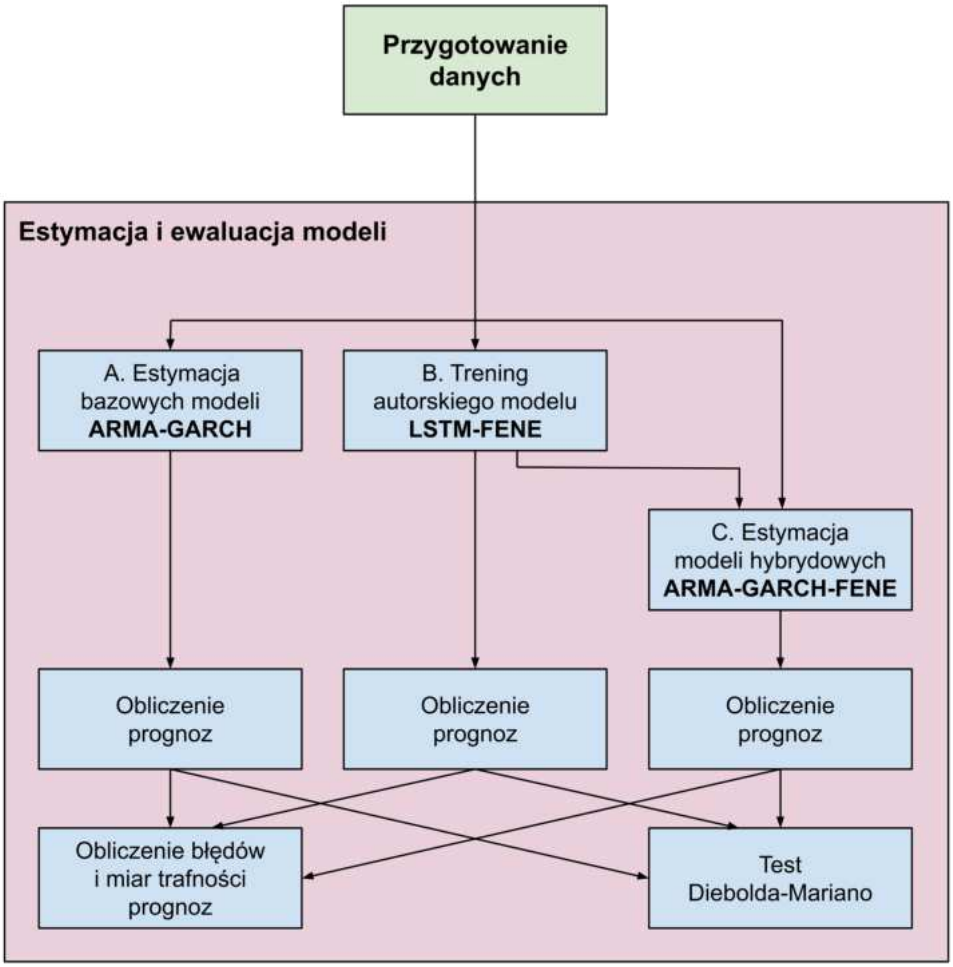
nałożeniu wzorca sezonowości, tj. przemnożeniu przez estymator śródddziennej składowej periodycznej. Modele oparte o sieci neuronowe obliczane są na danych pierwotnych (bez usuwania składowej periodycznej), ponieważ w ich przypadku sezonowość reprezentowana jest przez model automatycznie.

Tabela ze wszystkimi zmiennymi jest zapisywana w lokalnej bazie danych w formacie CSV. Pozwala to uniknąć każdorazowego dokonywania obliczeń przed rozpoczęciem modelowania. Autorski system do prognozowania notowań giełdowych bazuje na danych utworzonych w ten sposób do budowania każdego z modeli.

4.2.2. Estymacja i ewaluacja modeli

Bardzo istotną częścią systemu jest moduł służący do estymacji bazowych modeli ARMA-GARCH napisany w języku R (oznaczony literą A na rysunku 4.8). Wykorzystywany jest w nim pakiet *rugarch*, który umożliwia estymowanie różnych specyfikacji modeli wykorzystując też dane egzogeniczne. W niniejszych badaniach używane są wzorce sezonowości. Utworzone modele mogą być od razu wykorzystane do obliczenia (i zapisania w formacie CSV) prognoz.

Równolegle, z punktu widzenia przepływu danych, uruchamiany jest moduł budowania i trenowania autorskiego modelu LSTM-FENE (oznaczony literą B. na rysunku 4.8, zaś dokładny opis architektury modelu znajduje się z rozdziale 4.3). Ta część systemu jest stworzona w języku Python i bazuje głównie na bibliotece *TensorFlow*. Po wczytaniu danych muszą one zostać dodatkowo przygotowane dla sieci typu LSTM, tj. podzielone na krótsze szeregi czasowe o ustalonej długości, równej wybranemu maksymalnemu opóźnieniu danych. Budowa wykorzystywanego modelu została opisana dalej. Obliczenie prognoz dla walidacyjnego zbioru danych może być przeprowadzone bezpośrednio po wytrenowaniu modelu.



Rysunek 4.8. Schemat budowy systemu estymującego modele i weryfikującego ich jakość. Wszystkie komponenty wykorzystują dane przygotowane w pierwszym etapie działania systemu. Na ich podstawie stworzone zostają bazowe modele ARMA-GARCH, autorski model LSTM-FENE oraz hybrydowe modele ARMA-GARCH-FENE. Wszystkie typy modeli zostają wykorzystane do wygenerowania prognoz, a następnie porównania ich jakości poprzez obliczenie mierników trafności prognoz i wyników testu Diebolda-Mariano.
Źródło: Opracowanie własne.

Moduł FENE może być wykorzystywany do obliczenia wektora dodatkowych danych egzogenicznych modeli ARMA-GARCH. Jest to dodatkowy typ modelu oznaczony literą C na rysunku 4.8. Taka kombinacja modelowania tworzy nową część systemu pozwalającą zweryfikować, czy hybrydowe modele bazujące na klasycznych rozwiązaniach ekonometrycznych dają lepsze wyniki niż w pełni sieciowe modele. Moduł do Estymacji modeli hybrydowych również generuje i zapisuje prognozy umożliwiając ich późniejsze porównanie.

Prognozy dla każdego z badanych instrumentów uzyskane przy pomocy wszystkich typów modeli zostają porównane przez dwa końcowe moduły (napisane w języku Python). Pierwszy z nich oblicza błędy oraz miary trafności prognoz, drugi zaś przeprowadza test Diebolda-Mariano. Dzięki wynikom tych modułów możliwe jest wybranie najlepszego modelu dla każdego aktywa, a także weryfikacja hipotez badawczych.

Opisane części systemu mogą być uruchamiane indywidualnie i asynchronicznie. Część z wykonywanych obliczeń jest czasochłonna, dlatego dogodne jest wykorzystanie rozwiązań umożliwiających zrównoleglenie i przyspieszenie prac. Infrastruktura, na których prowadzone były obliczenia, została dobrana tak, aby zapewniać kompromis między kosztami a czasem oczekiwania na wyniki. Znaczna ilość danych powodowała, że etap estymacji nie był pomijalnie krótki w całym procesie badań. Zwłaszcza etap treningu sieci neuronowych wymaga użycia komputerów o dużej mocy obliczeniowej.

Do estymacji bazowych modeli ARMA-GARCH wykorzystany został serwis Google Cloud Compute Engine umożliwiający uruchomienie maszyn wirtualnych z szybkimi procesorami [71]. Biblioteka języka R, wykorzystywana do estymacji modeli bazowych, pochodzi z repozytorium udostępnionego wraz z publikacją Perlina M. et al [150]. Kod do estymacji wykorzystuje pakiet *rugarch* [62], którego nie da się uruchomić na maszynach z akceleratorami GPU lub TPU. Ze względu na prowadzenie obliczeń w schemacie rekursywnym konieczne było wielokrotne re-estymowanie modelu. Dla jednego instrumentu proces zajmował około 1 godziny w przypadku akcji oraz około 2-3 w przypadku metali szlachetnych i kryptowalut, dla których obliczenia prowadzone były dla większej liczby próbek.

Autorski model LSTM-FENE zbudowany na bazie architektury sieci neuronowych jest stosunkowo duży przez wykorzystanie dużego modelu językowego (moduł FENE). Model ten, w zależności od doboru hiperparametrów, składał się z 4-5 milionów parametrów. Praca z tak dużym modelem była możliwa dzięki użyciu chmurowego serwisu Google Colab [70], który umożliwia pracę z klastrami maszyn wyposażonych w GPU

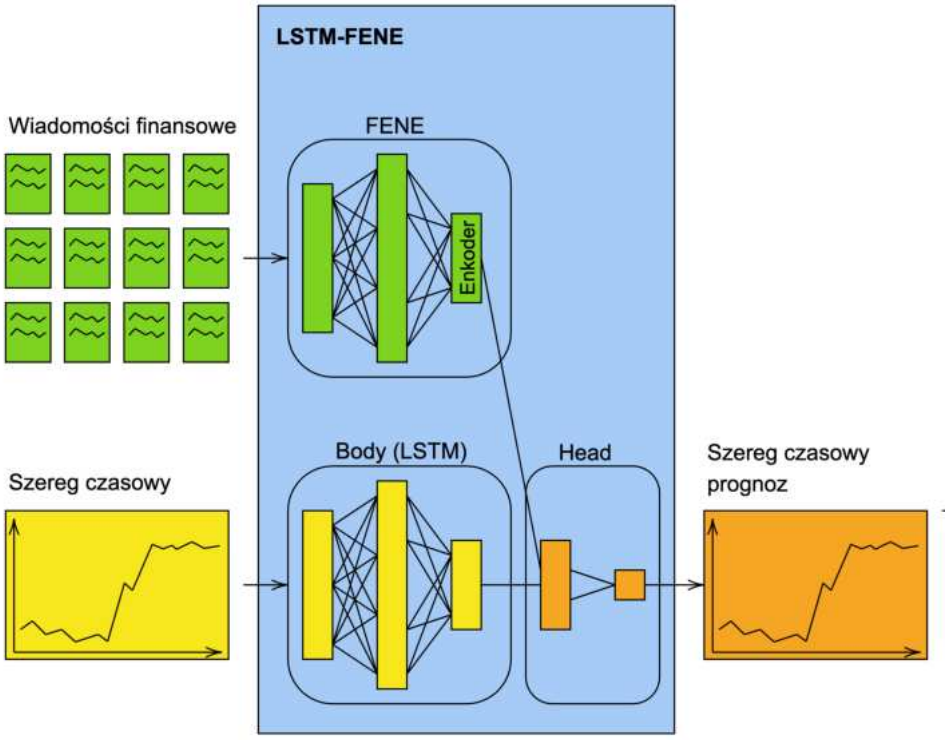
oraz TPU. W cyklu całych badań przeprowadzono kilkadziesiąt eksperymentów z różnymi architekturami sieci, których wytrenowanie zajmowało od 2 do 4 godzin.

Dużym wyzwaniem była estymacja hybrydowych modeli ARMA-GARCH-FENE, w których dane wejściowe składały się z oryginalnego szeregu czasowego logarytmicznych stóp zwrotu oraz wielowymiarowego wektora otrzymywanego z modułu FENE dla każdej chwili t_i. Dla każdego instrumentu zbadane zostały 3 wielkości modułu FENE. Obliczenia wykorzystywały pakiet *rugarch* i nie mogły być prowadzone z użyciem maszyn z akceleratorami GPU lub TPU. W efekcie estymacja dla każdej spółki trwała około 6, 12 i 24 godzin dla poszczególnych rozmiarów enkodera (w sumie około 42 godziny na instrument). Dla kryptowalut i metali szlachetnych było to po 12-24, 24-48 oraz 48-72 godziny (w sumie 84-144 godziny na instrument).

4.3. Propozycja modelu predykcji wspomaganej analizą treści wiadomości

Współczesne metody analizy tekstu pisanego przy użyciu głębokich sieci neuronowych, pozwalają w zautomatyzowany sposób ekstrahować najważniejsze informacje, odpowiadać na pytania zadane w otwarty sposób, a także tłumaczyć dowolne pary języków. Jednocześnie Internet dostarcza olbrzymiego strumienia danych – wiadomości z rynków finansowych, geopolitycznych, raportów spółek, postów w mediach społecznościowych. Wszystkie te dane mogą mieć krótko- jak i długoterminowy wpływ na zachowanie notowań giełdowych. Chan W. [22] zbadał różnice w zachowaniu notowań spółek w okresach, kiedy pojawiają się wiadomości na ich temat oraz w okresie, kiedy takich wiadomości nie ma. Zauważył istotne różnice pozwalające lepiej przewidzieć przyszłe ruchy cenowe na podstawie analizy treści opublikowanych wiadomości. Ilość źródeł danych, ich typów i form, jak i sama objętość tekstu, znacznie przekracza możliwości kognitywne człowieka. Odbiór tych informacji przez uczestników giełdy powoduje, że w końcu znajdują one swoje odzwierciedlenie w wycenach, jednak ich odpowiednio szybka analiza i reakcja może dawać przewagę nad innymi uczestnikami rynku.

Rysunek 4.9 przedstawia ogólną architekturę proponowanego w niniejszych badaniach modelu. Poza historycznymi przebiegami stóp zwrotu, wykorzystywane są również informacje zawarte w wiadomościach opublikowanych na temat spółek, dla których wyliczane są prognozy. Przedstawiony model składa się z trzech głównych modułów, oznaczonych kolorami na rysunku 4.9:



Rysunek 4.9. Ogólny schemat architektury proponowanego modelu wykorzystującego wiadomości finansowe LSTM-FENE. Model składa się z trzech głównych części: moduł FENE ekstrahuje informacje zawarte w wiadomościach finansowych, Body to sieć LSTM przyjmująca na wyjściu szereg czasowy logarytmicznych stóp zwrotu, Head (głowa modelu) łączy informacje obu poprzednich modułów i generuje prognozę.
Źródło: Opracowanie własne

— moduł FENE zaznaczony na zielono,
— część przetwarzająca finansowe szeregi czasowe zaznaczona na żółto (Body (LSTM)),
— głowa modelu zaznaczona na pomarańczowo (Head)).

Moduł FENE to głęboka sieć neuronowa, która powstaje przez wykorzystanie pre-trenowanego modelu językowego. Wykorzystywany w tym procesie bazowy model, to zwykle model językowy z maskowaniem, dla którego zadanie T_1 to odgadywanie słów, które zostały zamaskowane w danych treningowych (proces ten został dokładniej opisany w rozdziale 3.3.1). Z tak otrzymanego modelu usunięte zostają górne warstwy. Wówczas najwyższa z pozostałych warstw stanowi enkoder otrzymywanego tekstu przedstawiając go w reprezentacji, która może być dalej wykorzystana. Danymi wejściowymi jest pełen tekst pochodzący z artykułów opublikowanych na temat modelowanych spółek. Kluczowe jest, aby moment, w którym **moduł FENE** otrzymuje te dane, był zsynchronizowany z danymi w finansowych szeregach czasowych otrzymywanych przez drugą część modelu. Danymi wyjściowymi tej części jest wektor $\mathbf{v}_e$ przechowujący informacje wyekstrahowane z wiadomości. Optymalna długość tego wektora (czyli wielkość enkodera) jest jednym z przedmiotów badań w niniejszej pracy.

Część przetwarzająca finansowe szeregi czasowe (lub inaczej ciało modelu - ang. *Body*) jest odpowiedzialna za reprezentację wzorców czasowych, które pojawiają się w historycznych danych finansowych spółek. Ta część sieci neuronowej składa się z kilku warstw typu LSTM. Dane wejściowe mają postać wielowymiarowego szeregu czasowego zawierającego przede wszystkim logarytmiczne stopy zwrotu. Dodatkowymi wymiarami wejściowego wektora może być wolumen obrotu, bezwzględny poziom cen czy moment w czasie przedstawiony w postaci harmonicznej. Dane wyjściowe to wektor $\mathbf{v}_{ts}$ przedstawiający informacje, jakie ta część modelu może otrzymać na podstawie historycznych szeregów czasowych.

Funkcją **głowy modelu** (ang. *Head*) jest zebranie danych obliczonych przez opisane wyżej części i obliczenie prognozy. Dane wejściowe **głowy modelu** to po prostu otrzymane wcześniej i połączone ze sobą wektory $\mathbf{v}_e$ i $\mathbf{v}_{ts}$. Ta część modelu składa się z gęsto połączonych warstw, których wyjściem jest wektor prognoz o arbitralnej długości. Wielkością prognozowaną są wyłącznie logarytmiczne stopy zwrotu.

Tak zbudowany model wymaga szczególnego rodzaju treningu. Moduł FENE składa się z o kilka rzędów wielkości większej ilości parametrów niż część przetwarzająca szeregi czasowe. Wytrenowanie tak dużej sieci może zająć wiele tygodni nawet przy użyt-

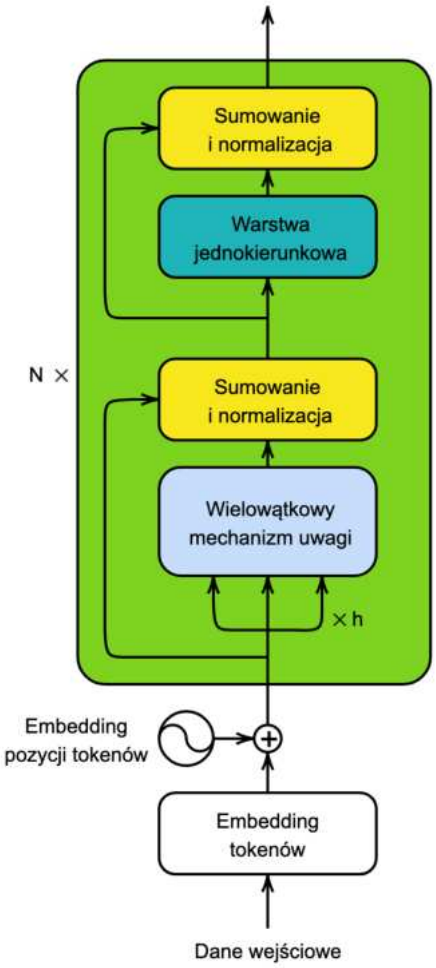
ciu bardzo dużych klastrów obliczeniowych. Wobec tego, korzystne jest wykorzystanie pre-trenowanych modeli językowych, a następnie użycie technik uczenia transferowego i *fine-tuningu*, które zostaną opisane w rozdziale 4.4. W tym procesie treningu ustalone zostaje zadanie T_2, które polega na równoległym wykorzystaniu danych tekstowych i historycznych szeregów czasowych do obliczenia prognozy dalszego przebiegu szeregu stóp zwrotów. Bardzo ogólnie można zatem określić zadanie T_2 jako regresję.

Zaproponowany w niniejszych badaniach model LSTM-FENE ma na celu wzbogacenie klasycznego podejścia do prognozowania stóp zwrotu na podstawie historycznych przebiegów cen instrumentów finansowych wraz z wolumenem obrotu. Dodatkowe informacje przekazywane do algorytmu to treść wiadomości, które odnoszą się do modelowanych spółek. Informacje te są dokładnie zsynchronizowane w czasie z przebiegiem poziomów cen, aby możliwe było zaobserwowanie efektu, jaki wywierają na dalsze trajektorie. Dane tekstowe nie mogą być bezpośrednio przetwarzane przez sieci neuronowe, wobec czego konieczne jest wykorzystanie sieci typu LM wraz z odpowiednim pre-processingiem.

Duże modele językowe wymagają znacznej mocy obliczeniowej i ogromnej ilości danych do zbudowania ich od podstaw. Mimo pozytywnego wpływu na gospodarkę i środowisko, jaki mogą mieć współczesne techniki NLP, należy ostrożnie podchodzić do zbędnego trenowania dużych modeli. Każdorazowy trening pochłania olbrzymie ilości energii i prowadzi do większego zanieczyszczenia środowiska. Estymuje się, że stworzenie modelu GPT-3 składającego się z 175 miliardów parametrów wymagało emisji około 500 ton dwutlenku węgla [122, 149]. Czas treningu GPT-3 z wykorzystaniem 1024 kart NVidia A100 wyniósł 34 dni. Wobec tak ogromnych kosztów czasowych jak i środowiskowych, odpowiedzialne podejście do badań rozwiązań NLP nakazuje wykorzystywać pre-trenowane modele, kiedy to tylko możliwe.

W niniejszych badaniach tekstowa część danych jest przetwarzana za pomocą ogólnodostępnych, pre-trenowanych modeli językowych o różnej wielkości. Modele te wykorzystują architekturę **enkodera opartego o mechanizm uwagi**, uznawany obecnie za najbardziej skuteczne podejście [20, 163]. Na rysunku 4.9 ta część modelu jest oznaczona na zielono.

Schemat budowy i działania enkodera został przedstawiony na rysunku 4.10 [39]. Pierwszym krokiem jest wstępne przetworzenie tekstów do postaci tokenów, a następnie obliczenie ich reprezentacji *word embeddings* (*embedding* tokenów) oraz *positional embeddings* (*embedding* pozycji tokenów), czyli struktury typowej dla mechanizmu uwagi



Rysunek 4.10. Schemat budowy enkodera w dużych modelach językowych. Dane wejściowe (tokeny) zostają przeliczone na ich reprezentacje *word embeddings* i wzbogacone o *embedding* pozycji. Tak utworzony wektor trafia do wielowątkowego mechanizmu uwagi, a później do jednokierunkowej warstwy (po każdym z tych kroków następuje normalizacja).
Źródło: Opracowanie własne na podstawie [39].

opisującej miejsce, jakie każdy token zajmuje w zdaniu. Wymiar otrzymanego wektora danych opisuje parametr d_{model}. Następnie wektor ten trafia na stos N identycznych struktur składających się z dwóch poziomów. Pierwszy poziom to **wielowątkowy mechanizmu uwagi** (opisany poniżej), gdzie h oznacza liczbę równoległych wątków. Dane wyjściowe są następnie sumowane z danymi wejściowymi tej warstwy oraz podlegają normalizacji, aby na wyjściu ponownie otrzymać wektor o wymiarze d_{model}. Drugi poziom to jedna warstwa gęsto połączonej jednokierunkowej sieci neuronowej, której dane wyjściowe są sumowane z danymi wejściowymi oraz normalizowane.

Mechanizm uwagi (funkcja uwagi) to mapowanie między zapytaniem (Q) oraz zbiorem par klucz-wartość (K, V), w którym zapytania, klucze i wartości są wektorami. Wynik mapowania jest ważoną sumą wartości, gdzie wagi, to odpowiedź funkcji kompatybilności między zapytaniem i odpowiednim kluczem. Większość najpopularniejszych zastosowań wykorzystuje funkcję uwagi zaimplementowaną z użyciem skalowanego iloczynu skalarnego, daną wzorem 4.3 i przedstawioną na rysunku 4.11a.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{4.3}$$

gdzie d_k jest wymiarowością zapytań Q i kluczy K, zaś $softmax$ to znormalizowana funkcja wykładnicza obliczona dla wektora o wymiarowości D:

$$softmax(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{d=1}^{D} e^{z_d}}. \tag{4.4}$$

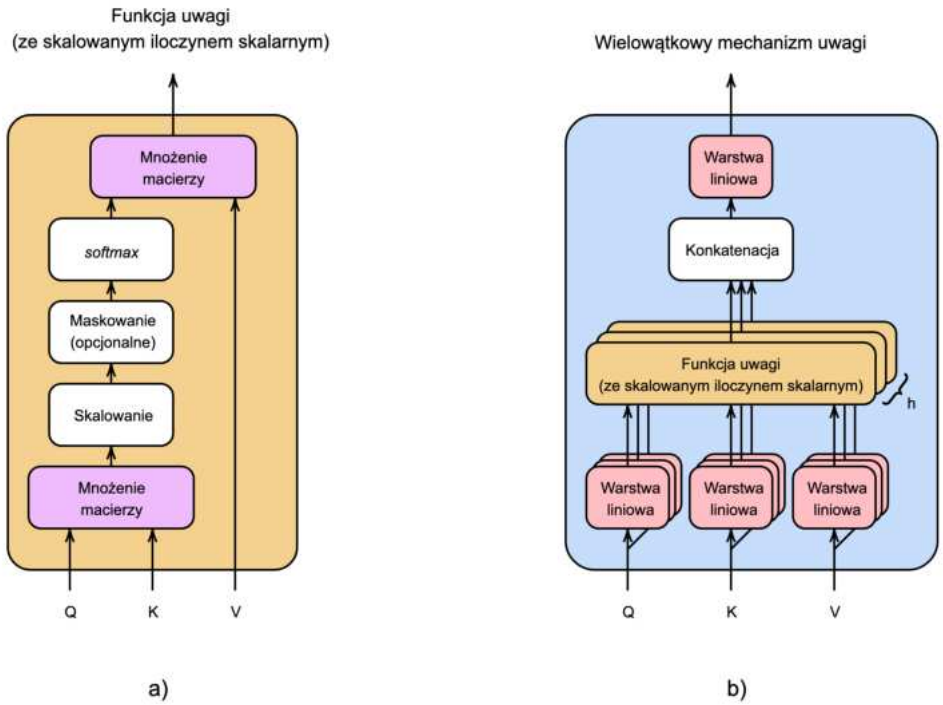
Wielowątkowy mechanizm uwagi polega na równoległym wykorzystaniu h instancji mechanizmu uwagi obliczonego dla takich samych danych wejściowych V, K, Q, a następnie połączeniu (konkatenacji) wektorów wejściowych, jak zostało to przedstawione na rysunku 4.11b. Matematycznie wielowątkowy mechanizm uwagi może być przedstawiony jako:

$$MultiHead(Q, K, V) = (head_1|...|head_h)W^O, \tag{4.5}$$

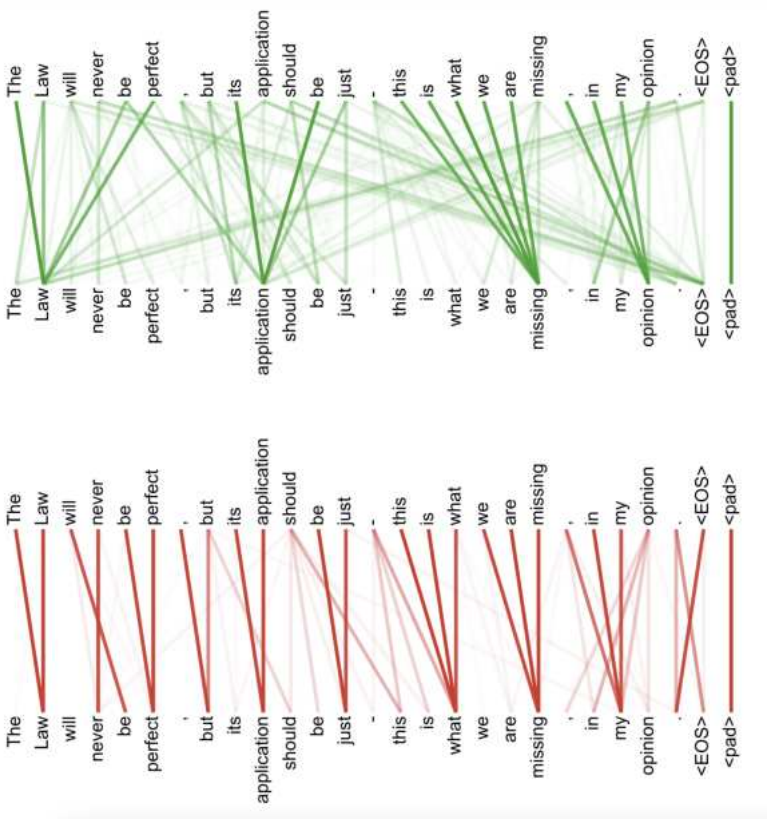
gdzie:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V), \tag{4.6}$$

gdzie W^O, W_i^Q, W_i^K W_i^V to macierze parametrów podlegających treningowi. Takie podejście pozwala, aby model wykorzystywał funkcje uwagi jednocześnie w wielu miejscach w zdaniu dla różnych reprezentacji tego tekstu. Rysunek 4.12 przedstawia wizu-



Rysunek 4.11. Schemat budowy mechanizmu uwagi w dużych modelach językowych. a) Funkcja uwagi składa się z mapowania zapytania Q i zbioru par klucz-wartość. W praktyce operacja ta realizowana jest przez mnożenie odpowiednich macierzy. b) Wielowątkowy mechanizm uwagi wykorzystuje wiele instancji (wątków) funkcji uwagi (zainicjalizowanych losowo różnymi współczynnikami), które następnie podlegają konkatenacji i trafiają do warstwy liniowej. Źródło: Opracowanie własne na podstawie [39].

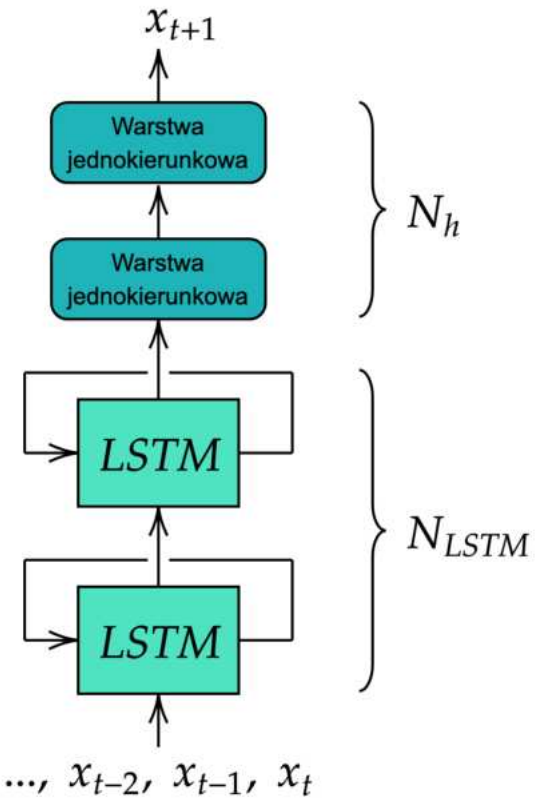


Rysunek 4.12. Przykład zależności wytrenowany w dwóch osobnych wątkach mechanizmu uwagi. Dzięki różnej (losowej) inicjalizacji każdego wątku, otrzymane są różne wyniki. W teorii pomaga to modelowi reprezentować różne aspekty zależności językowych. Źródło: [183].

alizację zależności wyuczonych przez dwa różne wątki mechanizmu uwagi w tym samym modelu.

Istnieje wiele modeli o architekturze bardzo podobnej do opisanej powyżej, różniących się wartościami hiperparametrów (N, h, d_k), danymi wykorzystanymi do wytrenowania modelu lub drobnymi szczegółami implementacji. Jedne z najpopularniejszych to modele BERT [39] i Small BERT [178]. W rozdziale 5 został opisany między innymi wpływ wyboru modelu językowego na jakość predykcji.

Model LSTM-FENE wykorzystuje historyczne przebiegi cen instrumentów finansowych jednocześnie z danymi tekstowymi. Na rysunku 4.9 część modelu odpowiedzialna za przetwarzanie finansowych szeregów czasowych oznaczona jest na niebiesko. Szczegółowy schemat architektury tego modułu przedstawia rysunek 4.13. Dane wejściowe to logarytmiczne stopy zwrotu, wolumen obrotu oraz sztuczne zmienne harmoniczne stworzone na podstawie danych czasowych (dzień roku i godzina w ciągu dnia). Tak utworzony wektor dla każdego okresu trafia do N_{LSTM} warstw typu LSTM, a następnie N_h jednokierunkowych warstw gęsto połączonych.



Rysunek 4.13. Schemat budowy części modelu przetwarzającej finansowe szeregi czasowe. Dane wejściowe (logarytmiczne stopy zwrotu, poziomy cen, wolumen obrotu, sztuczne zmienne harmoniczne) trafiają na N_{LSTM} warstw typu LSTM, odpowiedzialnych za znalezienie zależności w sekwencjach. Ich wynik trafia do N_h warstw gęsto połączonych.
Źródło: Opracowanie własne.

Ostatecznie wektor wyjściowy **modułu FENE** oraz wektor wyjściowy **części przetwarzającej finansowe szeregi czasowe** zostają połączone w jeden wektor, trafiający na ostatnią gęsto połączoną warstwę sieci neuronowej. Danymi wyjściowymi całego modelu jest prognoza jednopunktowa dla logarytmicznych stóp zwrotu, a zatem jest to model regresyjny.

Proces treningu zaproponowanego modelu odbywa się z wykorzystaniem techniki TL. W pierwszym kroku wszystkie parametry **modułu FENE** są zablokowane i nie podlegają treningowi - ich zadaniem jest jedynie przetwarzanie tekstu na wektory zrozumiałe przez resztę sieci neuronowej. Po wstępnym wytrenowaniu modelu, parametry **modułu FENE** zostają odblokowane, a trening zostaje wznowiony ze zmniejszonym współczynnikiem uczenia.

Przykłady pochodzące ze zbioru treningowego sieci neuronowej powinny być używane w losowej kolejności, zatem możliwa jest próba stworzenia wspólnego modelu dla wielu instrumentów. W rozdziale 5 zostały przedstawione porównania skuteczności dedykowanych modeli z modelami wspólnymi dla wielu instrumentów.

Hipotezy badawcze postawione we wstępie pracy zostały zweryfikowane przy pomocy tak zdefiniowanego modelu. Dla poszczególnych instrumentów finansowych wybrane zostały najlepsze instancje modeli, tj. z hiperparametrami dobranymi tak, aby prognoza była najdokładniejsza. Skuteczność każdego z modeli została zbadana w oparciu o zestaw wielu miar trafności prognoz przy wykorzystaniu rolowanego schematu prognozowania, które zostały przedstawione w rozdziale 4.4.

Proponowany w tym rozdziale model jest w całości zbudowany z części, które są różnego typu sieciami neuronowymi. Jednakże część odpowiedzialna za obróbkę finansowych szeregów czasowych może być zastąpiona modelem typu ARMA-GARCH, który wykorzystuje wektor v_e jako dodatkowe dane egzogeniczne. Wyniki eksperymentów z taką architekturą i jej porównanie z modelem złożonym w całości z warstw sieci neuronowej będą przedstawione w rozdziale 6.

4.4. Trening i walidacja modeli

4.4.1. Podział na zbiór treningowy, walidacyjny i testowy

Modele ARMA-GARCH dzięki względnej prostocie (liniowości) pozwalają na ocenę i interpretację poszczególnych współczynników oraz na ocenę dopasowania modelu do

danych empirycznych poprzez ocenę reszt modelu. Analiza reszt modelu możliwa jest m.in. przy wykorzystaniu funkcji ACF i PACF (patrz rozdział 2.1.1). W przypadku bardziej złożonych modeli uczenia maszynowego, które często bazują na przekształceniach nieliniowych, tego typu analiza jest trudniejsza do przeprowadzenia i zinterpretowania.

Mimo tych komplikacji bezpośrednie porównanie modeli różnych klas jest możliwe przez porównanie prognoz uzyskanych przy ich użyciu, z rzeczywistymi wartościami. Miarą efektywności P dla modeli prognozowania jest zwykle błąd prognozy (im mniejszy, tym lepszy) dla ustalonego horyzontu czasowego.

Ocena jakości prognoz pozwala na wytrenowanie serii modeli i wyboru najlepszego z nich. W tym celu zbiór dostępnych obserwacji dzieli się na 2 lub 3 podzbiory:

1. **Zbiór treningowy** - W czasie procesu trenowania (estymowania) modelu zostają znalezione parametry, które najdokładniej dopasowują go do danych treningowych. Porównanie predykcji modelu dla danych wejściowych, na których był trenowany, pozwala jedynie stwierdzić czy proces treningu znalazł pewne (lokalne) optimum, jednak nie daje informacji, czy predykcja na nowych próbkach będzie równie skuteczna.
2. **Zbiór walidacyjny** - Celem zbioru walidacyjnego jest dobranie *hiperparametrów* modelu, np. wielkość kolejnych warstw sieci neuronowych, rzędów opóźnień modeli ARMA-GARCH. Po wytrenowaniu modelu na zbiorze treningowym, zostaje on użyty do wygenerowania predykcji dla danych wejściowych pochodzących ze zbioru walidacyjnego, które następnie są porównywane ze znanymi zmiennymi modelowanymi przy użyciu miary P. Hiperparametry dobiera sie tak, aby zmaksymalizować miarę P (zminimalizować błąd prognozy) dla zbioru walidacyjnego, przy czym trening zawsze odbywa się na zbiorze treningowym. W przypadku modeli nie posiadających hiperparametrów (lub posiadających niewielką ich liczbę), czasami pomija się tworzenie zbioru walidacyjnego.
3. **Zbiór testowy** - Model z hiperparametrami maksymalizującymi miarę P na zbiorze walidacyjnym zostaje wybrany jako model docelowy. W celu ostatecznej oceny jakości prognoz, obliczana jest miara efektywności P dla predykcji przy użyciu zbioru testowego jako danych wejściowych.

Obliczone miary P można uznać za skuteczne tylko, jeśli ww. zbiory danych są niezależne i posiadają jednakowy rozkład prawdopodobieństwa (są wygenerowane przez ten sam proces).

Rysunek 4.14. Podział danych na zbiory: treningowy, walidacyjny i testowy. W zbiorze treningowym umieszcza się zwykle około 80% wszystkich obserwacji. Pozostałą część obserwacji dzieli się równo pomiędzy zbiór walidacyjny i testowy. Jeżeli obserwacji jest wystarczająco dużo, można zwiększyć ich procentowy udział w zbiorze treningowym.
Źródło: Opracowanie własne.

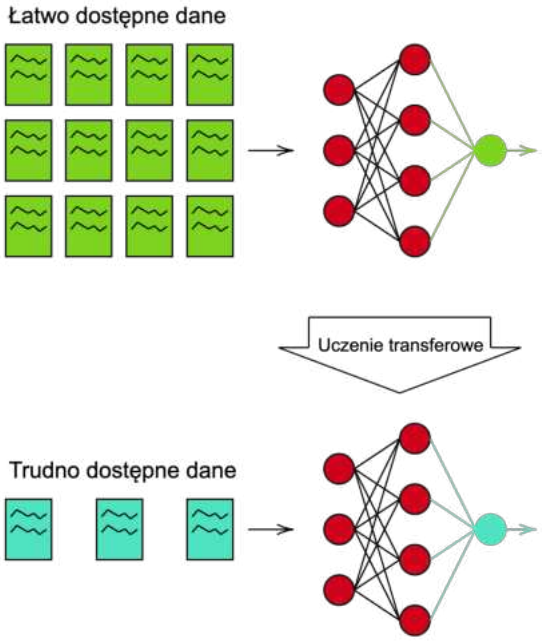
W pierwszym kroku ustalone zostają relatywne wielkości zbiorów danych. Zwykle zbiór treningowy ustala się na 60-80% wszystkich próbek, zaś reszta zostaje podzielona równo pomiędzy zbiór walidacyjny (o ile jest wykorzystywany) i testowy. W przypadku, gdy obserwacji jest bardzo dużo, zbiór treningowy może być jeszcze większy.

Następnie każdą z obserwacji należy przypisać do konkretnego i tylko jednego zbioru. W wielu zastosowaniach próbki zostają przydzielone całkowicie losowo, jednak w przypadku szeregów czasowych dane treningowe muszą pochodzić z okresu poprzedzającego dane walidacyjne, zaś te muszą pochodzić z okresu poprzedzającego dane testowe. Ma to na celu uniknięcie walidacji i testowania modelu na danych zawierających przyczyny, których skutki znalazłyby się w zestawie treningowym. Oprócz tego, większość modeli szeregów czasowych bazuje na jednostajnym próbkowaniu i nie można dopuścić do powstawania brakujących obserwacji z powodu przypisania ich do innego zbioru [93]. Wobec tych założeń, zbiór wszystkich dostępnych obserwacji zostaje podzielony jak przedstawiono na rysunku 4.14 [9].

4.4.2. Uczenie transferowe

Często pojawiającym się problemem przy tworzeniu modeli ML jest brak wystarczającej ilości danych treningowych. Jedną z metod radzenia sobie z nim jest TL. Technika ta polega na wytrenowaniu modelu (zwykle sieci neuronowej) z zadaniem T na innych, łatwiej dostępnych danych, a następnie wykorzystaniu jego pre-trenowanej części do stworzenia nowego modelu z zadaniem T', jak przedstawiono na Rysunku 4.15.

W ogólnym przypadku przestrzeń, z której pochodzą dane $\mathcal{X}_1$ oraz rozkład brzegowy dostępnych danych $P(X_1), X_1 = \{x_{1,i}\}, x_{1,i} \in \mathcal{X}_1$ określają domenę $\mathcal{D}_1$ jako:



Rysunek 4.15. Schemat działania uczenia transferowego (TL). Początkowo model trenowany jest przy wykorzystaniu dużej ilości łatwo dostępnych danych. W kolejnym kroku docelowe, trudniej dostępne dane zostają wykorzystane do *fine-tuningu* modelu.
Źródło: Opracowanie własne.

$$\mathcal{D}_1 = \{\mathcal{X}_1, P(X_1)\}. \tag{4.7}$$

Zadanie T_1 dla domeny $\mathcal{D}_1$ jest określone przez przestrzeń modelowaną $\mathcal{Y}_1$, funkcję predykcji f przewidującą wartość modelowaną $y_{1,i} \in \mathcal{Y}_1$ na podstawie danych wejściowych $x_{1,i} \in \mathcal{X}_1 : f(x_{1,i}) = y_{1,i}$:

$$T_1 = \{\mathcal{Y}_1, P(Y_1|X_1)\} = \{\mathcal{Y}_1, f\}, Y_1 = \{y_{1,i}\}, y_{1,i} \in \mathcal{Y}_1. \tag{4.8}$$

Dla domeny źródłowej $\mathcal{D}_1$ i odpowiadającemu jej zadaniu T_1 oraz domenie docelowej $\mathcal{D}_2$ i jej zadaniu T_2, celem uczenia transferowego jest znalezienie prawdopodobieństwa warunkowego $P(Y_2|X_2)$ w domenie $\mathcal{D}_2$ wykorzystując informacje zdobyte przez trenowanie modelu na domenie $\mathcal{D}_1$ i zadaniu T_1 [192].

4.4.3. *Fine-tuning* modeli

W przypadku wykorzystania uczenia transferowego znalezienie dobrze dostępnego, bogatego zestawu danych z domeny $\mathcal{D}_1$, która jest wystarczająco podobna do docelowej domeny $\mathcal{D}_2$, jest dopiero pierwszym z koniecznych kroków do otrzymania dobrego mo-

delu. Wyjątkowo istotny jest również sam proces trenowania modelu do wykonywania zadania T_2 [132].

Fine-tuning modelu (najczęściej głębokiej sieci neuronowej) zaczyna się od otrzymania modelu bazowego M_1 w procesie pre-trenowania, czyli trenowania do wykonywania zadania T_1 wykorzystując przy tym obserwacje z (łatwo dostępnej) domeny $\mathcal{D}_1$. Przykładem może być głęboka, konwolucyjna sieć neuronowa, której danymi wejściowymi były zdjęcia zwierząt, zaś zadaniem była klasyfikacja na dosyć ogólne kategorie ("pies", "kot", "ptak", itp.). Otrzymany model M_1 składa się z L warstw różnego typu, przy czym najwyższe warstwy są zwykle gęsto połączone i odpowiadają warunkowemu prawdopodobieństwu przypisywanemu każdej z dostępnych kategorii dla każdego ze zdjęć.

W kolejnym kroku na podstawie modelu M_1 budowany jest model M_2, co schematycznie przedstawia rysunek 4.15. Z bazowej sieci neuronowej zostaje usunięta arbitralna, często wyznaczana eksperymentalnie, ilość górnych warstw i zostaje zastąpiona nowymi warstwami (niekoniecznie z tą samą ich ilością) z losową inicjalizacją, które są przystosowane do realizacji zadania T_2. Tak otrzymany model musi zostać ponownie wytrenowany z wykorzystaniem danych pochodzących z docelowej domeny $\mathcal{D}_2$. Nawiązując do przykładu użytego powyżej, domeną $\mathcal{D}_2$ mogłyby być zdjęcia rzadkich gatunków ptaków, które z założenia sporadycznie będą występować w danych treningowych modelu M_1, a zadanie T_2 to klasyfikacja każdego ze zdjęć na konkretny gatunek występującego na nim ptaka. Model M_1 w czasie pre-treningu uczy się rozpoznawać takie elementy, jak oczy, dzioby, pióra czy nogi, a zatem jest to wiedza użyteczna również w nowym zadaniu T_2.

Analogicznie w dziedzinie NLP wykorzystuje się duże, pre-trenowane na ogólnodostępnych danych sieci neuronowe, tzw modele językowe. Funkcją modelu językowego jest ogólne wychwycenie struktur występujących w danych tekstowych, które po przetworzeniu przez dolne warstwy, mogą być wykorzystane do realizacji zadania T_2 za pomocą nowej struktury górnych warstw.

Najprostszym i najmniej wymagającym obliczeniowo sposobem, aby wykorzystać LM, to zastosowanie go w "zamrożonej" postaci. Dopasowanie modelu w tym reżimie polega na trenowaniu jedynie parametrów w nowych warstwach sieci neuronowej, traktując cały LM jako zaawansowane *embeddingi* kontekstu zawartego w tekście (ang. *contextual embeddings*). Podejście to jest stosowane, gdy dostępna moc obliczeniowa jest ograniczona lub gdy ilość dostępnych danych dla zadania T_2 nie pozwala na skutecz-

ne trenowanie dużych modeli. Mimo tak ograniczonego podejścia, dzięki wykorzystaniu zaawansowanych modeli językowych, otrzymane efekty są lepsze niż dotychczasowych modeli, np. dla analizy sentymentu [115].

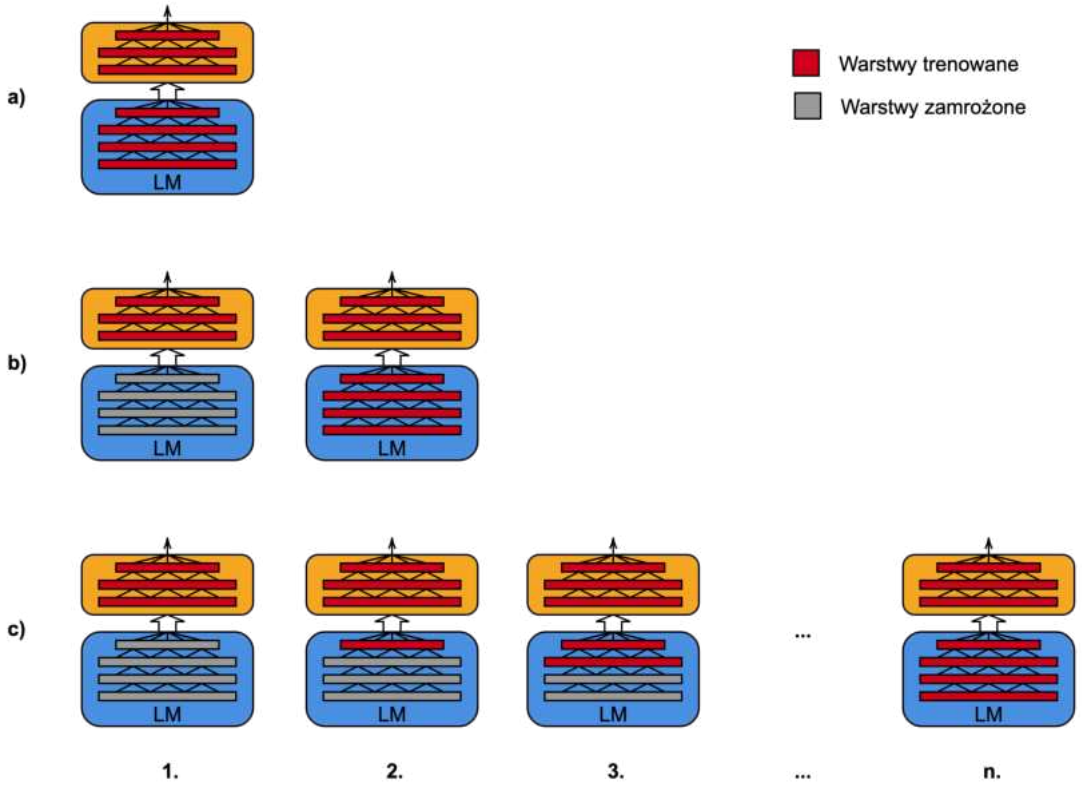
W przypadku, gdy dane dla zadanie T_2 są bardziej liczne, a dostępna moc obliczeniowa większa, lepsze wyniki można uzyskać przez *fine-tuning* nowej części sieci neuronowej wraz z modelem językowym - całym lub jego wybranymi warstwami. Podejście to sprawdza się szczególnie dobrze, gdy zadanie T_2 jest bardziej skomplikowane i wymaga predykcji struktury wewnątrz tekstu, np. NER [168]. Proces treningu modelu z *fine-tuningiem* może być przeprowadzony całościowo lub etapami (rysunek 4.16).

Pełny trening modelu (przedstawiony na rysunku 4.16a) nie różni się od zwyczajnego treningu. W takim przypadku model zbudowany jest z pre-trenowanego modelu LM oraz nowej części, która wcześniej nie podlegała treningowi. Od samego początku wagi wszystkich warstw sieci neuronowej są aktualizowane.

W przypadku trenowania etapami w pierwszym kroku część LM zostaje "zamrożona" i zajmuje się wyłącznie przetwarzaniem i przekazywaniem dalej danych wejściowych, zaś treningowi podlega jedynie dodana część modelu. W kolejnym kroku "odmrożona" zostaje cała reszta architektury, a trening jest kontynuowany od stanu osiągniętych wcześniej wag (rysunek 4.16b). Otrzymywane w ten sposób rezultaty są zwykle lepsze, ponieważ w procesie treningu nie ma momentu, w którym część warstw musi być aktualizowana dużo mocniej od reszty (co dzieje się, gdy część modelu nie podlegała wcześniej uczeniu, jak w przypadku a)). W niniejszych badaniach wykorzystywany był *fine-tuning* dwuetapowy.

W niektórych przypadkach zaleca się "odmrażanie" części LM w kilku krokach (rysunek 4.16c). Wyniki osiągane w ten sposób są zwykle najlepsze, jednak cały proces jest znacznie wydłużony i kosztowny. Dokładniejszy opis *fine-tuningu* modeli językowych można znaleźć m.in. w publikacji Howarda J. i Rudera S. [86].

W praktyce najczęściej wykorzystuje się duże zbiory danych udostępnione przez innych badaczy (z odpowiednimi licencjami) lub nawet gotowe modele. Repozytoria takich modeli i zbiorów danych, szczególnie dla dziedziny NLP zostały opisane w rozdziale 3.4.3.



Rysunek 4.16. Możliwe warianty *Fine-tuningu* modeli NLP opartych o bazowy model LM. Warianty te różnią się liczbą etapów, w których odmrażane są warstwy pre-trenowanego modelu LM. a) Pełny trening modelu.(wszystkich warstw jednocześnie) b) Dwuetapowy trening, w którym *fine-tuning* odbywa się jako osobna faza. c) Trening, w którym *fine-tuning* polega na odmrażaniu kolejnych warstw w następujących po sobie etapach.
Źródło: Opracowanie własne.

4.5. Budowa i ocena krótkookresowych prognoz stóp zwrotu i cen analizowanych instrumentów finansowych

4.5.1. Badane instrumenty finansowe

W przypadku analizy danych śródziennych konieczne jest ustalenie częstotliwości obserwacji cen instrumentów. Najbardziej dokładnym odwzorowaniem rynku byłaby analiza uwzględniająca dokładny czas każdego pojedynczego zlecenia. W takim przypadku należałoby zagłębić się w mikrostrukturę rynku, a częstotliwość nie byłaby stała, co uniemożliwiałoby wykorzystanie większości klasycznych modeli ekonometrycznych. W związku z tym, dane agreguje się do zadanej z góry częstotliwości. Wysokie częstotliwości (na przykład 1-minutowa) skutkują występowaniem bardzo silnej autokorelacji. Częstotliwości niższe mogą dawać gorszą jakość modeli prognozujących. Dodatkowym problemem technicznym jest duża złożoność i czas przygotowania modeli dla wysokich częstotliwości. W publikacji z 2010 roku Będowska-Sójka B. i Kliber A. zalecają wykorzystanie częstotliwości 5-minutowej [106]. **Wobec powyższych argumentów w niniejszej pracy dane dla wszystkich instrumentów i modeli zostały zagregowane do częstotliwości 5-minutowej**.

Trzon badań niniejszej pracy stanowi prognozowanie stóp zwrotu akcji spółek wykorzystujące informacje zawarte w wiadomościach finansowych. W celu zbadania czy zaproponowana metodologia ma również zastosowanie dla innych typów instrumentów finansowych, identyczne analizy (dla tego samego zakresu dat i 5-minutowej częstotliwości) zostały wykonane dla pięciu bardziej medialnych kryptowalut, a także dwóch metali szlachetnych - złota i srebra.

W przypadku akcji do badania zostały wykorzystane notowania spółek wchodzących w skład indeksu S&P 500. Są to duże i znane spółki - ich akcje charakteryzują się dużą płynnością, a ze względu na ich powszechną popularność, wiadomości na ich temat są publikowane często i są stosunkowo łatwe do pozyskania. Okres, z którego pochodzą dane dla każdej ze spółek, to 2021-03-10 do 2022-12-30, tj. 503 dni z otwartą sesją giełdową.

Do badań wybrano spółki o różnym charakterze. Część z nich to firmy, których usługi i produkty trafiają do klientów detalicznych, np. Apple. Wówczas znaczna część wiadomości dotyczy produktów i dóbr szeroko opisywanych w różnego typu portalach internetowych. W przypadku firm świadczących usługi typu B2B większość publika-

Tabela 4.1. Całkowita i średnia dzienna liczba zgromadzonych wiadomości na temat badanych spółek.

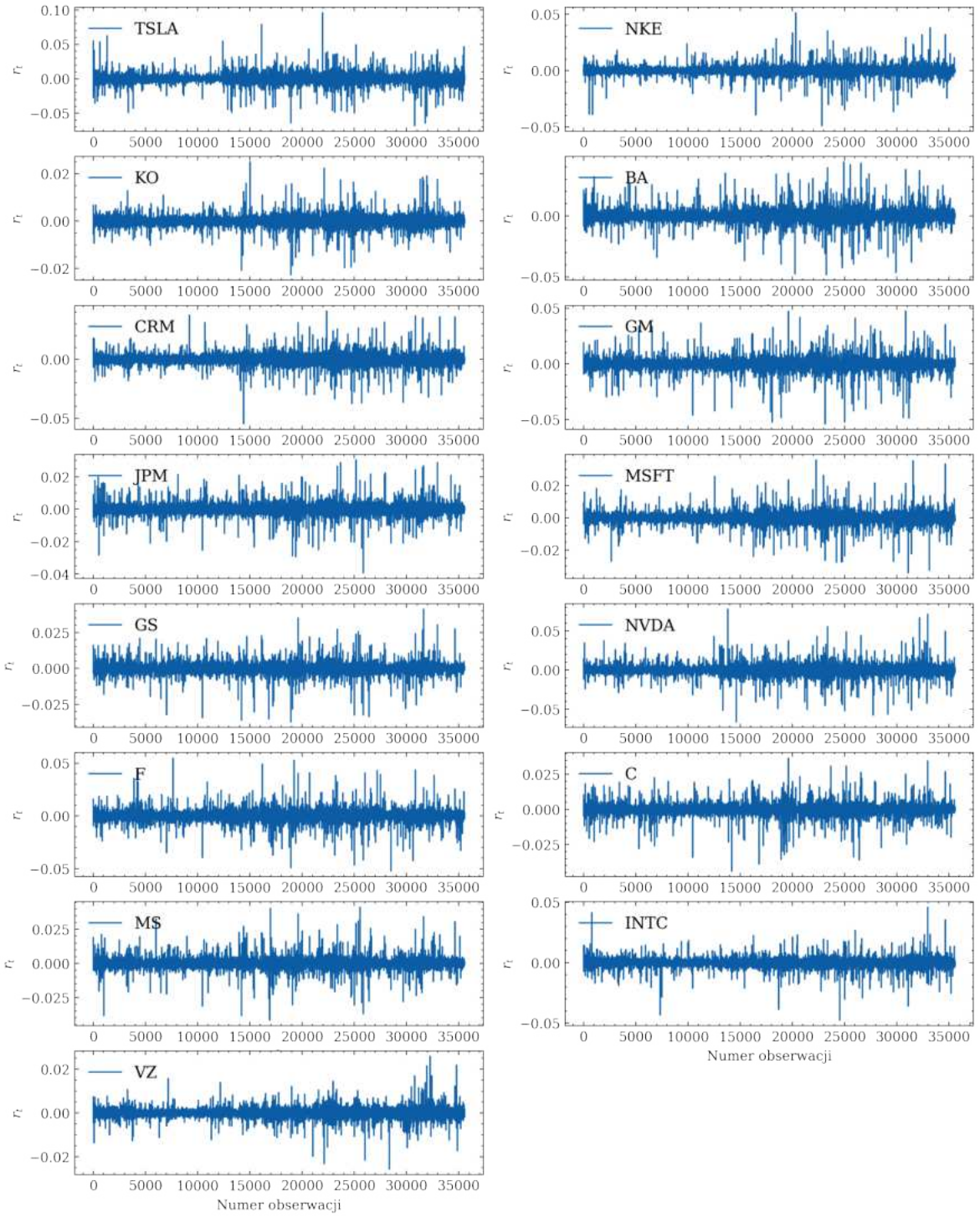
Spółka	Nazwa spółki	Liczba wiadomości	
		całkowita	średnia dzienna
TSLA	Tesla Inc.	3151	6.26
MSFT	Microsoft Corporation	1988	3.95
NVDA	NVIDIA Corporation	787	1.56
JPM	JPMorgan Chase & Co	774	1.54
GM	General Motors Company	736	1.46
BA	Boeing Company	591	1.17
NKE	NIKE Inc. Class B	434	0.86
INTC	Intel Corporation	423	0.84
GS	Goldman Sachs Group Inc.	375	0.75
C	Citigroup Inc.	357	0.71
F	Ford Motor Company	339	0.67
KO	Coca-Cola Company	266	0.53
VZ	Verizon Communications Inc.	261	0.52
CRM	salesforce.com inc.	245	0.49
MS	Morgan Stanley	231	0.46

Źródło: Opracowanie własne.

cji ma charakter czysto inwestycyjny. Większe i bardziej znane spółki są też chętniej opisywane.

Źródłem historycznych notowań giełdowych jest serwis Alpa Vantage [3]. URLe oraz nagłówki wiadomości finansowych są otrzymywane z serwisu CityFALCON [27], a następnie zostają pobrane ze stron internetowych pierwotnych źródeł (portali informacyjnych). Proces pobierania i przygotowywania danych do modelowania został opisany dokładniej w rozdziale 4.2.1. Tabela 4.1 przedstawia ilość publikacji zgromadzonych na temat spółek wybranych do badań.

Dla notowań akcji wybranych spółek zmienną modelowaną jest logarytmiczna stopa zwrotu, obliczona dla okresów w czasie otwarcia giełdy i z pominięciem zwrotów nocnych. Szeregi dla wybranych spółek zostały przedstawione na rysunku 4.17 dla pełnego zakresu dat. Dla kilku ze spółek (TSLA, NKE, KO, CRM, MSFT, NVDA, VZ) można zaobserwować mniejszą zmienność na początku okresu, która zwiększa się w okolicy października-listopada 2022 (około 12000-14000 próbek). Mniejsza zmienność pokrywa się z okresem, kiedy cały indeks S&P 500 znajdował się w trendzie wzrostowym, zaś zwiększona zmienność pojawia się w późniejszym okresie spadków. Dla wszystkich spółek widoczne są dłuższe okresy, w których zmienność utrzymywała się na podobnym poziomie, a także pojawianie się obserwacji nietypowych.



Rysunek 4.17. Szeregi czasowe logarytmicznych stóp zwrotu dla spółek uwzględnionych w badaniach.
Źródło: Opracowanie własne.

Tabela 4.2. Statystyki opisowe szeregów zwrotów 5-minutowych z akcji spółek wybranych do badania.

Spółka	Średnia	Odch. stand.	Min.	Maks.	Skośność	Kurtoza	Liczba obserwacji
TSLA	-1.66e-05	0.00410	-0.0682	0.0957	0.222	41.5	35576
MSFT	-8.34e-07	0.00192	-0.0342	0.0359	-0.204	40.0	35576
NVDA	4.59e-06	0.00381	-0.0665	0.0779	0.065	40.1	35576
JPM	-3.31e-07	0.00190	-0.0393	0.0304	-0.132	40.0	35580
GM	-9.94e-06	0.00287	-0.0539	0.0475	-0.886	51.8	35580
BA	-3.46e-06	0.00282	-0.0482	0.0440	0.018	39.2	35580
NKE	-1.26e-05	0.00215	-0.0489	0.0512	-0.374	61.2	35580
INTC	-1.74e-05	0.00214	-0.0476	0.0460	-0.670	47.5	35576
GS	1.94e-06	0.00211	-0.0372	0.0412	-0.342	43.7	35580
C	-1.58e-05	0.00220	-0.0439	0.0365	-0.866	43.9	35580
F	3.66e-06	0.00300	-0.0523	0.0548	0.100	41.4	35580
KO	4.98e-06	0.00127	-0.0227	0.0250	0.174	38.5	35580
VZ	-4.53e-06	0.00128	-0.0257	0.0259	-0.228	37.0	35580
CRM	-1.50e-05	0.00254	-0.0548	0.0410	-0.343	37.6	35580
MS	1.67e-06	0.00223	-0.0416	0.0412	0.095	47.6	35580

Źródło: Opracowanie własne.

Tabela 4.2 przedstawia podstawowe statystyki opisowe zmiennej modelowanej (logarytmicznych stóp zwrotu) dla wybranych spółek. Średnia stopa zwrotu dla wszystkich spółek jest bliska zera.

Większe odchylenie standardowe widoczne jest szczególnie dla stóp zwrotów Tesla Inc. (TSLA), NVIDIA Corporation (NVDA) oraz Ford Motor Company (F). Również dla tych spółek minimalna i maksymalna zaobserwowana wielkość mają największe wartości bezwzględne.

Stosunkowo duża ujemna skośność charakteryzuje stopy zwrotów General Motors Company (GM), Citigroup Inc. (C) oraz Intel Corporation (INTC). Skośność dla pozostałych spółek ma wartość bezwzględną poniżej 0,5. Kurtoza ma znaczną wielkość dla wszystkich spółek, co oznacza stosunkowo często pojawianie się obserwacji odstających.

Liczba obserwacji jest niemal jednakowa dla wszystkich szeregów. W kilku przypadkach serwis Alpha Vantage dostarczył szeregi historycznych notowań z brakującymi pojedynczymi obserwacjami. Punkty te zostały pominięte (nie zostały uzupełnione wartością poprzednią lub średnią).

Kryptowaluty, które znalazły się w niniejszych badaniach również zostały wybrane ze względu na ich dużą popularność. Zasady, na których oparte jest działanie Bitcoina, Ethereum, Cardano, Litecoina i Dogecoina, są dość podobne, choć różnią się technicznymi szczegółami. Mimo to, z punktu widzenia inwestora mogą być postrzegane dość

Tabela 4.3. Całkowita i średnia dzienna ilość zgromadzonych wiadomości na temat badanych kryptowalut.

Kryptowaluta	Ilość wiadomości	
	całkowita	średnia dzienna
Bitcoin	13239	20.02
Ethereum	4371	6.61
Dogecoin	1650	2.51
Cardano	1041	1.57
Litecoin	306	0.46

Źródło: Opracowanie własne.

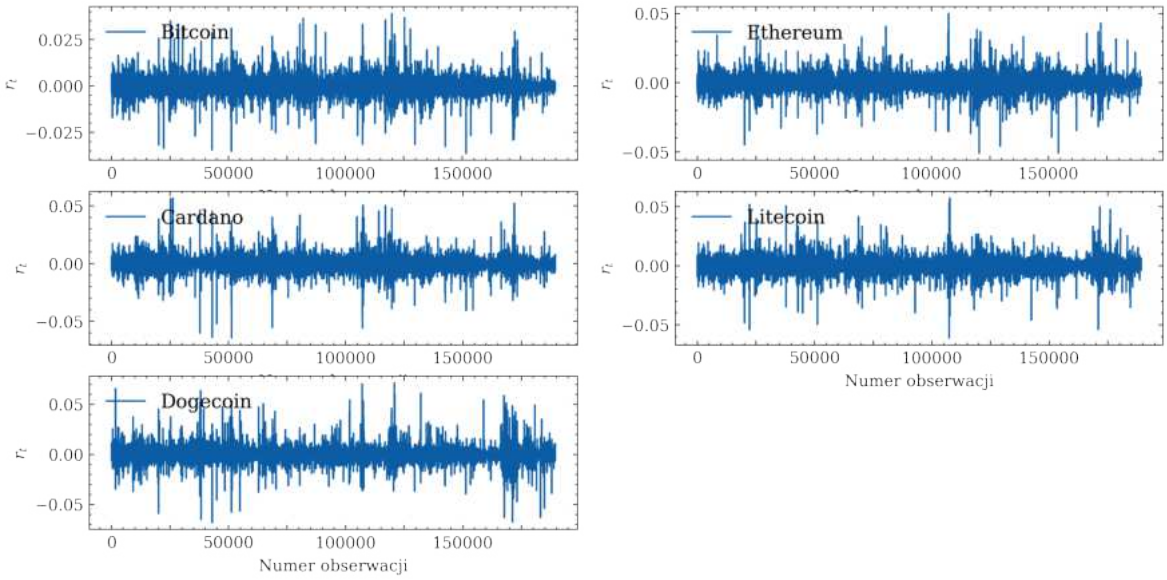
podobnie - jako instrument finansowy, w który można zainwestować po pojawieniu się korzystnych dla niego wiadomości.

Charakterystyka wiadomości opisujących kryptowaluty jest inna niż w przypadku spółek. Artykuły te często analizują zmiany cenowe w krótkim i długim okresie oraz starają się przewidywać przyszłe tendencje na podstawie różnych czynników, takich jak nowe regulacje, wydarzenia na rynku, zmiany technologiczne czy ogólna sytuacja gospodarcza. Mogą to być również doniesienia o przyjęciu kryptowalut przez różne instytucje finansowe czy firmy, co może mieć wpływ na ich wartość rynkową. Pojawiają się również wiadomości na temat zmian w regulacjach dotyczących kryptowalut, zarówno na poziomie krajowym, jak i międzynarodowym. Informacje o próbach regulacji, zakazach czy nowych przepisach prawnych mogą znacząco wpływać na rynek kryptowalut.

Tabela 4.3 przedstawia ilość wiadomości pobranych z serwisu CiyFALCON dla każdej z wybranych kryptowalut w badanym okresie. Handel kryptowalutami odbywa się również w weekendy i święta, więc dla tego samego zakresu dat daje to 661 dni z otwartą sesją. Można zaobserwować, że średnia dzienna ilość wiadomości na temat Bitcoina jest 3 razy większa niż dla najbardziej medialnej z badanych spółek, Tesla Inc. (TSLA).

Szeregi czasowe stóp zwrotów w 5-minutowych interwałach dla badanych kryptowalut zostały przedstawione na rysunku 4.18. Nie ma istotnych różnic w zmienności w prezentowanym okresie. Tak jak w przypadku akcji, przez większość czasu zmienność ma względnie stałą wielkość, a okazjonalnie pojawiają się obserwacje odstające.

W tabeli 4.4 znajdują się statystyki opisowe kryptowalut wybranych do badania. Wartość średnia logarytmicznej stopy zwrotu dla wszystkich z nich jest niemal równa zero. Odchylenie standardowe odpowiada wielkością bardziej zmiennym spółkom opisa-



Rysunek 4.18. Szeregi czasowe logarytmicznych stóp zwrotu dla kryptowalut uwzględnionych w badaniach.
Źródło: Opracowanie własne.

Tabela 4.4. Statystyki opisowe szeregów zwrotów 5-minutowych z kursów kryptowalut.

Instrument	Średnia	Odch. stand.	Min.	Maks.	Skośność	Kurtoza	Liczba obs.
Bitcoin	-4.12e-06	0.00207	-0.0363	0.0391	0.224	25.0	189792
Ethereum	-3.68e-06	0.00268	-0.0510	0.0501	0.047	21.1	189614
Cardano	-7.46e-06	0.00316	-0.0643	0.0567	0.223	21.2	189504
Litecoin	-4.13e-06	0.00303	-0.0613	0.0572	0.035	21.5	189225
Dogecoin	-5.91e-06	0.00363	-0.0682	0.0716	0.443	29.4	189504

Źródło: Opracowanie własne.

nym powyżej (TSLA, NVDA, F). Obserwacje minimalne i maksymalne co do wartości bezwzględnej również posiadają podobne zakresy.

Skośność dla wszystkich kryptowalut jest dodatnia i mniejsza niż 0,5. Kurtoza jest znacznie mniejsza niż w przypadku akcji - oznacza to, że dla wybranej częstotliwości obserwacje znacznie odstające są mniej prawdopodobne niż w przypadku akcji.

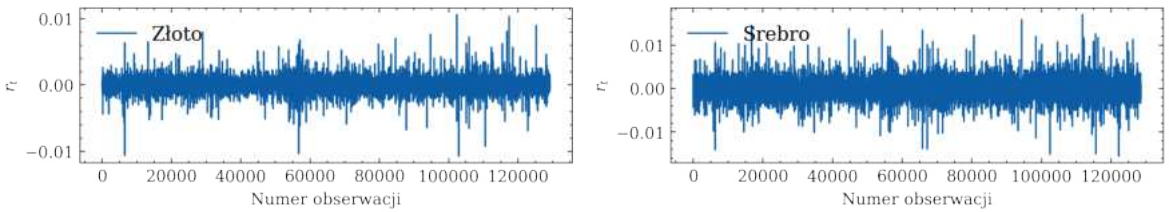
Liczba obserwacji dla kryptowalut, dla tego samego okresu jest znacznie większa, ponieważ handel odbywa się w sposób ciągły tzn. we wszystkie dni tygodnia i 24 godziny na dobę. Wartości nie są dokładnie jednakowe ze względu na sporadycznie brakujące obserwacje w danych pobranych z serwisu Alpha Vantage.

Zbadane zostały również stopy zwrotu dla dwóch metali szlachetnych - złota i srebra. Są to fizyczne aktywa, które mają duże znaczenie w światowej gospodarce.

Tabela 4.5. Całkowita i średnia dzienna ilość zgromadzonych wiadomości na temat badanych metali szlachetnych.

Metal	Ilość wiadomości	
	całkowita	średnia dzienna
Złoto	10189	20.26
Srebro	1916	3.81

Źródło: Opracowanie własne.



Rysunek 4.19. Szeregi czasowe logarytmicznych stóp zwrotu dla metali szlachetnych uwzględnionych w badaniach.
Źródło: Opracowanie własne.

Inwestorzy również mogą wykorzystywać je jako instrumenty finansowe, jednak mają też zastosowania przemysłowe. Poza tym, muszą one być wydobywane, a nie tworzone sztucznie jak kryptowaluty. Wiadomości, które pojawiają się w mediach na temat złota i srebra są często powiązane z gospodarką monetarną (i pokryciem walut w złocie przez banki centralne). Indywidualni inwestorzy często wykorzystują je jako "bezpieczną przystań" dla kapitału i zabezpieczenie na bardziej niespokojne czasy.

Statystyki dotyczące ilości wiadomości finansowych pojawiających się w mediach na temat wybranych metali szlachetnych przedstawia tabela 4.5. Średnie dzienne ilości wiadomości są wysokie i porównywalne z wartościami dla bardziej popularnych spółek i kryptowalut. Tak jak w przypadku akcji spółek giełdowych, handel metalami szlachetnymi nie odbywa się w weekendy i święta.

Na rysunku 4.19 zostały przedstawione przebiegi stóp zwrotów w 5-minutowych interwałach dla badanych metali szlachetnych. W wybranym zakresie dat zmienność jest dość jednorodna ze sporadycznymi obserwacjami odstającymi. Trudno jest wskazać wyróżniające się okresy.

Statystyki opisowe dla złota i srebra znajdują się w tabeli 4.6. Wartość średnia logarytmicznej stopy zwrotu dla obu jest niemal równa zero. Odchylenie standardowe jest niższe niż dla którychkolwiek z badanych akcji lub kryptowalut. Również moduły wartości minimalnych i maksymalnych są niskie.

Skośność rozkładów logarytmicznej stopy zwrotu dla obu metali jest niska co do

Tabela 4.6. Statystyki opisowe szeregów zwrotów 5-minutowych z cen metali szlachetnych.

Metal	Średnia	Odch. stand.	Min.	Maks.	Skośność	Kurtoza	Liczba próbek
Złoto	-2.45e-07	0.00055	-0.0107	0.0106	-0.108	24.9	129190
Srebro	-6.11e-07	0.00113	-0.0155	0.0170	0.129	11.3	128585

Źródło: Opracowanie własne.

wartości bezwzględnej (rozkłady są niemal symetryczne). Kurtoza złota ma wartość podobną do kurtozy dla kryptowalut, zaś kurtoza srebra jest znacznie niższa.

Handel metalami szlachetnymi odbywa się na wielu rynkach, co w efekcie powoduje, że ceny są aktualizowane przez cała dobę we wszystkie dni poza weekendami i świętami. Liczba próbek jest zatem większa niż dla akcji z tego samego zakresu dat. Różnice w dokładnych wartościach wynikają z brakujących obserwacji w serwisie Alpha Vantage.

4.5.2. Schemat prognozowania i walidacja modeli

Walidacja jest koniecznym krokiem w celu upewnienia się, że otrzymane modele potrafią dobrze *generalizować* - tj. czy dają dobre prognozy dla danych, które nie były wykorzystywane w procesie estymacji (trenowania) modelu. Poprawna walidacja wymaga odpowiedniego podziału dostępnych danych na część wykorzystaną do estymacji modelu i część do jego walidacji (oraz ewentualnie część testową). Konkretne miary jakości prognoz dla tak zdefiniowanych zbiorów zostały opisane dokładniej w rozdziale 4.5.3. Walidacja pozwala również upewnić się, że hiperparametry modelu (parametry, które nie podlegają procesowi treningu, lecz są narzucone przez badacza) zostały poprawnie wybrane.

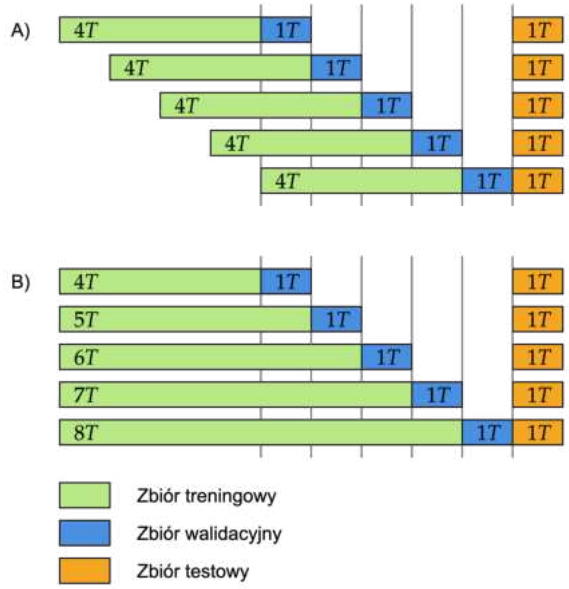
Walidacja jest stosowana, aby upewnić się, że model jest stabilny, tj. będzie dawał poprawne prognozy również dla danych, które nie były użyte do wytrenowania go. Podejście to polega na wykonaniu kilku podziałów danych na 3 opisane wcześniej podzbiory i przeprowadzeniu procedury treningu oraz walidacji dla każdego z nich. Zbiór testowy dla każdego z podziałów zawsze pozostaje ten sam i obejmuje najnowsze obserwacje.

W przypadku gdy hiperparametry modelu są już ustalone, miara trafności P dla prognoz na każdym ze zbiorów walidacyjnych powinna mieć możliwie niską wariancję. W przeciwnym przypadku istnieje możliwość, że model nie daje porównywalnych wyników w zależności od okresu, na którym został wytrenowany i nie będzie można skutecznie stosować go do prognozowania przyszłości.

Zgodnie z przyjmowaną w literaturze przedmiotu z zakresu prognozowania konwencją dotyczącą sposobu oceny zdolności prognostycznych rozważanych modeli, analiza może być przeprowadzona dla dwóch typów schematu prognozowania (schematycznie przedstawionych na rysunku 4.20): rolowanego schematu prognozowania (ang. *rolling scheme*) oraz rekursywnego schematu prognozowania (ang. *recursive scheme*). Różnica wynika ze stosowanego typu okna czasowego wykorzystywanego do estymacji modeli: stałego lub zmiennego.

Schemat rolowany (ze stałym oknem estymacji) charakteryzuje się zbiorem treningowym i walidacyjnym o stałej długość, zaś czas ich rozpoczęcia i końca jest ustalany ze stałym krokiem (rysunek 4.20A). Przykładowo dla spółki TSLA cały zbiór danych składa się z $T = 35576$ obserwacji, można nadać im numerację $t = 1, ..., T$, czyli $t = 1, ..., 35576$. Zakładając, że w badaniu empirycznym długość podokresu wykorzystanego do estymacji modelu jest ustalona i została oznaczona jako $s = 4914$, zbiór treningowy składa się z obserwacji $t = 1, ..., s$, czyli $t = 1, ..., 4914$. Niech x_t oznacza obserwację w chwili t. Wówczas **w pierwszym kroku** analizy zbiór treningowy składa się z obserwacji $x_1, ..., x_s$ (a zatem $x_1, ..., x_{4914}$), a prognoza na kolejny okres (o horyzoncie prognozy $h = 1$) z modelu estymowanego na tym zbiorze treningowym, to $\hat{x}_{s+1}$, czyli $\hat{x}_{4915}$. W **kolejnym kroku** wszystkie indeksy przesuwają się o 1. Zestaw treningowy to $x_2, ..., x_{4915}$, a prognoza zostaje wyliczona dla następnego okresu: $\hat{x}_{s+2}$. W następnych krokach zbiór treningowy przesuwa się, pozostając jednakowej długości s, pozwalając ostatecznie wygenerować szereg wszystkich prognoz: $\hat{x}_{T-s}, ..., \hat{x}_T$, tj. $\hat{x}_{30662}, ..., \hat{x}_{35576}$.

W **schemacie rekursywnym** (ze zmiennym oknem estymacji) zbiór treningowy zaczyna się zawsze od najstarszej dostępnej próbki i jest wydłużany tak, że koniec przypada w różnych chwilach, zaś zbiór walidacyjny ma stałą długość i zaczyna się w chwili zakończenia zbioru treningowego (rysunek 4.20B). Przyjmując takie same oznaczania, jak w przypadku schematu rolowanego, w **pierwszym kroku** zbiór treningowy i prognoza są takie same - odpowiednio $x_1, ..., x_{4914}$ oraz $\hat{x}_{s+1}$. W **kolejnym kroku** przesunięty zostaje jedynie indeks prawego końca zbioru uczącego. W takim przypadku do estymacji użyte są wszystkie próbki poczynając od najstarszej: $x_1, ..., x_{s+1}$. Otrzymana prognoza to $\hat{x}_{s+2}$. Ostatecznie wszystkie prognozy zostają wyliczony dla tego samego okresu, co w przypadku schematu rolowanego $\hat{x}_{s+1}, ..., \hat{x}_T$, (czyli $\hat{x}_{4915}, ..., \hat{x}_{35576}$), jednak każda kolejna z nich powstaje na podstawie modelu estymowanego na coraz dłuższym zakresie.



Rysunek 4.20. Dwa sposoby schematu prognozowania (podział danych do wykorzystania w celu walidacji). A) Schemat rolowany (ze stałym oknem czasowym) wykorzystuje szeregi o stałej długości do estymacji modelu. B) W schemacie rekursywnym okno czasowe obejmujące dane treningowe wydłuża się. Zbiór walidacyjny i testowy w obu schematach są stałej długości i są tak samo umiejscowione.
Źródło: Opracowanie własne.

Miary trafności prognoz modeli są wyliczane dla tak otrzymanych okresów walidacyjnych. W niniejszej pracy przyjęty został **schemat rolowany**.

4.5.3. Metody oceny trafności prognoz

Do porównywania trafności prognoz ex post w w literaturze przedmiotu z zakresu prognozowania szeregów czasowych zaproponowano bardzo dużo mierników. Szczegółowy opis tych miar znajduje się w wielu pracach m.in. w: Makridakis S. i in. (1982 [123], 1998 [196]), Hyndman R. i Koehler A. (2006) [94], Zeliaś A. i in. (2008) [202], Welfe W. (2009) [194], Doman M. i Doman R. (2009) [45], Weron R. (2014) [195].

W celu zdefiniowania najczęściej wykorzystywanych w badaniach empirycznych mierników oceny trafności prognoz ex post wprowadzone zostały następujące oznaczenia. Przyjęty w badaniu schemat prognozowania pozwala zbudować szereg czasowy, którego wartości są prognozami uzyskanymi na podstawie kolejnych oszacowanych modeli. Niech N będzie długością otrzymanego szeregu prognoz punktowych (przykładowo stóp zwrotu lub cen). W przypadku prognozowania procesu stochastycznego Y_t niech y_t oznacza wartość rzeczywistą zmiennej prognozowanej Y w chwili t, a $\hat{y}_t$ - prognozę

punktową zmiennej Y na chwilę t. Podstawową wielkością konieczną do zdefiniowania miar trafności prognoz jest (bezwzględny) błąd prognozy ex post opisany równaniem:

$$e_t = y_t - \hat{y}_t, \qquad t = 1, ..., N. \tag{4.9}$$

Bazując na powyższym równaniu, można również zdefiniować procentowy (względny) błąd prognozy ex post:

$$\delta_t = \frac{e_t}{y_t} \cdot 100\%, \qquad t = 1, ..., N. \tag{4.10}$$

Ze względu na sposób obliczania błędów prognoz (bezwzględnych lub względnych) istnieje wiele dostępnych miar. Podstawową i najprostszą miarą oceny trafności prognoz jest średni błąd prognozy - ang. *Mean Error* (ME). Jest to zwykła średnia z kolejnych błędów prognozy:

$$ME = \frac{\sum_{t=1}^{N} e_t}{N}, \tag{4.11}$$

gdzie N jest długością szeregu wyznaczonych prognoz.

Gdy wartość miary ME jest ujemna, oznacza to, że prognozy są niedoszacowane, w przeciwnym przypadku zaś przeszacowane. Wartości tej miary wykorzystują tę samą skalę oraz jednostkę, co wielkość modelowana. Oznacza to, że miara ME nie może być wykorzystywana do porównania ze sobą dwóch modeli utworzonych dla danych o różnych skalach.

Uśrednianie błędów razem z ich znakiem może prowadzić do otrzymania bardzo niskich wartości, zwłaszcza kiedy estymator (model) jest nieobciążony. Z tego powodu wprowadza się średni błąd bezwzględny - ang. *Mean Absolute Error* (MAE):

$$MAE = \frac{\sum_{t=1}^{N} |e_t|}{N}. \tag{4.12}$$

Miara MAE informuje, o ile średnio rzeczywiste realizacje zmiennej prognozowanej mogą odchylać się od prognoz co do ich bezwzględnej wartości. Miara ta sprawdza się szczególnie dobrze, gdy błędy są symetrycznie rozłożone, a w danych nie ma wyjątkowo mocno odstających wartości.

Błąd średniokwadratowy - ang. *Mean Squared Error* (MSE) wykorzystywany jest w celu przypisania większej wagi dużym odchyleniom. Znaczną wadą MSE jest jednak bardzo silny wpływ pojedynczych obserwacji odstających na wartość całej miary. W

niektórych zastosowaniach używa się pierwiastka kwadratowego MSE otrzymując miarę RMSE. Miary te są dane przez wzory 4.13 oraz 4.14:

$$MSE = \frac{\sum_{t=1}^{N} e_t^2}{N}, \tag{4.13}$$

$$RMSE = \sqrt{MSE}. \tag{4.14}$$

Jeżeli badany model jest estymatorem nieobciążonym, wówczas MSE może być interpretowany jako wariancja błędu, a RMSE jako jego odchylenie standardowe. Duża różnica wartości miar MAE oraz RMSE obliczonych dla tego samego okresu może wskazywać na występowanie w nim błędów o bardzo dużych wartościach. Zerowa wartość miar MSE i RMSE oznaczałaby idealny model niepopełniający żadnych błędów.

Kiedy wartość błędu jest trudna do zinterpretowania (na przykład ze względu na rząd wielkości lub brak zrozumiałej fizycznej interpretacji), można wykorzystać średni błąd procentowy- ang. *Mean Percentage Error* (MPE) dany wzorem 4.15 lub średni bezwzględny błąd procentowy- ang. *Mean Absolute Percentage Error* (MAPE) dany wzorem 4.16:

$$MPE = \frac{\sum_{t=1}^{N} \delta_t}{N}, \tag{4.15}$$

$$MAPE = \frac{\sum_{t=1}^{N} |\delta_t|}{N}. \tag{4.16}$$

Miara MAPE informuje o średniej wielkości błędów prognoz wyrażonych w procentach rzeczywistej wartości prognozowanej zmiennej.

Mimo wygody stosowania, miary MPE oraz MAPE mają również wady. W przypadku, gdy modelowana wielkość jest bliska zera (np. logarytmiczne stopy zwrotu), wartości otrzymywanych metryk mogą dążyć do nieskończoności. W przypadku zaś, gdy którakolwiek z obserwacji przyjmie wartość 0, miary te są nieokreślone.

Każda z wymienionych wyżej miar ma swoje wady i zalety - mogą sprawdzać się różnie w zależności od sytuacji i typu modelu. Dlatego ważne jest, aby badając nowe metody prognozowania nie skupiać się wyłącznie na jednej z nich, lecz przeprowadzić analizy wykorzystując przynajmniej kilka z nich.

Dodatkowo konieczne jest zdefiniowanie miernika, który pozwoli na porównanie ze

sobą wyników dwóch różnych modeli. Takim miernikiem jest np. R^2_{OS} zaproponowany przez Campbella i Thompsona w 2008 roku [19]:

$$R^2_{OS} = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}, \tag{4.17}$$

gdzie $\bar{y}_i$ jest historyczną średnią obliczoną w okresie $i-1$.

Równolegle z obliczaniem mierników oceny trafności prognoz w literaturze przedmiotu zwiazanej z prognozowaniem porównuje się trafność prognoz za pomocą testów statystycznych. Najbardziej znanym podejściem do testowania różnic między trafnością prognoz wygenerowanych różnymi metodami prognozowania (różnymi typami modeli) jest test Diebolda-Mariano opisany w 1995 roku [40]. Test ten przeprowadzany jest dla dwóch szeregów czasowych błędów prognoz (zdefiniowanych jak we wzorze 4.9) e_{1t} oraz e_{2t} otrzymanych z modeli 1 i 2. Na początku definiuje się różnicę funkcji strat dla obu prognoz:

$$d_t = g(e_{1t}) - g(e_{2t}), \tag{4.18}$$

gdzie g jest funkcją straty. Najczęściej wykorzystywany jest kwadrat błędu prognozy $g(e_{it}) = e^2_{it}$, lecz można użyć np. asymetrycznych funkcji strat.

Przy tak zdefiniowanej wartości d_t, zakłada się, że dwie prognozy mają jednakową dokładność wtedy i tylko wtedy, gdy wartość oczekiwana dla wszystkich t jest równa zero. Wówczas hipotezę H_0 testu DM można zdefiniować następująco:

$$H_0 : \forall_t \ \mathbb{E}(d_t) = 0. \tag{4.19}$$

Jedną z najistotniejszych cech tego testu jest brak konieczności porównywania modeli tej samej klasy ani nawet znajomości postaci badanych modeli. Dzięki temu możliwe jest porównanie klasycznych modeli ekonometrycznych z sieciami neuronowymi, co wykorzystane jest w niniejszej pracy.

W literaturze można także znaleźć skorygowane i uogólnione wersje testu DM. W publikacji z 1997r. Harvey i in. zaproponowano wersję testu Diebolda-Mariano dla małej próby [81]. W 2007r. Clark i West zaproponowali test do porównywania prognoz otrzymanych z dwóch modeli, z których jeden jest zagnieżdżony w drugim [28].

Oprócz testów porównujących modele prognostyczne parami, istnieją także testy porównujące wiele modeli łącznie. White H. zaproponował test Reality Check [197], w który weryfikowana jest hipoteza zerowa głosząca, że badany model jest lepszy od m alternatywnych modeli:

$$H_0 : \max_{k=1,\ldots,m} \mathbb{E}[d_{k,t+1|t}(\beta_k^*)] \leqslant 0, \tag{4.20}$$

zaś hipoteza alternatywna głosi, że przynajmniej jeden z modeli jest lepszy:

$$H_0 : \max_{k=1,\ldots,m} \mathbb{E}[d_{k,t+1|t}(\beta_k^*)] > 0, \tag{4.21}$$

gdzie $d_{k,t+1|t}(\beta_k^*)$ jest różnicą funkcji strat między badanym modelem i m-tą alternatywą dla tego modelu.

Kolejnym podejściem jest test SPA (Superior Predictive Ability test) [78] oraz procedura MCS (Model Confidence Set procedure) [77] - oba opracowane przez Hansena P. w 2005 i 2011 roku w celu poprawienia wad testu Reality Check poprzez normalizację i centrowanie jego statystyki.

4.6. Podsumowanie

Metodologia proponowanych badań składa się głównie z obliczeń i symulacji komputerowych prowadzących do stworzenia skutecznego modelu prognostycznego. Wykorzystane metody weryfikacji hipotez badawczych polegają głównie na przeprowadzaniu testów statystycznych na wynikach otrzymanych modeli, a także porównaniu mierników trafności prognoz.

Modele wykorzystujące informacje z rynków finansowych muszą być w odpowiedni sposób estymowane (trenowane w przypadku sieci neuronowych). Zwykle w procesie tym wykorzystuje się część dostępnych historycznych danych, a następnie otrzymanym modelem oblicza się prognozę, którą można zweryfikować na pozostałej części zbioru danych. W celu walidacji modelu wykorzystuje się wybrane miary bazujące na błędach predykcji, co pozwala ostatecznie wybrać model o najlepszym dopasowaniu.

Efektem prowadzonych prac jest autorski system do prognozowania notowań giełdowych, w którego skład wchodzą moduły pobierające, czyszczące i wstępnie przetwarzające dane uzyskane od dostawców, a także moduły pozwalające w rzetelny i powtarzalny sposób estymować klasyczne modele ekonometryczne, modele oparte o sieci neuronowe (LSTM-FENE) oraz modele hybrydowe (ARMA-GARCH-FENE). W modelach zawierających części sieci neuronowych wykorzystane zostało nowatorskie podejście do tworzenia prognoz oparte o techniki NLP.

Rozdział 5

Badania efektywności predykcji finansowych szeregów czasowych wspomaganej narzędziami NLP

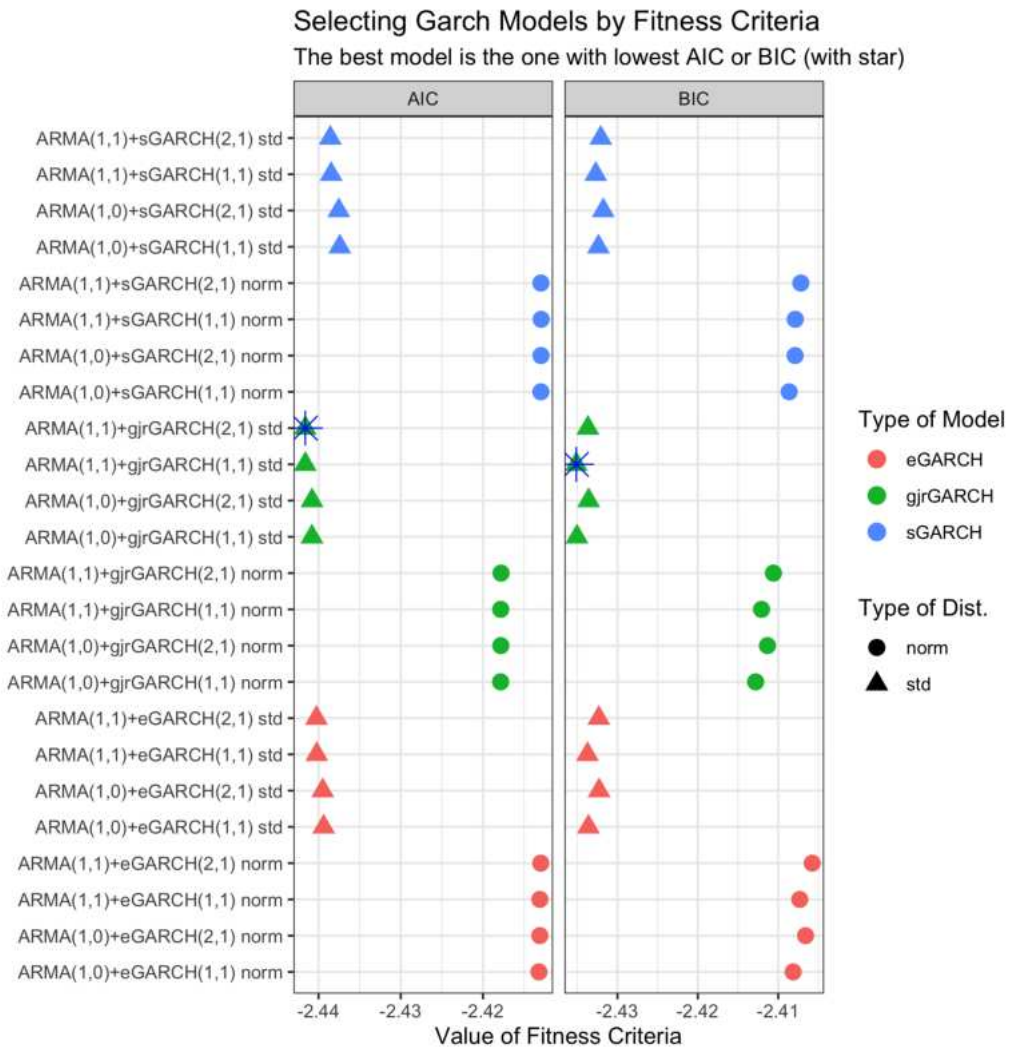
Efektywność modeli prognostycznych zależy od szeregu czynników, takich jak: rodzaj wiadomości (źródło ich pochodzenia), ilość wiadomości dotyczących badanej spółki w jednostce czasu (ilość dodatkowych informacji dostępnych dla inwestorów), a także hiperparametrów opisujących model (jego architekturę, proces trenowania i wybrane zmienne objaśniające).

Niniejszy rozdział przedstawia wyniki prognozowania finansowych szeregów czasowych, wspomaganego narzędziami automatycznej analizy tekstów. Jako punkt odniesienia przyjęto prognozy otrzymane przy pomocy klasycznych modeli ekonometrycznych, dających możliwie najlepsze wyniki. Modele te nazywane będą **modelami referencyjnymi**.

5.1. Wybór modeli referencyjnych

Dla każdego aktywa model referencyjny dobierany jest indywidualnie spośród trzech specyfikacji: ARMA-GARCH, ARMA-GJR-GARCH oraz ARMA-EGARCH. Dla każdej z tych specyfikacji badane są dwa rozkłady warunkowe: rozkład normalny oraz rozkład t-Studenta. Dla tak otrzymanych kombinacji wybierany jest optymalny rząd opóźnień.

Wybór specyfikacji, rozkładu warunkowego oraz rzędu modelu podyktowany był kryteriami AIC oraz BIC (patrz rozdział 2.1.1). Do automatycznego estymowania tych modeli oraz wyboru specyfikacji GARCH posłużono się repozytorium udostępnionym wraz z publikacją Perlina M. i in. [150]. Przykładowe podsumowanie analizy rzędów opóźnień oraz specyfikacji modeli dla spółki Tesla Inc. (TSLA) przedstawia Rysunek 5.1. Im niższa wartość kryteriów AIC oraz BIC tym lepsze dopasowanie modelu do danych. Dla każdego z kryteriów najlepszy model został oznaczony gwiazdką. Szcze-



Rysunek 5.1. Wybór najlepszej specyfikacji modelu ARMA-GARCH na podstawie kryteriów AIC oraz BIC dla spółki Tesla Inc. (TSLA). Fioletowa gwiazdka oznacza, który typ modelu osiąga najniższą wartość każdego z kryteriów. Różne specyfikacje modelu GARCH oznaczone są różnymi kolorami, zaś różne rozkłady kształtami znaczników.
Źródło: Obliczenia własne, wizualizacja na podstawie [150].

gółowe wyniki takiej analizy dla każdego z badanych instrumentów zostały zebrane w aneksie (rozdział A.2).

Podsumowanie rzędów opóźnień, specyfikacji i rozkładów prawdopodobieństwa otrzymanych na podstawie kryterium informacyjnego AIC dla każdego instrumentu znajduje się w tabeli 5.1. Analogiczne wyniki dla kryterium BIC przedstawia tabela 5.2. Różnice w wartości obu kryteriów były niewielkie między otrzymywanymi modelami. Do dalszych analiz modele były wybierane na podstawie kryterium AIC.

W przypadku 5-minutowych interwałów najlepsze wyniki były osiągane przez modele ARMA-GARCH estymowane w schemacie rekursywnym z oknem o długości 63 dni (ilość sesji giełdowych w ciągu 3 miesiący) po 78 obserwacji dziennie, tj. 4914 obserwacji

Tabela 5.1. Optymalne rzędy opóźnień, specyfikacje i warunkowe rozkłady prawdopodobieństwa dla modeli ARMA-GARCH dla badanych instrumentów finansowych na podstawie kryterium informacyjnego AIC.

	ARMA		GARCH			
Instrument	p	q	q	p	specyfikacja	rozkład
TSLA	1	1	2	1	**GJR-GARCH**	t-Studenta
NKE	1	0	2	1	EGARCH	t-Studenta
KO	1	1	2	1	EGARCH	t-Studenta
BA	1	0	2	1	EGARCH	t-Studenta
CRM	1	1	1	1	EGARCH	t-Studenta
GM	1	0	2	1	EGARCH	t-Studenta
JPM	1	0	2	1	EGARCH	t-Studenta
MSFT	1	1	1	1	EGARCH	t-Studenta
GS	1	0	1	1	EGARCH	t-Studenta
NVDA	1	1	2	1	**GJR-GARCH**	t-Studenta
F	1	0	1	1	EGARCH	t-Studenta
C	1	1	2	1	EGARCH	t-Studenta
MS	1	0	2	1	EGARCH	t-Studenta
INTC	1	0	2	1	EGARCH	t-Studenta
VZ	1	1	2	1	EGARCH	t-Studenta
Złoto	1	0	2	1	EGARCH	t-Studenta
Srebro	1	1	2	1	EGARCH	t-Studenta
Bitcoin	1	1	2	1	EGARCH	t-Studenta
Cardano	1	1	2	1	EGARCH	t-Studenta
Ethereum	1	1	2	1	EGARCH	t-Studenta
Litecoin	1	1	2	1	EGARCH	t-Studenta
Dogecoin	1	1	2	1	EGARCH	t-Studenta

Źródło: Obliczenia własne.

Tabela 5.2. Optymalne rzędy opóźnień, specyfikacje i warunkowe rozkłady prawdopodobieństwa dla modeli ARMA-GARCH dla badanych instrumentów finansowych na podstawie kryterium informacyjnego BIC.

	ARMA		GARCH			
Instrument	p	q	q	p	specyfikacja	rozkład
TSLA	1	1	1	1	**GJR-GARCH**	t-Studenta
NKE	1	0	2	1	EGARCH	t-Studenta
KO	1	0	1	1	EGARCH	t-Studenta
BA	1	0	1	1	**GARCH**	t-Studenta
CRM	1	1	1	1	EGARCH	t-Studenta
GM	1	0	1	1	EGARCH	t-Studenta
JPM	1	0	1	1	EGARCH	t-Studenta
MSFT	1	0	1	1	EGARCH	t-Studenta
GS	1	0	1	1	EGARCH	t-Studenta
NVDA	1	0	1	1	**GJR-GARCH**	t-Studenta
F	1	0	1	1	EGARCH	t-Studenta
C	1	0	1	1	EGARCH	t-Studenta
MS	1	0	1	1	EGARCH	t-Studenta
INTC	1	0	2	1	EGARCH	t-Studenta
VZ	1	1	1	1	EGARCH	t-Studenta
Złoto	1	0	2	1	EGARCH	t-Studenta
Srebro	1	1	2	1	EGARCH	t-Studenta
Bitcoin	1	1	2	1	EGARCH	t-Studenta
Cardano	1	0	1	1	EGARCH	t-Studenta
Ethereum	1	0	2	1	EGARCH	t-Studenta
Litecoin	1	1	2	1	EGARCH	t-Studenta
Dogecoin	1	1	1	1	EGARCH	t-Studenta

Źródło: Obliczenia własne.

dla spółek giełdowych. Dla spójności ta sama ilość próbek została przyjęta również w przypadku kryptowalut i metali szlachetnych, mimo iż jest to inna długość przedziału czasowego, dla którego estymowany jest model (ze względu na inne godziny i dni tygodnia, w których prowadzony jest handel tymi instrumentami). Po ustaleniu najlepszej specyfikacji modelu na pierwszym oknie czasowym, była ona następnie utrzymana we wszystkich kolejnych oknach czasowych dla danego instrumentu finansowego.

Każdy z modeli wykorzystywał egzogeniczny szereg czasowy w postaci wzorca sezonowości (opisany w rozdziale 4.2.1) w celu uwzględnienia napływu dodatkowych informacji. Tak otrzymane modele zostały wykorzystane do obliczenia prognoz o horyzoncie $h = 1$ (krok = 5 minut), które następnie posłużyły do porównania ich z prognozami otrzymanymi przy użyciu autorskich modeli LSTM-FENE.

5.2. Analiza efektywności autorskich modeli

Skuteczność modeli w niniejszej pracy badana jest między innymi poprzez porównanie błędów prognoz z błędami prognoz bazowych modeli. Obliczana jest procentowa zmiana błędu prognozy dla badanego modelu względem błędu prognozy modelu bazowego. W ten sposób otrzymywane są mierniki:

$$\Delta RMSE = \frac{RMSE_{badany\ model} - RMSE_{bazowy\ model}}{RMSE_{bazowy\ model}} \cdot 100\%, \tag{5.1}$$

$$\Delta MAE = \frac{MAE_{badany\ model} - MAE_{bazowy\ model}}{MAE_{bazowy\ model}} \cdot 100\%, \tag{5.2}$$

$$\Delta MAPE = \frac{MAPE_{badany\ model} - MAPE_{bazowy\ model}}{MAPE_{bazowy\ model}} \cdot 100\%. \tag{5.3}$$

Mierniki zdefiniowane w ten sposób mają jednorodną interpretację - opisują procentową zmianę błędu prognozy poprzez wykorzystanie badanego modelu. Ujemna wartość oznacza, że w tym przypadku badany model jest lepszy od bazowego.

5.2.1. Ogólne porównanie modeli LSTM-FENE z modelami bazowymi

Tabela 5.3 zawiera porównanie błędów prognoz stóp zwrotu RMSE oraz MAE dla modeli LSTM-FENE (patrz rozdział 4.3) z błędami prognoz dla modeli bazowych ARMA-GARCH (patrz rozdział 2.1) dla badanych spółek indeksu S&P 500. Wszystkie błędy prognoz zostały obliczone dla horyzontu $h = 1$ z uwzględnieniem wszystkich

próbek - zarówno zawierających, jak i niezawierających nowych wiadomości na temat danego instrumentu. Dzięki temu możliwe jest ogólne porównanie skuteczności obu podejść do modelowania.

Tabela 5.3. Mierniki RMSE oraz MAE oceny trafności prognoz stóp zwrotu dla badanych spółek indeksu S&P 500.

	LSTM-FENE		ARMA-GARCH	
Spółka	RMSE	MAE	RMSE	MAE
TSLA	0.009666	0.002564	0.012960	0.002658
NKE	0.002814	0.001303	0.003230	0.001481
KO	0.001187	0.000770	0.001326	0.000859
BA	0.002568	0.001539	0.002859	0.001763
CRM	0.002993	0.001531	0.003247	0.001731
GM	0.002748	0.001598	0.002909	0.001693
JPM	0.001935	0.001139	0.002045	0.001234
MSFT	0.002215	0.001220	0.002332	0.001300
GS	0.001962	0.001170	0.002007	0.001212
NVDA	0.003925	0.002165	0.003979	0.002392
F	0.003041	0.001678	0.003079	0.001720
C	0.002166	0.001299	0.002132	0.001329
MS	0.002241	0.001234	0.002137	0.001283
INTC	0.002772	0.001584	0.002640	0.001563
VZ	0.001792	0.000973	0.001664	0.001006

Źródło: Obliczenia własne.

Wybrane do analizy spółki charakteryzują się znaczną płynnością, a także dużą obecnością w mediach. W serwisach informacyjnych regularnie pojawiają się wiadomości na ich temat, dzięki czemu model LSTM-FENE może osiągnąć przewagę poprzez wykorzystanie dodatkowych informacji zawartych w treści wiadomości.

W tabeli 5.4 znajdują się procentowe zmiany poszczególnych błędów prognozy stóp zwrotu dla każdej z badanych spółek. Ujemna wartość oznacza spadek wielkości błędu prognozy uzyskanej przy wykorzystaniu autorskiego modelu LSTM-FENE względem wielkości błędu uzyskanego przy pomocy modeli typu ARMA-GARCH. Dodatkowo zaznaczona została dzienna średnia liczba wiadomości w okresie pokrywanym przez zbiór walidacyjny. Model LSTM-FENE osiągnął lepsze wyniki dla 11 z 15 badanych spółek. Wyłącznie wyniki dla spółek C, MS, INTC oraz VZ okazały się być gorsze niż w przypadku modelu bazowego.

Na podstawie prognoz stóp zwrotu możliwe jest obliczenie prognoz poziomów cen. Dla takich prognoz w analogiczny sposób zostały obliczone błędy i przedstawione w tabeli 5.5.

Tabela 5.4. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych spółek indeksuS&P 500 - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Spółka	ΔRMSE [%]	ΔMAE [%]	Dzienna liczba wiadomości
TSLA	-25.42	-3.54	5.42
NKE	-12.85	-11.99	0.98
KO	-10.46	-10.33	0.40
BA	-10.16	-12.68	1.16
CRM	-7.83	-11.55	0.67
GM	-5.55	-5.62	1.47
JPM	-5.40	-7.65	1.04
MSFT	-5.01	-6.21	2.45
GS	-2.27	-3.42	0.46
NVDA	-1.35	-9.51	1.70
F	-1.25	-2.44	0.27
C	1.58	-2.23	0.56
MS	4.91	-3.84	0.25
INTC	5.02	1.37	0.77
VZ	7.69	-3.27	0.40

Źródło: Obliczenia własne.

Tabela 5.5. Mierniki RMSE, MAE oraz MAPE oceny trafności prognoz cen instrumentów finansowych dla badanych spółek indeksu S&P 500.

	LSTM-FENE			ARMA-GARCH		
Spółka	RMSE	MAE	MAPE	RMSE	MAE	MAPE
TSLA	0.810801	0.482204	0.002546	0.919301	0.548233	0.002738
KO	0.071589	0.046818	0.000770	0.079398	0.051962	0.000859
NKE	0.303166	0.143985	0.001304	0.325516	0.152479	0.001481
GM	0.100544	0.058660	0.001598	0.107383	0.062981	0.001693
F	0.039632	0.021718	0.001679	0.041565	0.023272	0.001721
GS	0.656444	0.395351	0.001170	0.670350	0.408286	0.001211
MSFT	0.566544	0.315172	0.001220	0.571391	0.321687	0.001301
CRM	0.499337	0.252904	0.001531	0.503360	0.265500	0.001731
JPM	0.244176	0.144983	0.001139	0.243885	0.148327	0.001233
NVDA	0.591614	0.360672	0.002375	0.590873	0.355023	0.002392
BA	0.452191	0.274204	0.001539	0.448541	0.280247	0.001763
C	0.101620	0.061314	0.001299	0.099478	0.062115	0.001329
INTC	0.080715	0.046255	0.001584	0.076219	0.045570	0.001563
MS	0.196934	0.107767	0.001234	0.181509	0.109321	0.001283
VZ	0.070152	0.037937	0.000973	0.064481	0.039500	0.001006

Źródło: Obliczenia własne.

Tabela 5.6 przedstawia procentowe zmiany poszczególnych błędów prognozy poziomów cen dla badanych spółek. Ujemna wartość oznacza spadek wielkości błędu

Tabela 5.6. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Spółka	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]	Dzienna liczba wiadomości
TSLA	-11.80	-12.04	-7.00	5.42
KO	-9.83	-9.90	-10.33	0.40
NKE	-6.87	-5.57	-11.98	0.98
GM	-6.37	-6.86	-5.63	1.47
F	-4.65	-6.68	-2.45	0.27
GS	-2.07	-3.17	-3.41	0.46
MSFT	-0.85	-2.03	-6.23	2.45
CRM	-0.80	-4.74	-11.59	0.67
JPM	0.12	-2.25	-7.65	1.04
NVDA	0.13	1.59	-0.74	1.70
BA	0.81	-2.16	-12.68	1.16
C	2.15	-1.29	-2.20	0.56
INTC	5.90	1.50	1.39	0.77
MS	8.50	-1.42	-3.83	0.25
VZ	8.79	-3.96	-3.26	0.40

Źródło: Obliczenia własne.

prognozy uzyskanej przy wykorzystaniu autorskiego modelu LSTM-FENE względem wielkości błędu uzyskanego przy pomocy modeli typu ARMA-GARCH. W przypadku obliczeń dla poziomów cen również spółki C, MS, INTC i VZ osiągają wyniki widocznie gorsze niż dla modelu bazowego. Oprócz nich dodatkowo spółki JPM, NVDA i BA mają wyniki nieco gorsze niż model bazowy.

Szereg czasowy prognoz stóp zwrotu otrzymanych przy użyciu modelu LSTM-FENE został również porównany z prognozami modelu ARMA-GARCH za pomocą testu statystycznego Diebolda-Mariano (DM) [40]. W niniejszej pracy porównania testem Diebolda-Mariano były obliczany z wykorzystaniem miernika RMSE. Wyniki z wykorzystaniem miernika MAE były analogiczne. Zestawienie rezultatów przedstawia tabela 5.7. Ujemny wynik testu oznacza, że dla danego instrumentu bardziej skuteczny jest model LSTM-FENE. Wiersze, dla których $P < .05$, zostały pogrubione. Wyniki te wskazują, że model LSTM-FENE daje wyniki znacząco lepsze dla spółek TSLA, NKE, KO, BA, CRM, GM, JPM i MSFT, a zatem tych najbardziej medialnych. Wyniki są znacząco gorsze jedynie dla dwóch spółek: INTC i VZ.

Analogicznie przeprowadzony został test Diebolda-Mariano dla RMSE dla prognoz poziomów cen. Porównane zostały prognozy otrzymane przy użyciu modelu

Tabela 5.7. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu przy wykorzystaniu obu metod dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Spółka	Wynik testu DM	Poziom istotności
TSLA	**-4.05**	**5.31E-05**
NKE	**-2.99**	**.004**
KO	**-2.52**	**.012**
BA	**-2.59**	**.010**
CRM	**-2.49**	**.012**
GM	**-2.14**	**.033**
JPM	**-2.28**	**.022**
MSFT	**-1.99**	**.047**
GS	-0.90	.367
NVDA	-0.29	.770
F	-0.35	.726
C	1.11	.266
MS	1.77	.077
INTC	**1.97**	**.049**
VZ	**2.46**	**.014**

Źródło: Obliczenia własne.

Tabela 5.8. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen przy wykorzystaniuobu metod dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Spółka	Wynik testu DM	Poziom istotności
TSLA	**-3.28**	**.001**
KO	**-2.87**	**.005**
NKE	**-2.46**	**.014**
GM	**-2.45**	**.014**
F	**-2.15**	**.032**
GS	**-1.97**	**.049**
MSFT	**-1.10**	**.271**
CRM	**-0.69**	**.492**
JPM	0.26	.798
NVDA	0.86	.390
BA	1.13	.258
C	1.84	.066
INTC	**2.06**	**.039**
MS	**2.78**	**.006**
VZ	**2.70**	**.007**

Źródło: Obliczenia własne.

LSTM-FENE z prognozami otrzymanymi z modelu ARMA-GARCH. Zestawienie rezultatów przedstawia tabela 5.8. Tak jak w przypadku stóp zwrotu, ujemny wynik testu oznacza, że dla danego instrumentu bardziej skuteczny jest model LSTM-FENE. Wiersze, dla których $P < .05$, zostały pogrubione. Prognozy poziomów cen są znacząco

lepsze dla modelu LSTM-FENE w przypadku spółek TSLA, KO, NKE, GM, F, GS, MSFT oraz CRM, a znacząco gorsze w przypadku INTC, MS i VZ.

Tabela 5.9. Błędy prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych kryptowalut i metali szlachetnych - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

	LSTM-FENE		ARMA-GARCH	
Instrument	RMSE	MAE	RMSE	MAE
Dogecoin	0.003638	0.002006	0.003661	0.001997
Litecoin	0.002625	0.001663	0.002636	0.001659
Ethereum	0.002344	0.001397	0.002351	0.001389
Cardano	0.002322	0.001450	0.002329	0.001445
Bitcoin	0.001647	0.000957	0.001651	0.000952
Złoto	0.000567	0.000374	0.000568	0.000374
Srebro	0.001277	0.000856	0.001277	0.000855

Źródło: Obliczenia własne.

Model LSTM-FENE został także przeanalizowany pod kątem zastosowania dla prognozowania kursów metali szlachetnych i kryptowalut. Analogiczne pomiary skuteczności prognozowania stóp zwrotu dla tych instrumentów finansowych zostały przedstawione w tabelach 5.9 oraz 5.10. Model LSTM-FENE osiągnął lepsze wyniki pod względem miernika RMSE dla wszystkich badanych kryptowalut i metali szlachetnych.

Tabela 5.10. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych kryptowalut i metali szlachetnych - porównanie pomiIdzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Instrument	ΔRMSE [%]	ΔMAE [%]	Dzienna liczba wiadomości
Dogecoin	-0.63	0.47	3.73
Litecoin	-0.43	0.24	0.74
Ethereum	-0.30	0.53	9.23
Cardano	-0.29	0.33	2.08
Bitcoin	-0.23	0.54	21.62
Złoto	-0.16	-0.03	23.29
Srebro	-0.03	0.07	5.39

Źródło: Obliczenia własne.

Tabela 5.11 przedstawia porównanie błędów prognozy poziomów cen kryptowalut i metali szlachetnych otrzymane przy wykorzystaniu autorskiego modelu LSTM-FENE oraz bazowego modelu ARMA-GARCH. W tabeli 5.12 znajdują się wartości procentowych zmian odpowiednich błędów wynikających z zastosowania autorskiego modelu, zamiast modeli klasycznych. Dla prognoz poziomów cen model LSTM-FENE również

osiągnął lepsze wyniki pod względem miernika RMSE dla wszystkich badanych kryptowalut i metali szlachetnych.

Tabela 5.11. Błędy prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych kryptowalut i metali szlachetnych - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

	LSTM-FENE			ARMA-GARCH		
Instrument	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Dogecoin	0.000352	0.000167	0.002006	0.000354	0.000166	0.001997
Litecoin	0.163518	0.102354	0.001663	0.164311	0.102117	0.001659
Ethereum	3.380586	2.016226	0.001397	3.392216	2.005878	0.001389
Bitcoin	31.800549	18.620194	0.000957	31.884488	18.525539	0.000952
Cardano	0.000946	0.000585	0.001450	0.000948	0.000583	0.001445
Złoto	0.974674	0.644492	0.000374	0.976254	0.644711	0.000374
Srebro	0.026200	0.017530	0.000856	0.026211	0.017518	0.000855

Źródło: Obliczenia własne.

Tabela 5.12. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych kryptowalut imetali szlachetnych - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Instrument	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]	Dzienna liczba wiadomości
Dogecoin	-0.56	0.60	0.45	3.73
Litecoin	-0.48	0.23	0.24	0.74
Ethereum	-0.34	0.52	0.53	9.23
Cardano	-0.24	0.31	0.33	2.08
Bitcoin	-0.26	0.51	0.54	21.62
Złoto	-0.16	-0.03	0.00	23.29
Srebro	-0.04	0.07	0.08	5.39

Źródło: Obliczenia własne.

Tabela 5.13. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu przy wykorzystaniu obu metod dla badanych kryptowalut i metali szlachetnych - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Instrument	Wynik testu DM	Poziom istotności
Dogecoin	**-2.58**	**.010**
Litecoin	**-2.14**	**.033**
Ethereum	-1.66	.096
Cardano	-1.50	.134
Bitcoin	-1.26	.208
Złoto	-1.05	.295
Srebro	-0.33	.742

Źródło: Obliczenia własne.

Tabela 5.14. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen przy wykorzystaniu obu metod dla badanych kryptowalut i metali szlachetnych - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Instrument	Wynik testu DM	Poziom istotności
Dogecoin	**-2.76**	**.006**
Litecoin	**-2.25**	**.024**
Ethereum	-1.89	.059
Bitcoin	-1.32	.188
Cardano	-1.73	.084
Złoto	-1.23	.219
Srebro	-0.52	.601

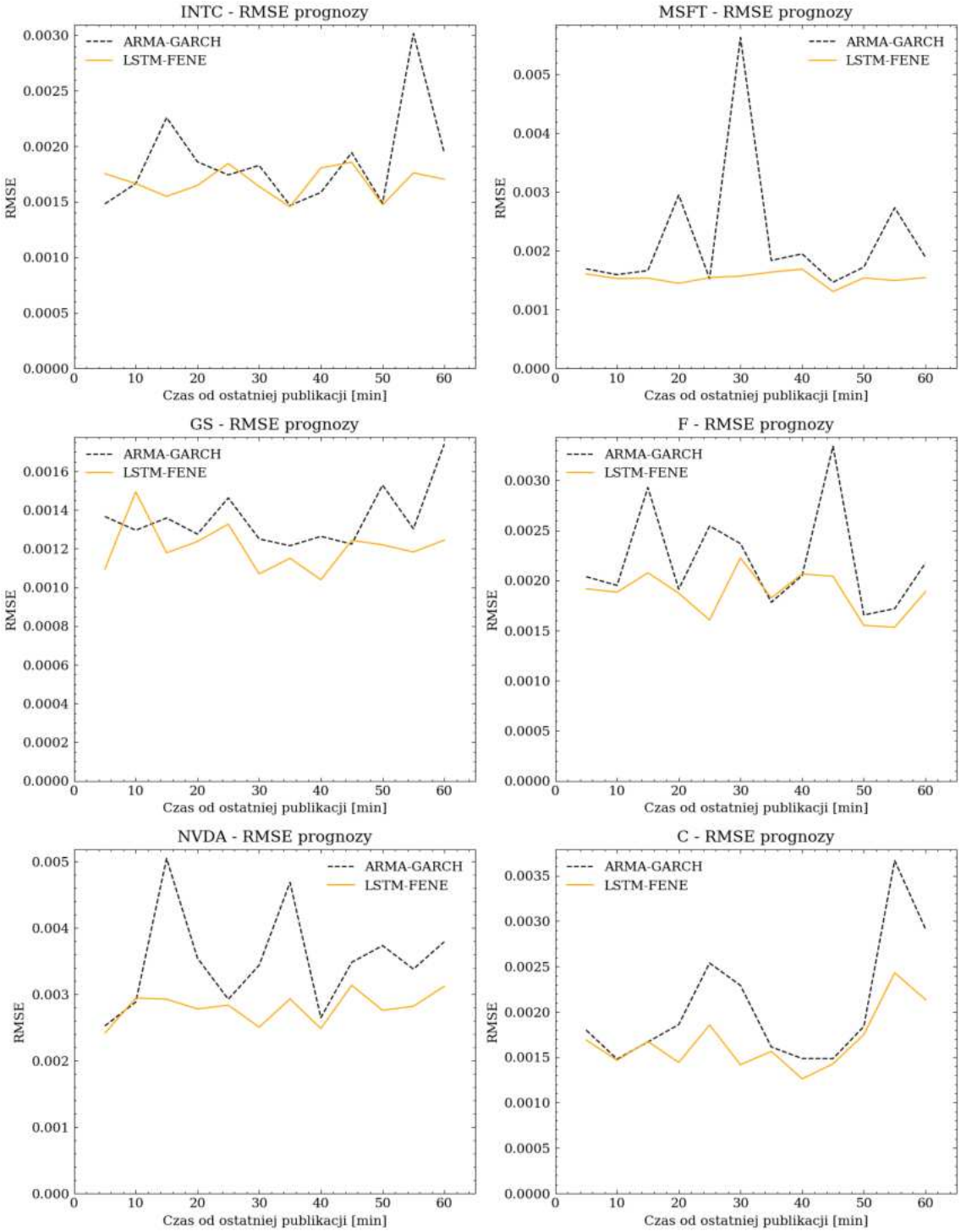
Źródło: Obliczenia własne.

Tabele 5.13 oraz 5.14 zawierają wyniki testów Diebolda-Mariano dla RMSE przeprowadzonych odpowiednio na prognozach stóp zwrotu i prognozach poziomów cen. W obu przypadkach porównywany jest autorski model LSTM-FENE z bazowym modelem ARMA-GARCH. Wyniki otrzymane przez autorski model są lepsze w statystycznie znaczący sposób wyłącznie dla kryptowalut Dogecoin oraz Litecoin.

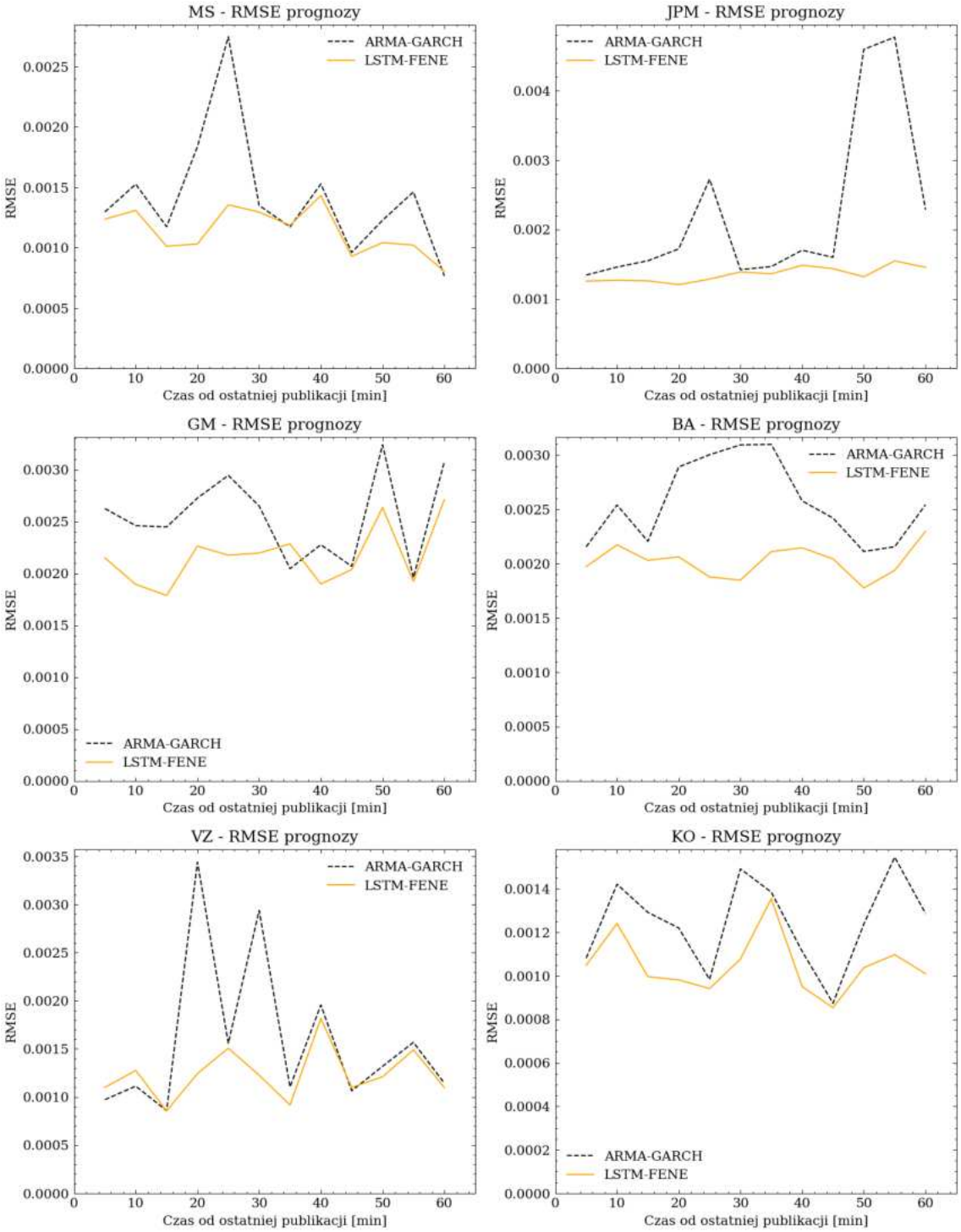
Poprawa trafności prognoz stóp zwrotu dla kryptowalut i metali szlachetnych nie jest tak duża, jak w przypadku spółek giełdowych. Może to wynikać z faktu, iż rynki kryptowalut i metali szlachetnych charakteryzują się znacznie większą zmiennością i jeszcze większą ilością pojawiających się wiadomości niż w przypadku klasycznych rynków giełdowych. Potwierdzenie słuszności tej hipotezy wymagałoby przeprowadzenia badań przy większej rozdzielczości czasowej.

5.2.2. Porównanie zachowania modeli po wystąpieniu nowych wiadomości

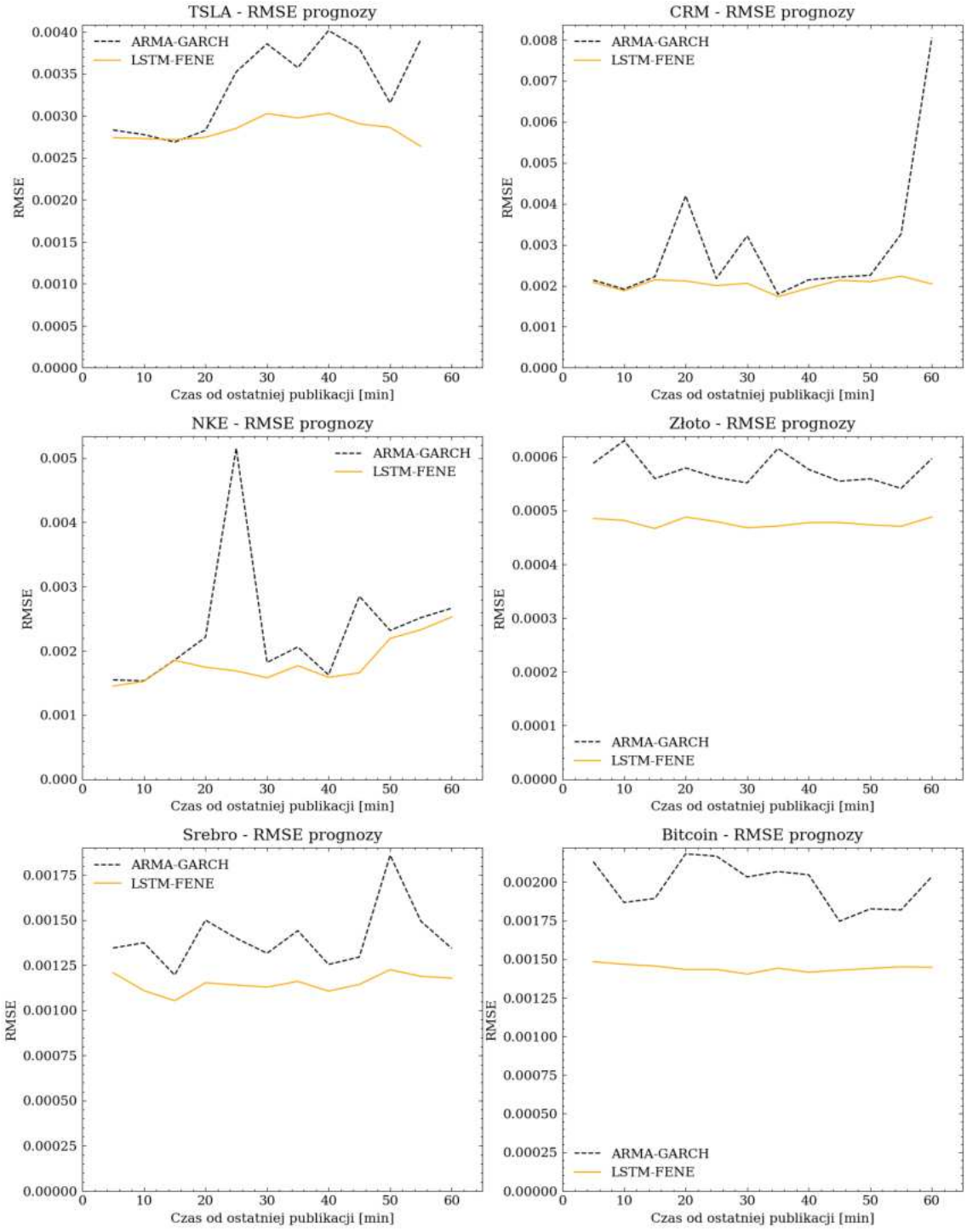
Działanie modelu LSTM-FENE opiera się na wykorzystywaniu dodatkowych informacji zawartych w wiadomościach na temat spółek. Największą przewagę nad klasycznymi modelami ARMA-GARCH osiąga zatem model w okresach bezpośrednio po wystąpieniu nowych publikacji na temat modelowanych spółek. Rysunki 5.2, 5.3, 5.4 oraz 5.5 przedstawiają przebieg błędu RMSE stóp zwrotu dla badanych spółek, kryptowalut i metali szlachetnych w ciągu godziny od pojawienia się nowych wiadomości. Wykresy zostały przygotowane wykorzystując bibliotekę SciencePlots [63].



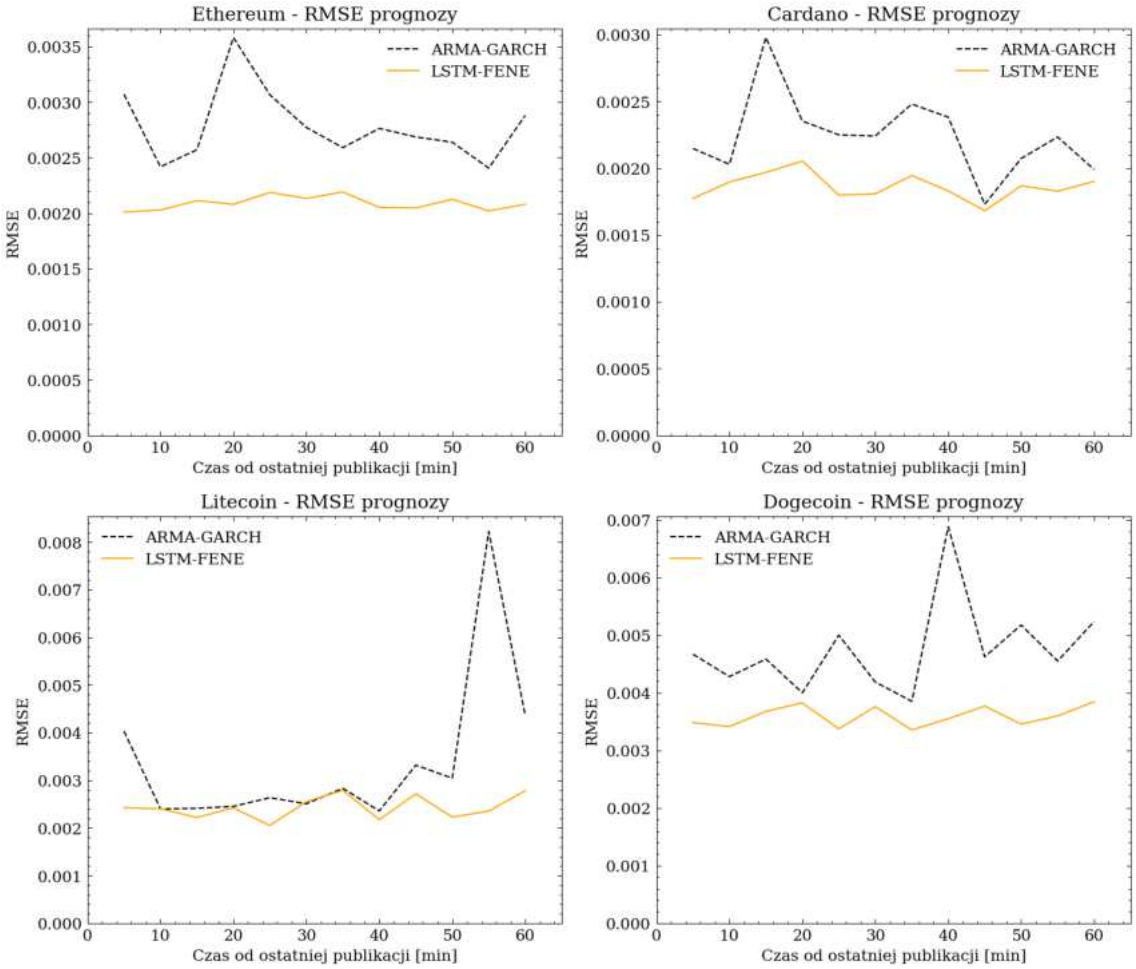
Rysunek 5.2. Uśrednione błędy prognoz RMSE dla prognoz stóp zwrotu w okresie 1 godziny od pojawienia się wiadomości - część 1 (im mniejsza wartość miernika, tym lepszy model). Źródło: Obliczenia własne.



Rysunek 5.3. Uśrednione błędy prognoz RMSE dla prognoz stóp zwrotu w okresie 1 godziny od pojawienia się wiadomości - część 2 (im mniejsza wartość miernika, tym lepszy model). Źródło: Obliczenia własne.

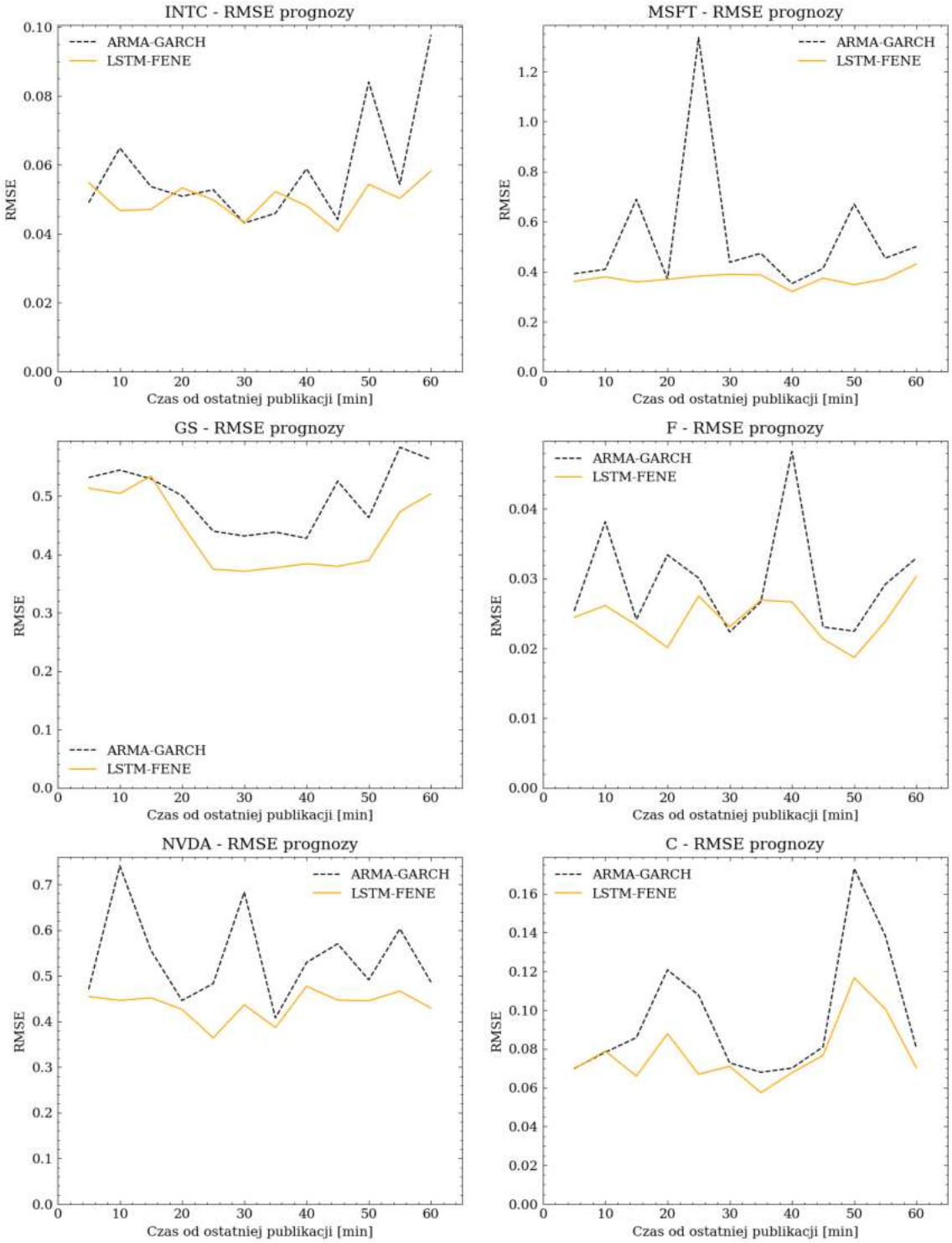


Rysunek 5.4. Uśrednione błędy prognoz RMSE dla prognoz stóp zwrotu w okresie 1 godziny od pojawienia się wiadomości - część 3 (im mniejsza wartość miernika, tym lepszy model). Źródło: Obliczenia własne.

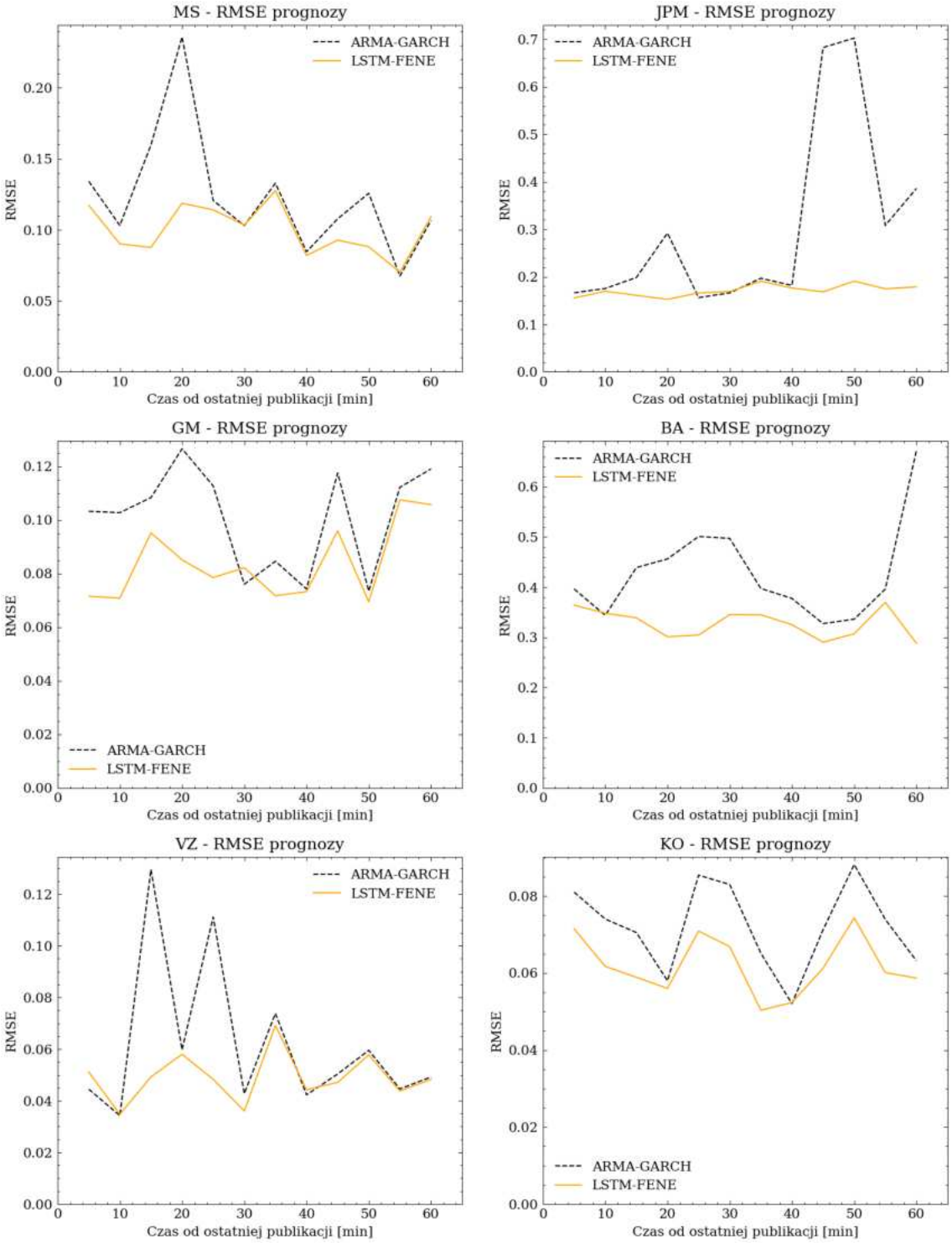


Rysunek 5.5. Uśrednione błędy prognoz RMSE dla prognoz stóp zwrotu w okresie 1 godziny od pojawienia się wiadomości - część 4 (im mniejsza wartość miernika, tym lepszy model). Źródło: Obliczenia własne.

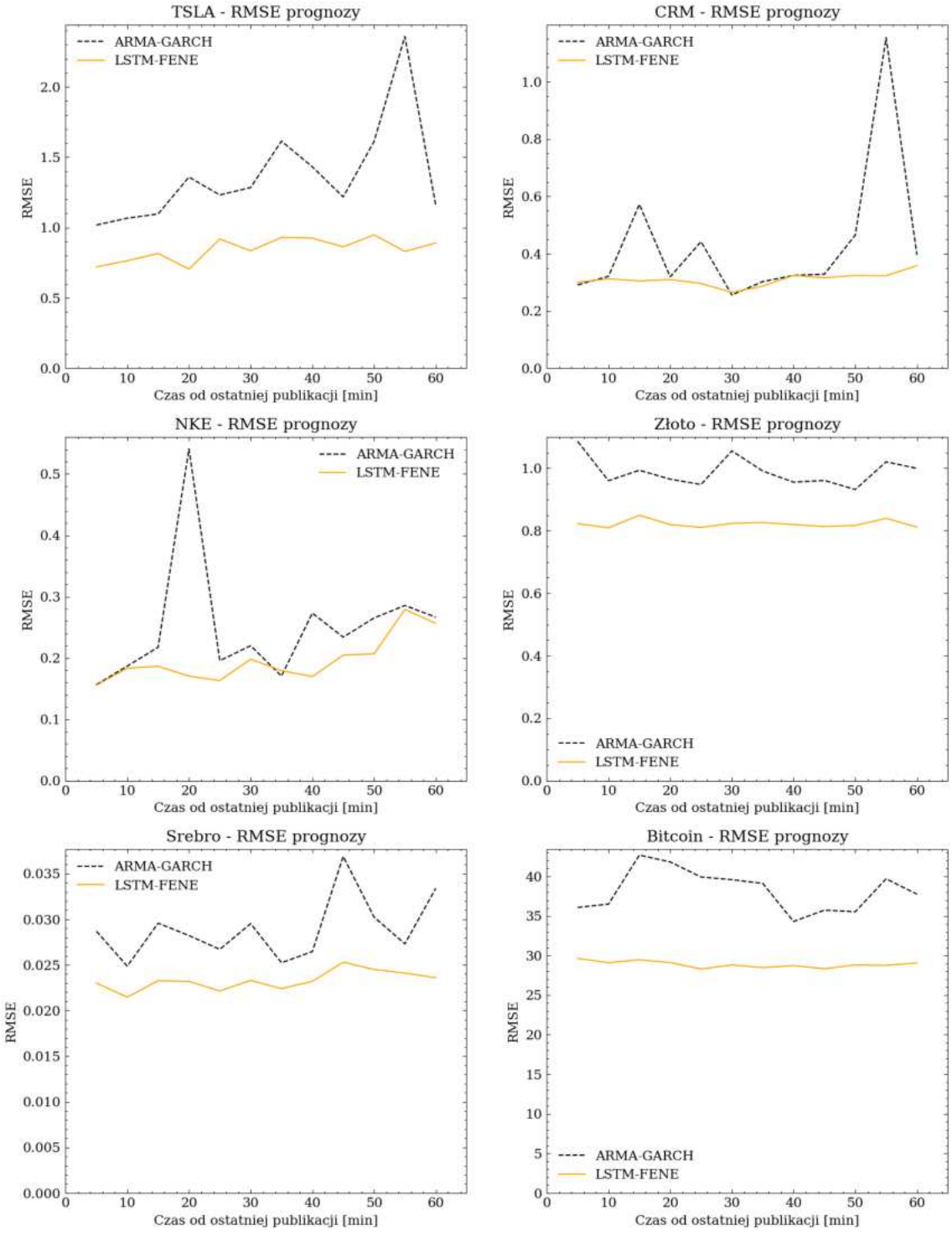
Analogiczne wykresy zostały przygotowane dla prognoz poziomów cen. Rysunki 5.6, 5.7, 5.8 i 5.9 przedstawiają przebieg błędu RMSE poziomów cen dla badanych spółek, kryptowalut i metali szlachetnych w ciągu godziny od pojawienia się nowych wiadomości.



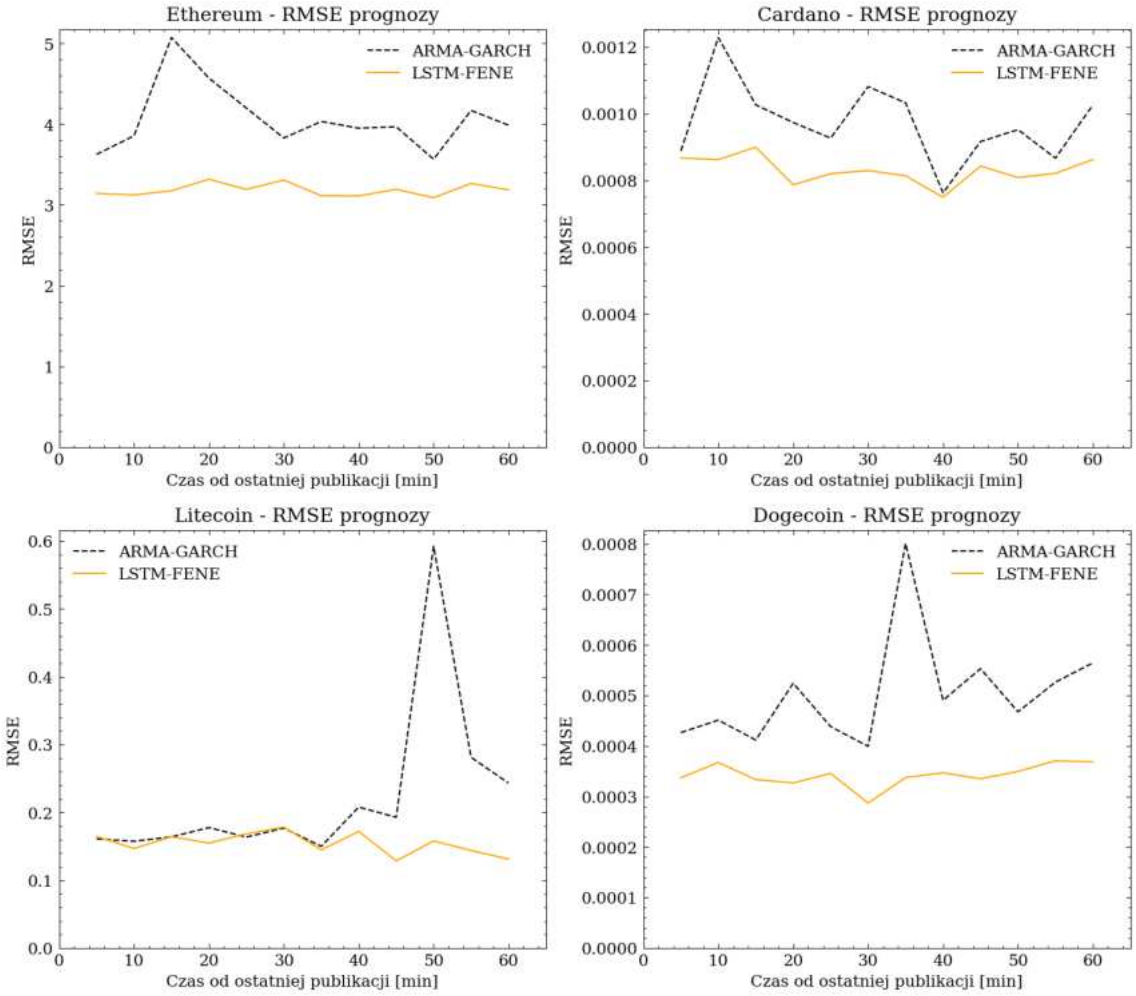
Rysunek 5.6. Uśrednione błędy prognoz RMSE dla prognoz poziomu cen w okresie 1 godziny od pojawienia się wiadomości - część 1 (im mniejsza wartość miernika, tym lepszy model). Źródło: Obliczenia własne.



Rysunek 5.7. Uśrednione błędy prognoz RMSE dla prognoz poziomu cen w okresie 1 godziny od pojawienia się wiadomości - część 2 (im mniejsza wartość miernika, tym lepszy model). Źródło: Obliczenia własne.



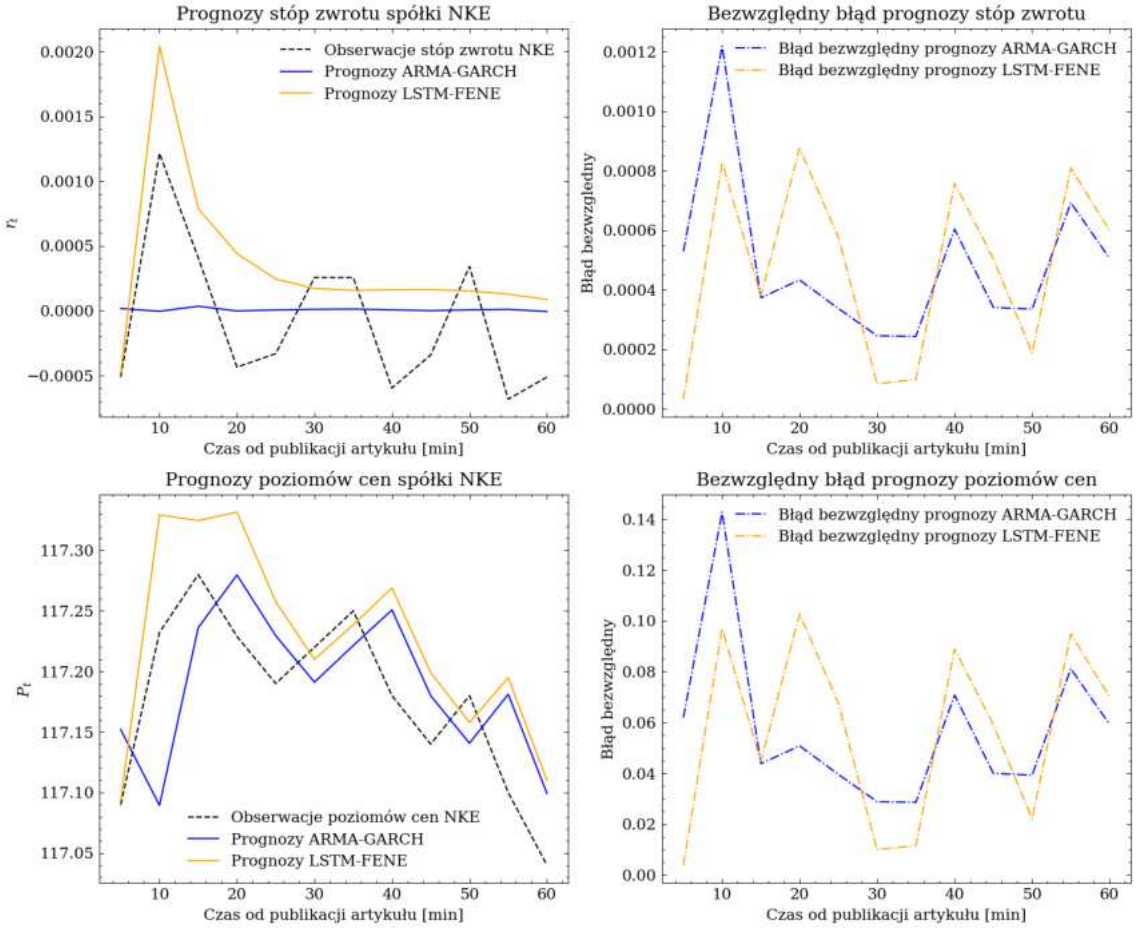
Rysunek 5.8. Uśrednione błędy prognoz RMSE dla prognoz poziomu cen w okresie 1 godziny od pojawienia się wiadomości - część 3 (im mniejsza wartość miernika, tym lepszy model). Źródło: Obliczenia własne.



Rysunek 5.9. Uśrednione błędy prognoz RMSE dla prognoz poziomu cen w okresie 1 godziny od pojawienia się wiadomości - część 4 (im mniejsza wartość miernika, tym lepszy model). Źródło: Obliczenia własne.

Przykładowe przebiegi prognoz stóp zwrotu po wystąpieniu konkretnych wiadomości pozwalają zaobserwować w jaki sposób oba modele reagują na dodatkowe informacje w nich zawarte. Rysunek 5.10 przedstawia wykres reakcji modelu na publikację na temat Nike Inc. Tytuł, treść i źródło wiadomości znajdują się w tabeli 5.15. W przeciwieństwie do modelu ARMA-GARCH, model LSTM-FENE silniej reaguje na pojawienie się pozytywnych wiadomości na temat spółki.

Kolejnym przykładem jest reakcja modeli na wiadomości na temat spółki Coca-Cola Company. Na rysunku 5.11 widać lekko spóźnioną reakcję na wiadomości o pozytywnym wydźwięku dla spółki (względem jej bezpośredniego konkurenta). Tabela 5.16 przedstawia źródło, tytuł i treść wiadomości.



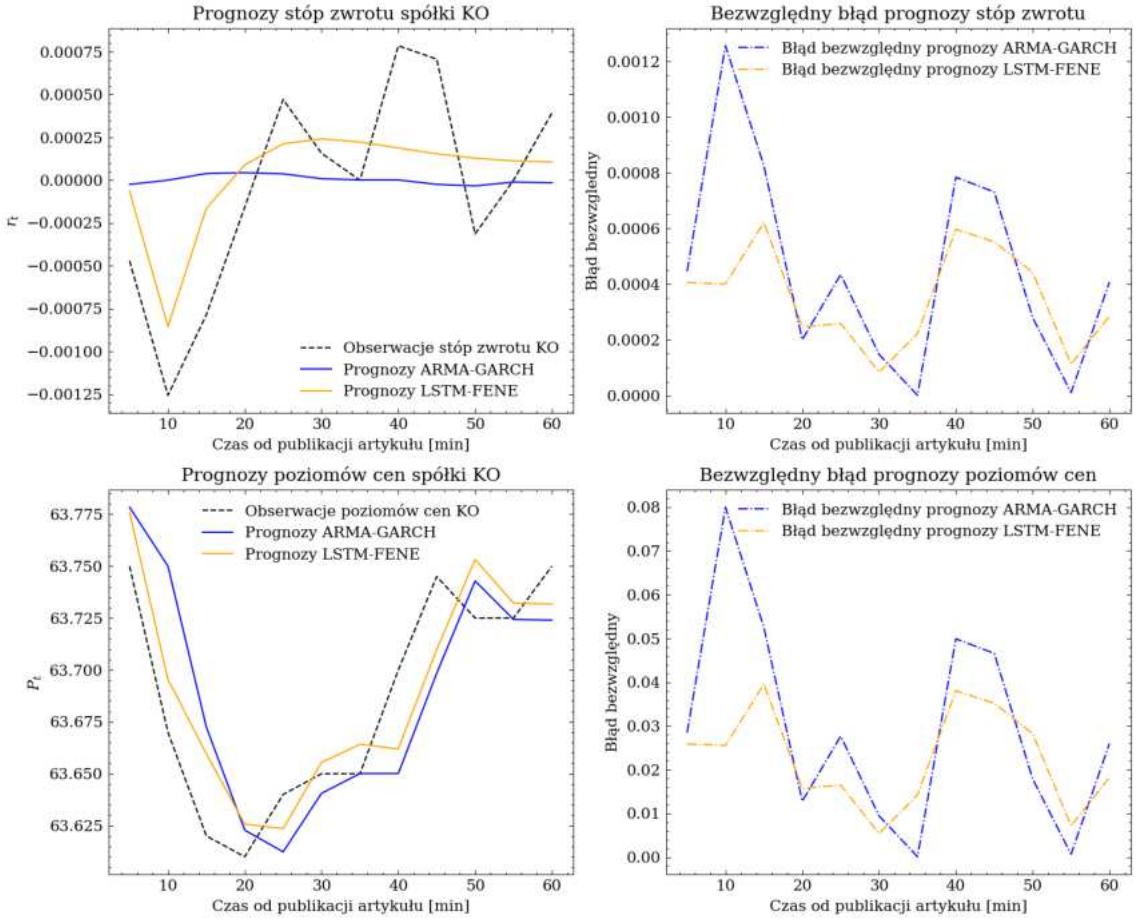
Rysunek 5.10. Wartości rzeczywiste, prognozy i bezwzględne błędy prognoz dla poziomów cen i stóp zwrotu dla spółki NKE po wystąpieniu wiadomości na jej temat - porównanie modeli LSTM-FENE i ARMA-GARCH.

Źródło: Obliczenia własne.

Tabela 5.15. Wiadomość na temat modelowanej spółki - przykład 1.

Spółka	NKE
Źródło	financialpost.com
Tytuł	European shares post best day in nearly 6 weeks as upbeat Nike results lift retailers.
European shares rose on Wednesday, as banks extended gains and upbeat results from Nike boosted the region's battered retail sector, while improving economic outlook for the euro zone also lifted sentiment. The region-wide STOXX 600 index closed 1.7% higher, with consumer discretionary firms such as Adidas and Puma leading gains after U.S. peer Nike beat quarterly revenue and profit expectations. Article content Shares of Adidas and Puma rose 6.8% and 9.5%, respectively, while London-listed JD Sports jumped 6.1%, lifting the retail index. Article content Rising borrowing costs and soaring cost of living have bludgeoned the European retail sector this year, with the industry index on track for its worst annual performance since 2008. "It looks like there is a last-ditch effort to achieve a Santa rally," said Victoria Scholar, head of investment at Interactive Investor. "Lighter-than-normal volumes around the holidays could also be exacerbating the moves." Germany's DAX index rose 1.5%. A GfK institute survey showed consumer sentiment in Germany, Europe's largest economy, is set to extend its recovery as government relief measures meant to take the bite out of soaring energy prices seem to be having an effect. This follows figures on Tuesday showing an uptick in euro zone consumer confidence Article content in December. "The market was pricing a very bad recession with potentially rolling blackouts and stop of industrial plans due to insufficient supply of energy and now the market is pricing out a bit of that," said Davide Oneglia, chief Europe economist at TS Lombard. "The winter is just really beginning so we don't know what is going to happen, but at the moment certainly markets are feeling more positive about the outlook and so that's good for stocks." The euro STOXX 50 volatility index hit its lowest level since January, reflecting easing anxiety among investors. The STOXX 600 has shed 2% so far in December after two straight months of gains that were underpinned by hopes that cooling inflation would allow the Federal Reserve and other central banks to go slow on interest rate hikes. Banks advanced 1.6% but were still on track for an annual loss, as fears of recession outweighed gains from rising interest rates. Philips gained 5.7% after the Dutch health technology company said independent tests on its respiratory devices involved in a major global recall had shown positive results. British retailers also reported a surprise pick-up in demand in December, a Confederation of British Industry survey showed. (Reporting by Amruta Khandekar and Bansari Mayur Kamdar; editing by Uttaresh.V, Saumyadeb Chakrabarty and Maju Samuel)	

Źródło: Opracowanie własne.



Rysunek 5.11. Wartości rzeczywiste, prognozy i bezwzględne błędy prognoz dla poziomów cen i stóp zwrotu dla spółki KO po wystąpieniu wiadomości na jej temat - porównanie modeli LSTM-FENE i ARMA-GARCH.

Źródło: Obliczenia własne.

Tabela 5.16. Wiadomość na temat modelowanej spółki - przykład 2.

Spółka	KO
Źródło	barrons.com
Tytuł	KO vs. PEP: Why Coca-Cola Stock Looks Tastier

Investors and traders have been in recession mode for much of this year, leading to debates about which industries can be classified as recession-resistant. For some consumers, beverages and snacks straddle the line between discretionary and staples, making this industry a bit challenging for investors to classify. In this piece, we compared KO and PEP, the two leading soda stocks. When analyzing the two, KO stock came out slightly ahead. Coca-Cola (NYSE:KO) and PepsiCo (NASDAQ:PEP) are both up 9% to 7.5% year to date, demonstrating that they hold up better than most of the broader market. Although these companies have gone head-to-head for decades, they utilize slightly different strategies, as clearly seen in the types of brands they own. Coca-Cola (KO) A key difference between Coca-Cola Co. and PepsiCo, Inc. is that the former owns a wide array of beverage brands, while the latter also owns Frito Lay and many other snack brands. Both companies have pricing power due to the strength of their brands, but Coca-Cola is significantly more profitable, and its P/E multiple is slightly less volatile. Thus, a bullish view appears appropriate. While PepsiCo and Coca-Cola are both stalwarts in the food and beverage industry, one thing that makes Coca-Cola more attractive is its profitability. Its gross margin is stable, hovering around 60% annually compared to PepsiCo's 55% margin. However, a deeper dive reveals just how much more profitable Coca-Cola is. The company experienced a hiccup during the pandemic, as its sales fell from $37.3 billion in 2019 to $33 billion in 2020, probably due to lost sales from shuttered restaurants. However, Coca-Cola recovered quickly, reaching $38.7 billion in sales in 2021 and $42.3 billion in the last 12 months. Meanwhile, the beverage maker's net income rose from a pre-pandemic $9 billion in 2019 to $9.8 billion in 2021 and $10 billion over the last 12 months. PepsiCo's numbers are below, but Coca-Cola is significantly more profitable despite its smaller overall size. At around 27.5 times, Coca-Cola's P/E ratio has been relatively stable since December 2020, having fallen back down to earth in early 2019 after irrational exuberance drove its P/E ratio to well over 100 times. There could be more upside for Coca-Cola from current levels, but what makes this stock so attractive is its stability and solid dividend yield of 2.8%, compared to the sector average of 2.1%. What is the Price Target for KO Stock? Coca-Cola has a Strong Buy consensus rating based on 10 Buys, two Holds, and zero Sell ratings over the last three months. At $65.58, the average price target for Coca-Cola implies upside potential of 3%. PepsiCo (PEP) Dividend investors can't go wrong with Coca-Cola or PepsiCo, but PepsiCo's lower profitability, more volatile P/E multiple, and lower dividend yield are enough reasons to favor Coca-Cola — albeit only slightly. A neutral view generally looks appropriate for PepsiCo, although dividend investors might still want to hold it. On the plus side, PepsiCo's revenues grew steadily, even during the pandemic, jumping significantly from $67.2 billion in 2019 to $70.4 billion in 2020 and $79.5 billion in 2021. PepsiCo's net income rose from $7.4 billion in 2019 to $7.7 billion in 2021. For the last 12 months, the beverage maker recorded $83.6 billion in sales and $9.7 billion in net income. However, it is significantly less profitable than Coca-Cola. PepsiCo's yield of 2.5% is attractive for dividend investors, but its P/E ratio has been volatile, dropping from about 31 times earlier this year to about 26 times. The company's P/E plunged from about 33 times to 15 times in 2019, although it looks fairly valued now since Coca-Cola deserves a bit of a premium due to its higher profitability. What is the Price Target for PEP Stock? PepsiCo has a Moderate Buy consensus rating based on seven Buys, five Holds, and one Sell rating assigned over the last three months. At $181, the average price target for PepsiCo implies downside potential of 1.1%. Conclusion: Bullish on Coca-Cola, Neutral on PepsiCo Ultimately, Coca-Cola and PepsiCo both look like solid long-term plays, especially for dividend investors during periods of economic uncertainty. While Coca-Cola looks slightly better due to its higher profitability and more stable valuation, both are attractive due to their dividend yields and stability during recessions. Both are dividend kings, meaning they've boosted their dividends annually for at least the last 50 years, and they're both "value stocks," which typically do better in a recession. In short, investors who already own PepsiCo might not want to sell it, and either way, dividend investors might still want to own the shares. However, those who can pick only one may prefer Coca-Cola. Disclosure

Źródło: Opracowanie własne.

Wyniki osiągnięte z wykorzystaniem autorskiego modelu LSTM-FENE są lepsze od wyników osiąganych za pomocą klasycznych modeli ekonometrycznych. Udoskonalona zdolnosć prognostyczna wynika z wykorzystania dodatkowych informacji zawartych w wiadomościach finansowych, które zostały przeanalizowane przy pomocy technik NLP, tj. odpowiedniego modelu językowego w połączeniu z siecią neuronową typu LSTM przetwarzającą szeregi czasowe stóp zwrotu. Potwierdzona została zatem postawiona we wstępie hipoteza, że *możliwe jest efektywne zautomatyzowanie analizy finansowych wiadomości oraz postów w mediach społecznościowych poprzez użycie współczesnych metod przetwarzania języka naturalnego – głównie głębokich sieci neuronowych.*

Prowadząc badania na danych zgromadzonych dla 5-minutowych interwałów udało się dodatkowo potwierdzić hipotezę, że *wykorzystanie szeregu czasowego notowań o 5 minutowej częstotliwości jest wystarczające, aby zaobserwować zmienioną charakterystykę zwrotów spowodowaną publikacją wiadomości.*

5.3. Rodzaje wiadomości i ich wpływ na notowania giełdowe

5.3.1. Ocena jakości różnych źródeł informacji

Wiadomości pojawiające się w serwisach informacyjnych mogą mieć różny wpływ na prognozę nie tylko ze względu na zawartą w nich treść, ale także z powodu wiarygodności i popularności źródła, z którego pochodzą. Model LSTM-FENE uwzględnia źródło pochodzenia danych w postaci dodatkowej zmiennej, która różnym źródłom przypisuje wektory w trakcie procesu treningu, tzw. *embeddings*.

Ostatecznie model może wykorzystać informację o pochodzeniu opublikowanej wiadomości przypisując większą wagę tym źródłom, które są rzeczywiście opiniotwórcze wśród inwestorów i wywierają wpływ na dalsze przebiegi stóp zwrotów.

W tabeli 5.17 znajduje się zestawienie poprawy błędów prognozy stóp zwrotu modelu LSTM-FENE względem modeli ARMA-GARCH w zależności od źródła pochodzenia wiadomości użytej w prognozach. W zestawieniu zostały wykorzystane tylko źródła o znacznej (wyższej niż średnia) dziennej liczbie publikowanych wiadomości, aby wyeliminować wartości odstające źródeł, które pojawiają się wyłącznie kilkukrotnie w całym okresie walidacyjnym.

Można zaobserwować, że poprawa błędu prognozy stóp zwrotu (ujemna wartość $\Delta RMSE$ i ΔMAE) jest obecna dla wszystkich źródeł o dużej liczbie publikacji. Wy-

Tabela 5.17. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu w zależności od źródła wykorzystywanych wiadomości - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Źródło	ΔRMSE [%]	ΔMAE [%]	Dzienna liczba wiadomości
heraldsun.com.au	-31.41	-37.00	0.12
standard.co.uk	-30.47	-31.87	0.27
forbes.com	-24.63	-24.63	0.01
forbesmiddleeast.com	-24.19	-31.17	0.02
wired.com	-23.31	-25.79	0.07
cbc.ca	-21.90	-23.14	0.83
allnews.ng	-21.72	-25.20	0.05
bbc.com	-20.51	-30.98	0.07
zeebiz.com	-18.92	-22.03	0.02
businessinsider.in	-18.80	-28.68	0.03
foxnews.com	-18.55	-23.71	0.31
dw.com	-18.33	-18.33	0.01
washingtonpost.com	-16.23	-17.58	0.31
realmoney.thestreet.com	-16.07	-16.27	0.52
fxstreet.com	-13.13	-10.95	0.22
guardian.ng	-10.95	-15.37	0.05
siasat.com	-10.85	-15.40	0.16
fxempire.com	-10.78	-11.95	0.46
financialpost.com	-10.47	-9.34	3.47
indiatoday.in	-10.25	-12.78	0.15

Źródło: Obliczenia własne.

korzystanie informacji zawartych w publikacjach z tych źródeł pomaga podnieść jakość prognoz. Do źródeł powodujących największą poprawę błędów prognoz można zaliczyć znane i opiniotwórcze firmy medialne: Herald Sun, London Evening Standard, Forbes (oraz zlokalizowane wersje Forbes), Wired, BBC, Fox News, The Washington Post, The Guardian, Financial Post. Zgodnie z oczekiwaniami te źródła informacji wywołują najsilniejsze reakcje wśród inwestorów. Na ich podstawie autorski model LSTM-FENE osiąga najlepsze wyniki.

Podobne wyniki zostały otrzymane na podstawie prognoz poziomów cen wygenerowanych przez modele LSTM-FENE oraz ARMA-GARCH. Tabela 5.18 zawiera wyniki otrzymane dla wszystkich źródeł o dużej liczbie publikacji. Większość serwisów informacyjnych pokrywa się w obu przypadkach.

Tabela 5.18. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen w zależności od źródła wykorzystywanych wiadomości - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Źródło	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]	Dzienna liczba wiadomości
zerohedge.com	-30.99	-37.86	-40.71	0.40
realmoney.thestreet.com	-22.99	-21.60	-24.15	0.28
nypost.com	-14.96	-19.53	-15.08	0.59
fortune.com	-11.40	-10.86	-1.53	0.39
barrons.com	-11.38	-12.40	-3.70	0.93
reuters.com	-9.74	-6.57	-10.52	0.93
thestreet.com	-9.48	-12.87	-5.04	1.41
investing.com	-9.27	-10.69	-12.06	2.53
investors.com	-9.18	-11.98	-9.29	0.98
business-standard.com	-9.16	-16.95	-11.39	0.47
fxempire.com	-8.89	-1.89	-20.92	0.27
marketwatch.com	-8.54	-12.35	-12.21	1.35
financialpost.com	-8.12	-10.08	-12.54	1.65
timesofindia.indiatimes	-6.80	-10.38	-3.78	0.30
cnet.com	-5.60	-7.10	-4.19	0.38
investorplace.com	-3.82	-10.96	-15.96	1.58
businessinsider.com	-3.35	-0.33	2.32	0.81
tipranks.com	-2.34	-1.26	1.15	0.69
economictimes.indiatimes	-1.31	4.37	12.14	0.39
cnbc.com	-1.30	-5.08	-4.81	1.21

Źródło: Obliczenia własne.

5.3.2. Badania siły efektu wiadomości na różnych typach rynków

Siła wpływu wykorzystania wiadomości jest zależna od typu rynku, z którego pochodzą modelowane instrumenty finansowe. Wyróżnione zostały 3 kategorie: spółki akcyjne, metale szlachetne oraz kryptowaluty. Tabela 5.19 zawiera zestawienie poprawy błędów prognoz dla prognoz stóp zwrotu dla każdego z typów instrumentów finansowych wykorzystanych w niniejszych badaniach.

Uśrednione błędy prognoz dla każdego z rynków obliczone dla prognoz poziomów cen zostały zebrane w tabeli 5.20.

Poprawa jakości prognoz stóp zwrotu i poziomów cen opartych o model LSTM-FENE jest największa w przypadku spółek akcyjnych. Rynki kryptowalut i metali szlachetnych są wyjątkowo płynne, dlatego wykorzystanie 5-minutowych interwałów może być niewystarczające, zaś wpływ nowych wiadomości mniejszy lub krótszy, niż w przypadku spółek akcyjnych. W celu odrzucenia lub przyjęcia tej hipotezy,

Tabela 5.19. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu w zależności od typu analizowanego rynku - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Instrumenty finansowe	ΔRMSE [%]	ΔMAE [%]
Spółki akcyjne	-4.56	-6.19
Metale szlachetne	-0.10	0.02
Kryptowaluty	-0.38	0.42

Źródło: Obliczenia własne.

Tabela 5.20. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen w zależności od typu analizowanego rynku - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Instrumenty finansowe	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]
Spółki akcyjne	-1.12	-3.93	-5.84
Metale szlachetne	-0.10	0.02	0.03
Kryptowaluty	-0.39	0.41	0.42

Źródło: Obliczenia własne.

konieczne byłyby dalsze badania z wykorzystaniem trudno dostępnych danych na poziomie pojedynczych *tick*ów.

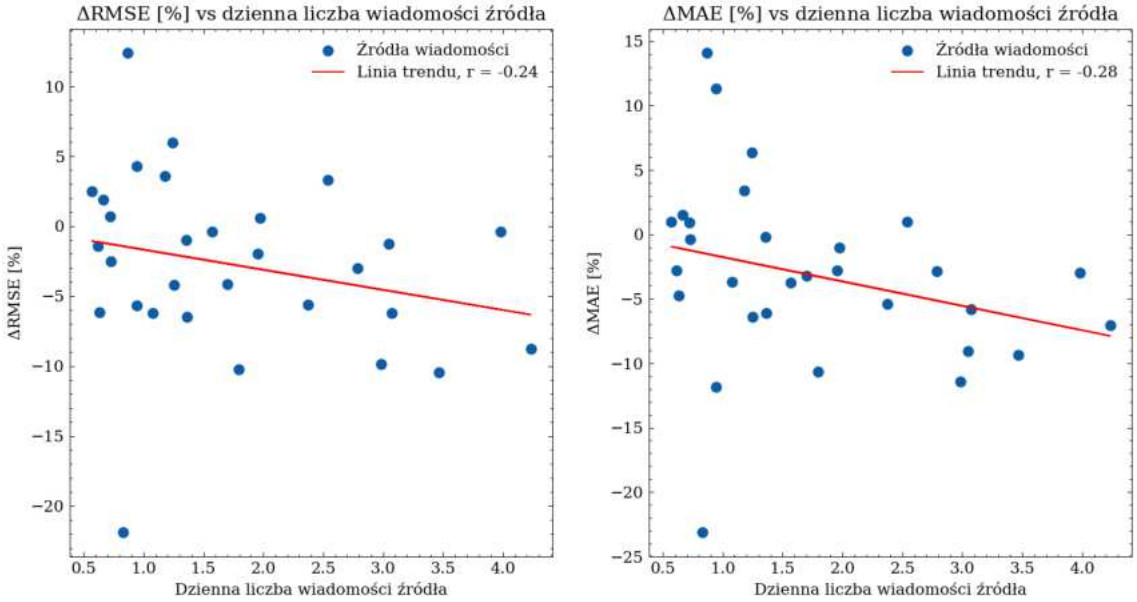
Engle R. i Sun Z. przedstawili modele zmienności dla danych na poziomie pojedynczych *tick*ów, uwzględniające mikrostrukturę rynku [55]. Ich model osiąga dobre wyniki, jednak jednym z najważniejszych wniosków z ich badań były duże trudności z oddzieleniem sygnału od szumu. W przypadku autorskich modeli dodatkowym wyzwaniem byłaby konieczność bardzo dokładnego dopasowania czasu pojawienia się danych tekstowych (publikowania wiadomości) tak, aby poszczególne transakcje mogły być skorelowane z wykorzystaniem nowych informacji przez inwestora.

5.4. Zależność jakości predykcji od liczby wiadomości finansowych

5.4.1. Badania wpływu dziennej liczby wiadomości na podatność wycen na zmiany

Uwzględnienie nowych publikacji ma pozytywny wpływ na jakość modelu. Mimo to, sama ilość pojawiających się wiadomości nie przekłada się bezpośrednio na stopień poprawy prognoz.

Rysunek 5.12 przedstawia wykresy poprawy błędów prognozy stóp zwrotu RMSE oraz MAE dla każdego ze źródeł informacji i odpowiadającej mu średniej dziennej liczbie wiadomości. Na wykresach zaznaczono też linię regresji. W obu przypadkach wartość współczynnika korelacji liniowej r (również zaznaczony na wykresie) wskazuje na słabą zależność między badanymi wielkościami.

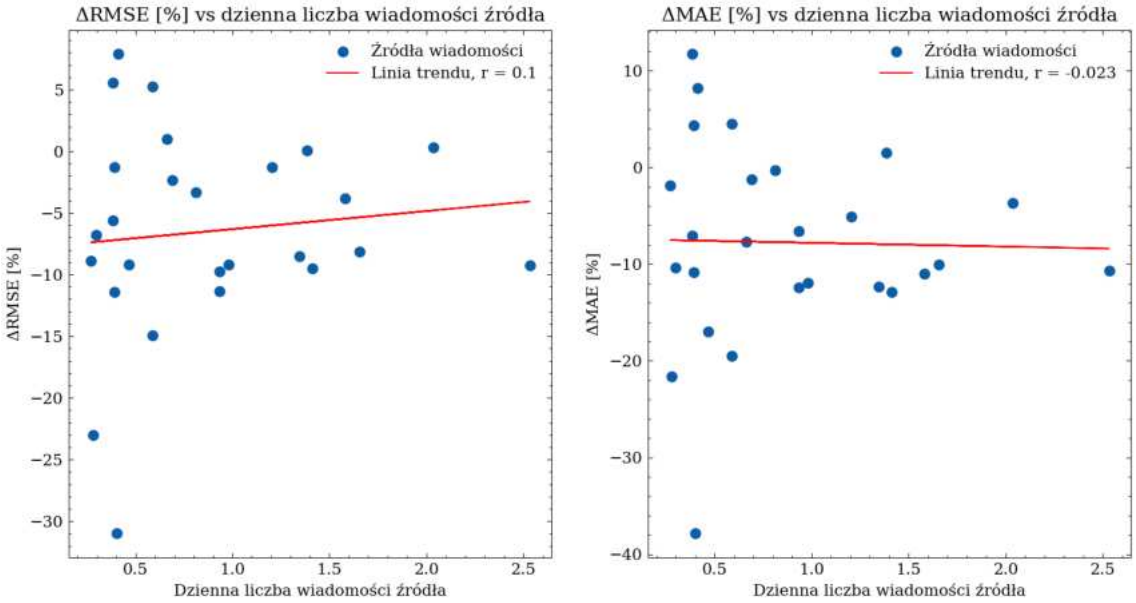


Rysunek 5.12. Zależność procentowej poprawy błędów prognozy stóp zwrotu RMSE oraz MAE od średniej dziennej liczby wiadomości pochodzących z danego źródła - diagram korelacyjny, linia regresji oraz wartość współczynnika korelacji liniowej r.
Źródło: Obliczenia własne.

Wyniki dla prognoz poziomów cen znajdują się na rysunku 5.13. Obliczone zostały zmiany błędów prognozy poziomów cen RMSE oraz MAE dla każdego ze źródeł informacji wraz z odpowiadającą mu średnią dzienną ilością wiadomości. Na wykresach zaznaczono linię regresji. Współczynniki regresji liniowej r mają bardzo niską wartość, co wskazuje na brak bezpośredniej zależności między dzienną liczbą wiadomości a wynikami otrzymywanymi przez model.

W tabeli 5.21 zostały przedstawione wyniki regresji liniowej obliczonej na zmianach błędów prognoz w zależności od dziennej liczby wiadomości dla różnych źródeł. W tabeli tej zostały przedstawione wartości współczynniki linii regresji znajdujących się na rysunkach 5.12 i 5.13.

Analogiczne obliczenia zostały przeprowadzone dla grupowania danych po instrumencie finansowym. Na rysunku 5.14 znajdują się wykresy zależności poprawy błędów RMSE oraz MAE dla stóp zwrotu od dziennej ilości wiadomości dla każdego z badanych instrumentów finansowych. Również w tym przypadku nie ma dostrzegalnie istotnej



Rysunek 5.13. Zależność procentowej poprawy błędów prognozy poziomu cen RMSE oraz MAE od średniej dziennej liczby wiadomości pochodzących z danego źródła - diagram korelacyjny, linia regresji oraz wartość współczynnika korelacji liniowej r.
Źródło: Obliczenia własne.

Tabela 5.21. Wyniki regresji liniowej zmian błędów prognoz w zależności od dziennej liczby wiadomości dla różnych źródeł.

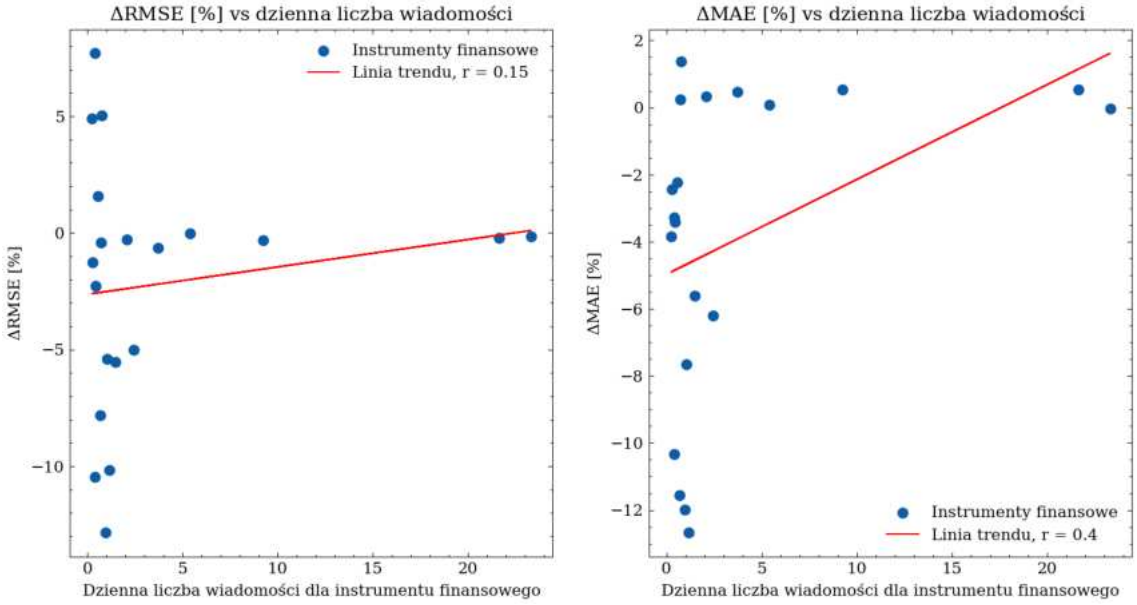
Prognozowana wielkość	Miernik jakości prognoz	Współczynnik	Wartość wsp.	Błąd std.	t	$P > \|t\|$
Stopy zwrotu	ΔRMSE	Wyraz wolny	-0.25	2.22	-0.11	0.912
		Wsp. nach.	-1.44	1.09	-1.32	0.197
	ΔMAE	Wyraz wolny	0.13	2.45	0.05	0.958
		Wsp. nach.	-1.89	1.20	-1.57	0.127
Poziom cen	ΔRMSE	Wyraz wolny	-7.78	3.03	-2.57	0.017
		Wsp. nach.	1.47	2.86	0.51	0.613
	ΔMAE	Wyraz wolny	-7.42	3.71	-2.00	0.057
		Wsp. nach.	-0.39	3.50	-0.11	0.913

Źródło: Obliczenia własne.

korelacji. Szczególnie procentowa zmiana błędu ΔMAE nie wskazuje na zależność od dziennej liczby wiadomości dla danego instrumentu.

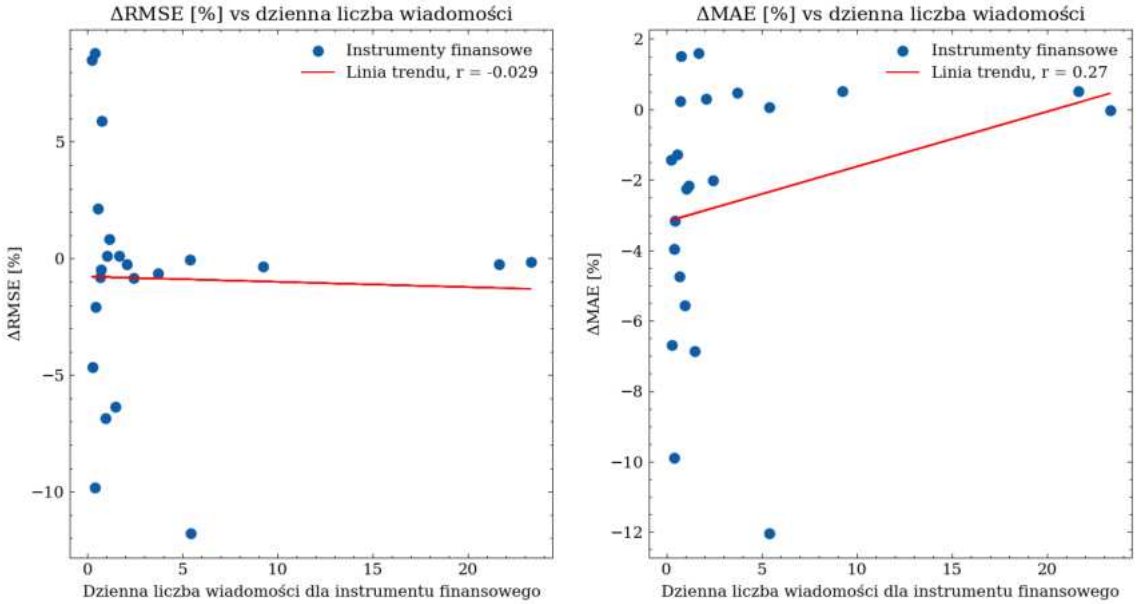
Rysunek 5.15 przedstawia wykresy zależności poprawy błędów RMSE oraz MAE dla poziomów cen, od dziennej ilości wiadomości dla każdego z badanych instrumentów finansowych. Zależność między badanymi zmiennymi jest słaba.

Analogicznie tabela 5.22 przedstawia wyniki regresji liniowej dla zmian błędów prognoz w zależności od dziennej liczby wiadomości dla różnych instrumentów finanso-



Rysunek 5.14. Zależność procentowej poprawy błędów prognozy stóp zwrotu RMSE oraz MAE od średniej dziennej liczby wiadomości opublikowanych na temat dane instrumentu finansowego - diagram korelacyjny, linia regresji oraz wartość współczynnika korelacji liniowej r.

Źródło: Obliczenia własne.



Rysunek 5.15. Zależność procentowej poprawy błędów prognozy poziomu cen RMSE oraz MAE od średniej dziennej liczby wiadomości opublikowanych na temat dane instrumentu finansowego - diagram korelacyjny, linia regresji oraz wartość współczynnika korelacji liniowej r.

Źródło: Obliczenia własne.

wych. W tabeli tej zostały przedstawione wartości współczynniki linii regresji znajdujących się na rysunkach 5.14 i 5.15.

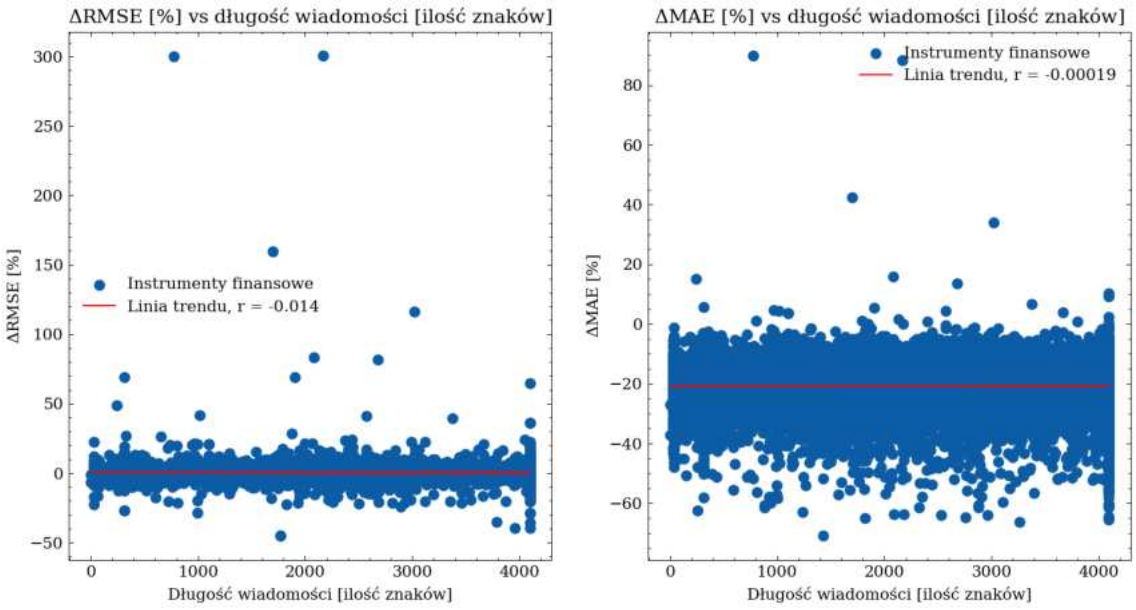
Tabela 5.22. Wyniki regresji liniowej zmian błędów prognoz w zależności od dziennej liczby wiadomości dla różnych instrumentów finansowych.

Prognozowana wielkość	Miernik jakości prognoz	Współczynnik	Wartość wsp.	Błąd std.	t	$P > \|t\|$
Stopy zwrotu	ΔRMSE	Wyraz wolny	-2.64	1.41	-1.86	0.079
		Wsp. nach.	0.12	0.19	0.63	0.536
	ΔMAE	Wyraz wolny	-4.97	1.15	-4.33	0.000
		Wsp. nach.	0.28	0.15	1.87	0.078
Poziom cen	ΔRMSE	Wyraz wolny	-0.78	1.28	-0.61	0.550
		Wsp. nach.	-0.02	0.17	-0.13	0.898
	ΔMAE	Wyraz wolny	-3.18	0.92	-3.48	0.002
		Wsp. nach.	0.16	0.12	1.26	0.224

Źródło: Obliczenia własne.

5.4.2. Długość pojedynczych wiadomości a ilość niesionych informacji

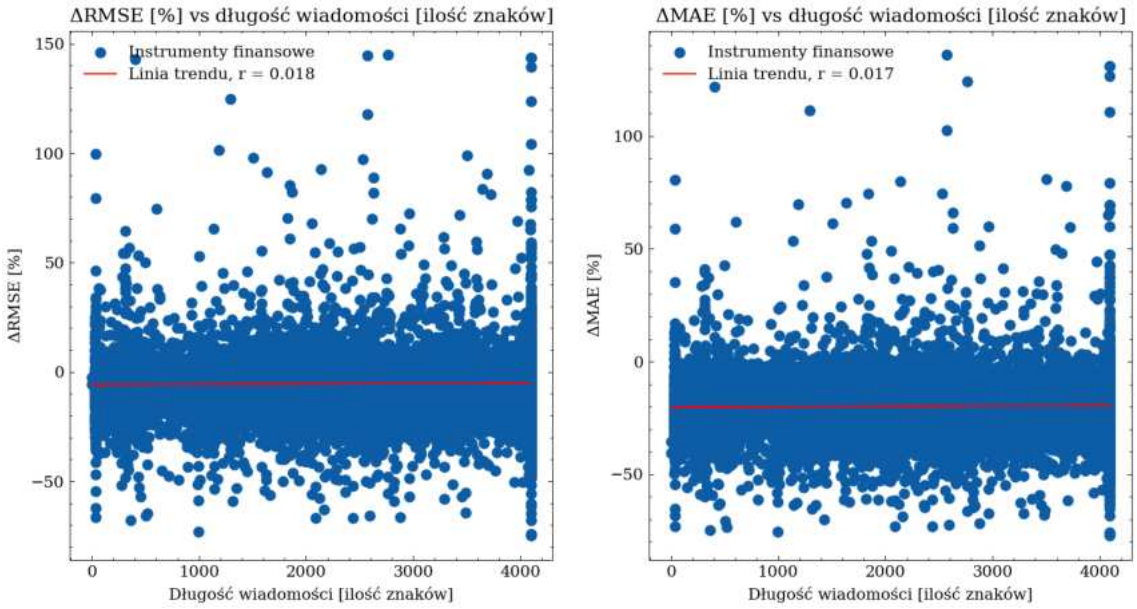
Zbadano także wpływ długości ogłaszanej wiadomości na procentową poprawę błędów prognozy stóp zwrotu. Ze względu na liczne skrótowce i symbole specyficzne dla tekstów o tematyce finansowej, do określenia wiadomości posłużyła ilość znaków po wcześniejszym usunięciu powtarzających się tzw. białych znaków. Maksymalna długość wiadomości wynika z możliwości przetwarzania sieci typu BERT.



Rysunek 5.16. Zależność procentowej poprawy błędów prognozy stóp zwrotu RMSE oraz MAE od długości publikowanej wiadomości - diagram korelacyjny, linia regresji oraz wartość współczynnika korelacji liniowej r.
Źródło: Obliczenia własne.

Zależność procentowej zmiany błędów prognozy stóp zwrotu od długości wiadomo-

ści została przedstawiona na rysunku 5.16. Także w tym przypadku korelacja między poprawą błędów, a długością wykorzystywanych wiadomości, jest słaba.



Rysunek 5.17. Zależność procentowej poprawy błędów prognozy poziomu cen RMSE oraz MAE od długości publikowanej wiadomości - diagram korelacyjny, linia regresji oraz wartość współczynnika korelacji liniowej r.
Źródło: Obliczenia własne.

Z kolei rysunek 5.17 ukazuje zależność procentowej zmiany błędów prognozy poziomów cen od długości wiadomości. Nie ma zauważalnej korelacji między tymi wielkościami.

W tabeli 5.23 zostały przedstawione wyniki regresji liniowej obliczonej na zmianach błędów prognoz w zależności od długości wiadomości. W tabeli tej zostały przedstawione wartości współczynniki linii regresji znajdujących się na rysunkach 5.16 i 5.17.

Mimo zauważalnej poprawy jakości prognoz stóp zwrotu, dzięki wykorzystaniu informacji zawartych w wiadomościach finansowych, nie udało się zaobserwować, jaka dokładnie ilość nowych informacji daje najlepszy efekt. Nie istnieje znacząca korelacja między stopniem poprawy prognoz, a średnią liczbą wiadomości pojawiających się dla konkretnego instrumentu ani średnią liczbą wiadomości pochodzących z konkretnego źródła. Nie zaobserwowano również znaczącej zależności między długością publikacji a procentową zmianą błędów prognoz.

Tabela 5.23. Wyniki regresji liniowej zmian błędów prognoz w zależności od długości wiadomości.

Prognozowana wielkość	Miernik jakości prognoz	Współczynnik	Wartość wsp.	Błąd std.	t	$P > \|t\|$
Stopy zwrotu	ΔRMSE	Wyraz wolny	0.40	0.09	4.25	0.000
		Wsp. nach.	-5.96E-05	3.39E-05	-1.76	0.079
	ΔMAE	Wyraz wolny	-20.83	0.15	-139.81	0.000
		Wsp. nach.	-1.24E-06	5.34E-05	-0.02	0.981
Poziom cen	ΔRMSE	Wyraz wolny	-5.85	0.27	-21.32	0.000
		Wsp. nach.	2.00E-04	9.97E-05	1.94	0.053
	ΔMAE	Wyraz wolny	-20.18	0.27	-73.66	0.000
		Wsp. nach.	2.00E-04	9.96E-05	1.80	0.071

Źródło: Obliczenia własne.

5.5. Jakość proponowanych modeli w zależności od hiperparametrów

Wyjątkowo istotnym etapem tworzenia modelu prognostycznego jest dobór jego hiperparametrów. Ze względu na znaczny czas treningu modelu LSTM-FENE, nie było możliwe pełne przeszukanie przestrzeni hiperparametrów, tzw. *grid search*. Wobec tego zbadane zostały tylko najbardziej istotne kombinacje hiperparametrów pod kątem ich wpływu na poprawę jakości predykcji względem klasycznych modeli z klasy ARMA-GARCH.

Przeszukiwanie przestrzeni zostało podzielone na dwa etapy. W pierwszym zmieniane są wyłącznie hiperparametry modelu językowego. W drugim dobierane są hiperparametry części przetwarzającej finansowe szeregi czasowe. Oprócz wyboru najkorzystniejszych wartości hiperparemetrów, istotna jest również stabilność wyników. Zachowanie modelu nie powinno różnić się znacznie przy niewielkich zmianach hiperparametrów. Opisane w rozdziałach 5.2, 5.3 oraz 5.4 wyniki bazują na najlepszym zbadanym zbiorze hiperparametrów.

5.5.1. Dobór hiperparametrów modelu NLP

Moduł FENE jest opisywany przez trzy najbardziej istotne parametry: N - ilość bloków typu Transformer w modelu, d_{model} - wielkość wektora *word embeddings* i h - ilość wątków mechanizmu uwagi w każdym z N bloków. Parametry te zostały szerzej opisane w rozdziale 4.2.

W tabelach 5.24 oraz 5.25 zestawione zostały wyniki zmian hiperparametrów, które miały bezpośrednie przełożenie na wielkość wykorzystywanego modelu językowego. Tabela 5.24 zawiera wyniki dla prognoz stóp zwrotu, zaś tabela 5.25 wyniki modeli z tymi samymi kombinacjami hiperparametrów przeliczonych na poziomy cen.

Tabela 5.24. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych spółek indeksu S&P500 dla różnych kombinacji hiperparametrów modelu językowego FENE - porównanie pomiędzy modelem LSTM-FENE z różnymi kombinacjami hiperparametrów modułu FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Hiperparametry				
N	d_{model}	h	ΔRMSE [%]	ΔMAE [%]
2	**128**	**2**	**-3.99**	**-6.00**
4	512	8	-4.44	-5.89
10	768	12	19.54	80.18
12	128	2	-4.42	-5.99
12	512	8	-1.62	-5.34

Źródło: Obliczenia własne.

Tabela 5.25. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych spółek indeksu S&P500 dla różnych kombinacji hiperparametrów modelu językowego FENE - porównanie pomiędzy modelem LSTM-FENE z różnymi kombinacjami hiperparametrów modułu FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Hiperparametry					
N	d_{model}	h	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]
2	**128**	**2**	**-0.57**	**-4.12**	**-7.08**
4	512	8	-0.96	-4.25	-6.41
10	768	12	19.70	34.78	21.81
12	128	2	-0.80	-3.88	-6.10
12	512	8	1.46	-3.77	-6.28

Źródło: Obliczenia własne.

W tabelach 5.26 i 5.27 znajdują się analogiczne wyniki (dla odpowiednio: stóp zwrotu i poziomów cen) obliczone dla kryptowalut i metali szlachetnych.

W przypadku modelowania tak spółek giełdowych, jak i metali szlachetnych oraz kryptowalut dobrym kompromisem zapewniającym poprawę błędów RMSE oraz MAE jest stosunkowo niewielki model BERT o hiperparametrach $N = 2$, $d_{model} = 128$, $h = 2$. Tak wybrany model językowy składa się w sumie z 4,4 miliona parametrów (w porównaniu do 110 milionów parametrów oryginalnego modelu BERT). Dodatkową zaletą wyboru małego modelu językowego jest jego szybszy trening i generowanie

Tabela 5.26. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych metali szlachetnych i kryptowalut dla różnych kombinacji hiperparametrów modelu językowego FENE - porównanie pomiędzy modelem LSTM-FENE z różnymi kombinacjami hiperparametrów modułu FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Hiperparametry				
N	d_{model}	h	ΔRMSE [%]	ΔMAE [%]
2	**128**	**2**	**-0.27**	**0.40**
4	512	8	-0.12	0.74
10	768	12	0.62	2.71
12	128	2	2.35	6.76
12	512	8	-0.26	0.46

Źródło: Obliczenia własne.

Tabela 5.27. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz poziomu cen dla badanych metali szlachetnych i kryptowalut dla różnych kombinacji hiperparametrów modelu językowego FENE - porównanie pomiędzy modelem LSTM-FENE z różnymi kombinacjami hiperparametrów modułu FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Hiperparametry					
N	d_{model}	h	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]
2	**128**	**2**	**-0.22**	**0.39**	**0.41**
4	512	8	-0.11	0.70	0.73
10	768	12	0.58	2.60	2.70
12	128	2	2.38	6.57	6.74
12	512	8	-0.31	0.44	0.46

Źródło: Obliczenia własne.

prognoz. We wszystkich powyższych tabelach wybrana kombinacja hiperparametrów została zaznaczona jako pogrubiony wiersz.

5.5.2. Dobór hiperparametrów modeli szeregów czasowych

Po przyjęciu (i ustaleniu) hiperparametrów modelu językowego, wybierane są hiperparametry części LSTM przetwarzającej finansowe szeregi czasowe. Najistotniejsze hiperparametry tej części modelu to d_{hidden} - ilość neuronów warstwy LSTM, $d_{encoder}$ - wielkość enkodera (długość wektora, który ostatecznie przekazywany jest dalej z części tekstowej), N_{LSTM} - liczba warstw LSTM; a także D - *dropout*, tj wielkość opisująca prawdopodobieństwo, z jakim neurony mogą być wyłączane w procesie treningu w celu regularyzacji.

Tabele 5.28 oraz 5.29 zawierają wyniki poprawy błędów prognozy odpowiednio stóp zwrotu oraz poziomów cen w zależności od doboru kombinacji hiperparametrów części LSTM przetwarzającej finansowe szeregi czasowe - ujemna wartość oznacza przewagę

Tabela 5.28. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych spółek akcyjnych dla różnych kombinacji hiperparametrów modelu szeregów czasowych LSTM - porównanie pomiędzy modelem LSTM-FENE z różnymi kombinacjami hiperparametrów modułu LSTM a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Hiperparametry					
N_{LSTM}	D	d_{hidden}	$d_{encoder}$	ΔRMSE [%]	ΔMAE [%]
2	0.05	16	16	-4.54	-6.13
1	0.05	32	16	-4.55	-6.12
1	0.05	16	16	-3.99	-6.00
1	0.05	8	16	-3.33	-4.64
1	**0.05**	**4**	**16**	**-4.56**	**-6.19**
1	0.05	2	16	-4.56	-6.18
1	0.05	16	32	-4.55	-6.12
1	0.05	16	8	-4.55	-6.14
1	0.05	16	4	-4.55	-6.17
1	0.05	4	4	-4.49	-5.96
1	0.2	16	16	-4.24	-4.46
1	0.5	16	16	-4.53	-6.11

Źródło: Obliczenia własne.

Tabela 5.29. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych spółek akcyjnych dla różnych kombinacji hiperparametrów modelu szeregów czasowych LSTM - porównanie pomiędzy modelem LSTM-FENE z różnymi kombinacjami hiperparametrów modułu LSTM a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Hiperparametry						
N_{LSTM}	D	d_{hidden}	$d_{encoder}$	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]
2	0.05	16	16	-1.09	-4.05	-6.47
1	0.05	32	16	-0.96	-3.79	-6.83
1	0.05	16	16	-0.57	-4.12	-7.08
1	0.05	8	16	-0.15	-2.98	-6.24
1	**0.05**	**4**	**16**	**-1.12**	**-3.93**	**-6.33**
1	0.05	2	16	-0.99	-3.77	-6.55
1	0.05	16	32	-1.06	-4.08	-6.17
1	0.05	16	8	-1.02	-3.70	-6.45
1	0.05	16	4	-1.05	-4.34	-5.85
1	0.05	4	4	-1.11	-3.64	-6.42
1	0.2	16	16	-0.87	-3.67	-5.67
1	0.5	16	16	-1.12	-3.99	-6.44

Źródło: Obliczenia własne.

badanegomodelu LSTM-FENE nad modelem bazowym ARMA-GARCH. W każdej z tabel pogrubiony został wiersz z ostatecznie wybranymi najkorzystniejszymi hiperparametrami. Analizy przeprowadzone w rozdziałach 5.2, 5.3 oraz 5.4 wykorzystywały właśnie model z tymi hiperparametrami.

Tabela 5.30. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych metali szlachetnych i kryptowalut dla różnych kombinacji hiperparametrów modelu szeregów czasowych LSTM - porównanie pomiędzy modelem LSTM-FENE z różnymi kombinacjami hiperparametrów modułu LSTM a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Hiperparametry					
N_{LSTM}	D	d_{hidden}	$d_{encoder}$	ΔRMSE [%]	ΔMAE [%]
2	0.05	16	16	-0.15	0.79
1	0.05	32	16	-0.26	0.41
1	0.05	16	16	-0.27	0.40
1	0.05	8	16	-0.12	0.80
1	**0.05**	**4**	**16**	**-0.30**	**0.31**
1	0.05	2	16	-0.30	0.39
1	0.05	16	32	-0.12	0.77
1	0.05	16	8	-0.01	1.05
1	0.05	16	4	-0.24	0.47
1	0.05	4	4	-0.19	0.59
1	0.2	16	16	0.20	1.56
1	0.5	16	16	-0.24	0.40

Źródło: Obliczenia własne.

Tabela 5.31. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych metali szlachtnych i kryptowalut dla różnych kombinacji hiperparametrów modelu szeregów czasowych LSTM - porównanie pomiędzy modelem LSTM-FENE z różnymi kombinacjami hiperparametrów modułu LSTM a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Hiperparametry						
N_{LSTM}	D	d_{hidden}	$d_{encoder}$	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]
2	0.05	16	16	-0.13	0.76	0.79
1	0.05	32	16	-0.28	0.40	0.41
1	0.05	16	16	-0.22	0.39	0.41
1	0.05	8	16	-0.15	0.77	0.77
1	**0.05**	**4**	**16**	**-0.31**	**0.30**	**0.31**
1	0.05	2	16	-0.29	0.38	0.39
1	0.05	16	32	-0.09	0.74	0.77
1	0.05	16	8	-0.04	1.02	1.04
1	0.05	16	4	-0.24	0.45	0.46
1	0.05	4	4	-0.19	0.57	0.58
1	0.2	16	16	0.22	1.51	1.55
1	0.5	16	16	-0.26	0.40	0.40

Źródło: Obliczenia własne.

W tabelach 5.30 i 5.31 znajdują się analogiczne wyniki poprawy błędów prognozy w zależności od doboru kombinacji hiperparametrów części przetwarzającej finansowe szeregi czasowe dla metali szlachetnych i kryptowalut. Ujemna wartość oznacza przewagę badanego modelu LSTM-FENE nad modelem bazowym ARMA-GARCH. W

obu tabelach pogrubioną czcionką zaznaczono wybrane kombinacje hiperparametrów modułu LSTM.

Dla wszystkich typów instrumentów finansowych najkorzystniejszy okazał się model o hiperparametrach ($N_{LSTM} = 1$, $D = 0.05$, $d_{hidden} = 4$, $d_{encoder} = 16$). We wszystkich przypadkach faworyzowany był zatem model z mniejszą liczbą parametrów (pojedyncza warstwa LSTM, mała liczba neuronów). Wyniki modelu były jednak stabilne i nie różniły się znacząco przy zmianach hiperparametrów.

5.6. Wykorzystanie nagłówków wiadomości zamiast ich pełnej treści

Po ustaleniu najlepszych hiperparametrów możliwe były dalsze eksperymenty z modelem LSTM-FENE. Kolejnym krokiem było wytrenowanie modelu, który zamiast pełnej treści wiadomości finansowych, wykorzystuje wyłącznie nagłówki. Otrzymane wyniki również świadczą o poprawie błędów prognoz stóp zwrotu dla większości ze spółek, jednak poprawa jest mniejsza niż w przypadku modelu opartego o pełną treść wiadomości.

Tabela 5.32. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych spółek indeksu S&P500 przy użyciu wyłącznie nagłówków wiadomości w module FENE - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym. Dla wygody otrzymane wyniki zostały zestawione wraz z wynikami z tabeli 5.4.

	Wyłącznie nagłówki		Pełna treść wiadomości	
Spółka	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
TSLA	-20.94	-2.82	-25.42	-3.54
BA	-10.68	-12.69	-10.16	-12.68
NKE	-10.16	-10.96	-12.85	-11.99
KO	-9.64	-9.95	-10.46	-10.33
CRM	-6.63	-11.12	-7.83	-11.55
GM	-6.57	-6.56	-5.55	-5.62
JPM	-6.53	-7.80	-5.40	-7.65
GS	-5.07	-5.18	-2.27	-3.42
MSFT	-4.33	-6.14	-5.01	-6.21
C	-2.42	-4.22	1.58	-2.23
F	-2.36	-3.97	-1.25	-2.44
NVDA	-1.66	-8.90	-1.35	-9.51
MS	0.91	-5.19	4.91	-3.84
INTC	1.38	1.00	5.02	1.37
VZ	4.05	-3.16	7.69	-3.27

Źródło: Obliczenia własne.

Tabela 5.33. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych spółek indeksu S&P500 przy użyciu wyłącznie nagłówków wiadomości w module FENE - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym. Dla wygody otrzymane wyniki zostały zestawione wraz z wynikami z tabeli 5.6.

Spółka	Wyłącznie nagłówki			Pełna treść wiadomości		
	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]
TSLA	-10.32	-7.66	-4.13	-11.80	-12.04	-7.00
KO	-9.33	-9.78	-9.95	-9.83	-9.90	-10.33
GM	-6.24	-6.59	-6.56	-6.37	-6.86	-5.63
NKE	-4.98	-5.75	-10.95	-6.87	-5.57	-11.98
F	-4.06	-6.39	-3.97	-4.65	-6.68	-2.45
GS	-3.65	-3.90	-5.17	-2.07	-3.17	-3.41
C	-2.22	-3.73	-4.21	2.15	-1.29	-2.20
INTC	-2.14	-1.57	1.00	5.90	1.50	1.39
JPM	-1.49	-2.69	-7.80	0.12	-2.25	-7.65
CRM	-1.38	-6.40	-11.15	-0.80	-4.74	-11.59
BA	-1.25	-3.01	-12.69	0.81	-2.16	-12.68
MSFT	-1.20	-3.01	-6.16	-0.85	-2.03	-6.23
VZ	1.18	-5.91	-3.16	8.79	-3.96	-3.26
MS	5.23	-2.25	-5.18	8.50	-1.42	-3.83
NVDA	38.39	12.22	-8.97	0.13	1.59	-0.74

Źródło: Obliczenia własne.

Tabela 5.32 przedstawia porównanie błędów prognoz stóp zwrotu modelu LSTM-FENE (opartego o nagłówki) z błędami prognoz bazowych modeli ARMA-GARCH dla badanych spółek indeksu S&P 500. W tabeli 5.33 znajdują się wyniki dla prognoz poziomów cen obliczonych na podstawie prognoz stóp zwrotu. Okazuje się, że model otrzymujący pełne wiadomości osiągnął lepsze rezultaty, szczególnie dla spółek TSLA, NKE, KO, BA. W tabelach z wynikami modelu opartego o nagłówki zostały powtórzone wyniki dla modelu wykorzystującego pełne treści wiadomości z tabel 5.4 i 5.6 w celu łatwiejszego porównania rezultatów).

Prognozy modelu opartego o nagłówki wiadomości zostały porównane z prognozami klasycznych modeli ARMA-GARCH przy pomocy testu Diebolda-Mariano dla RMSE. Wyniki testu dla prognoz stóp zwrotu przedstawia tabela 5.34, zaś dla poziomów cen tabela 5.35.

Poprawa jakości prognoz stóp zwrotu jest statystycznie istotna dla spółek TSLA, BA, NKE, KO, CRM, GM. Poprawa jakości prognoz poziomów cen jest statystycznie istotna dla spółek TSLA, KO, GM, NKE (pogorszenie wyników jest statystycznie istotne dla spółek MS i NVDA). Dla pozostałych spółek również widoczna jest korzystna

Tabela 5.34. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu dla badanych spółek indeksu S&P 500 przy wykorzystaniu wyłącznie nagłówków wiadomości w module FENE - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Spółka	Wynik testu DM	Poziom istotności
TSLA	**-3.74**	**1.92E-04**
BA	**-2.75**	**.006**
NKE	**-2.55**	**.011**
KO	**-2.24**	**.025**
CRM	**-2.20**	**.028**
GM	**-2.07**	**.038**
JPM	-1.95	.052
GS	-1.62	.105
MSFT	-1.62	.105
C	-1.00	.316
F	-1.19	.234
NVDA	-0.93	.352
MS	0.74	.461
INTC	0.97	.333
VZ	1.66	.096

Źródło: Obliczenia własne.

Tabela 5.35. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen dla badanych spółek indeksu S&P 500 przy wykorzystaniu wyłącznie nagłówków wiadomości w module FENE - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Spółka	Wynik testu DM	Poziom istotności
TSLA	**-3.51**	**5.90E-04**
KO	**-3.22**	**.001**
GM	**-2.27**	**.023**
NKE	**-2.02**	**.043**
F	-1.96	.050
GS	-1.62	.105
C	-1.16	.247
INTC	-1.36	.175
JPM	-0.92	.357
CRM	-0.94	.347
BA	-0.95	.343
MSFT	-0.90	.367
VZ	1.01	.312
MS	**2.20**	**.028**
NVDA	**7.02**	**5.57E-12**

Źródło: Obliczenia własne.

zmiana, jednak różnice nie są na zakładanym poziomie istotności 0.05. Wykorzystanie jedynie nagłówków pozwala osiągać wyniki lepsze niż w przypadku modeli bazowych, jednak gorsze niż pełny model LSTM-FENE.

Analogiczne analizy przeprowadzono dla kryptowalut i metali szlachetnych. W ta-

Tabela 5.36. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych metali szlachetnych i kryptowalut przy użyciu wyłącznie nagłówków wiadomości w module FENE - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym. Dla wygody otrzymane wyniki zostały zestawione wraz z wynikami z tabeli 5.10.

	Wyłącznie nagłówki		Pełna treść wiadomości	
Instrument	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
Srebro	-0.57	-0.32	-0.03	0.07
Złoto	-0.15	0.30	-0.16	-0.03
Dogecoin	0.34	1.44	-0.63	0.47
Litecoin	1.05	1.52	-0.43	0.24
Cardano	2.78	2.61	-0.29	0.33
Bitcoin	3.00	3.36	-0.23	0.54
Ethereum	3.08	3.00	-0.30	0.53

Źródło: Obliczenia własne.

Tabela 5.37. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych metali szlachetnych i kryptowalut przy użyciu wyłącznie nagłówków wiadomości w module FENE - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym. Dla wygody otrzymane wyniki zostały zestawione wraz z wynikami z tabeli 5.12.

	Wyłącznie nagłówki			Pełna treść wiadomości		
Instrument	ΔRMSE	ΔMAE	ΔMAPE	ΔRMSE	ΔMAE	ΔMAPE
Dogecoin	-0.73	0.69	1.45	-0.63	0.47	0.48
Srebro	-0.33	-0.03	-0.31	-0.04	0.07	0.08
Litecoin	0.08	0.86	1.52	-0.48	0.23	0.24
Złoto	0.18	0.63	0.30	-0.16	-0.03	-0.03
Ethereum	1.65	2.21	3.00	-0.34	0.52	0.53
Bitcoin	3.47	3.56	3.36	-0.26	0.51	0.54
Cardano	3.94	3.14	2.61	-0.24	0.31	0.33

Źródło: Obliczenia własne.

belach 5.36 i 5.37 opisane są wyniki osiągnięte przez model wykorzystujący wyłącznie nagłówki wiadomości zamiast pełnej ich treści. Dla tych instrumentów widoczne jest pogorszenie się wyników (w porównaniu z pełnym modelem LSTM-FENE). Większość mierników świadczy o wyższości modelu bazowego. Jedynie prognozy dla złota i srebra (prognozy stóp zwrotu) oraz Dogecoin i srebra (prognozy poziomu cen) są lepsze niż przy użyciu modelu bazowego. W tabelach z wynikami modelu opartego o nagłówki zostały powtórzone wyniki dla modelu wykorzystującego pełne treści wiadomości z tabel 5.10 i 5.12 w celu łatwiejszego porównania rezultatów.

Wyniki otrzymane za pomocą testu Diebolda-Mariano dla RMSE (tabele 5.38 i 5.39) wskazują na statystycznie istotne pogorszenie jakości prognoz dla Ethereum, Bit-

Tabela 5.38. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu dla badanych metali szlachetnych i kryptowalut przy wykorzystaniu wyłącznie nagłówków wiadomości w module FENE - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Instrument	Wynik testu DM	Poziom istotności
Srebro	**-2.42**	**.015**
Złoto	-1.09	.276
Dogecoin	1.86	.063
Litecoin	**3.46**	**6.98E-04**
Cardano	**6.28**	**4.36E-10**
Bitcoin	**6.31**	**3.12E-10**
Ethereum	**6.29**	**3.94E-10**

Źródło: Obliczenia własne.

Tabela 5.39. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen dla badanych metali szlachetnych i kryptowalut przy wykorzystaniu wyłącznie nagłówków wiadomości w module FENE - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Instrument	Wynik testu DM	Poziom istotności
Dogecoin	**-3.34**	**9.57E-04**
Srebro	-1.84	.066
Litecoin	0.88	.380
Złoto	1.40	.162
Ethereum	**4.81**	**1.64E-06**
Bitcoin	**7.50**	**4.67E-13**
Cardano	**8.35**	**1.51E-16**

Źródło: Obliczenia własne.

coina i Cardano. Wyniki prognoz stóp zwrotu są istotnie lepsze dla srebra, a prognozy poziomu cen jedynie dla kryptowaluty Dogecoin.

Model LSTM-FENE osiąga lepsze wyniki od modeli bazowych nawet przy użyciu jedynie nagłówków wiadomości dla badanych spółek, jednak poprawa wyników nie jest tak znacząca, jak w przypadku pełnej treści wiadomości. Świadczy to o pozyskiwaniu przez model dodatkowych użytecznych informacji, które zawiera pełny tekst publikacji. Wyniki otrzymane dla metali szlachetnych i kryptowalut nie pozwalają stwierdzić wyższości tego modelu, jeśli nie są dostępne pełne treści wiadomości na temat tych instrumentów.

Powyższe wyniki potwierdzają postawioną we wstępie hipotezę: *wykorzystanie treści całej wiadomości w analizie pozwala na osiągnięcie lepszych wyników prognozowania niż w przypadku wykorzystania wyłącznie treści nagłówków.* Hipoteza ta jest potwierdzona dla prawie wszystkich badanych spółek. Efekt ten jest jeszcze bardziej widoczny dla metali szlachetnych i kryptowalut, gdzie wyniki modelu opartego o nagłówki uległy pogorszeniu względem modeli bazowych. Nagłówki składają się z kilku słów i bardzo

pobieżnego podsumowania treści wiadomości. Pełne wiadomości mogą nieść znacznie więcej kontekstu i nowych informacji.

5.7. Model LSTM bez wiadomości finansowych

Aby potwierdzić, że włączanie informacji zawartych w treściach wiadomości finansowych (lub samych nagłówków) jest znaczące, wytrenowany został model o identycznej architekturze, jednak bez modułu FENE. W ten sposób możliwe było modelowanie stóp zwrotu i poziomów cen przy pomocy sieci neuronowej otrzymującej identyczne informacje, co bazowe modele ARMA-GARCH.

Tabela 5.40. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych spółek indeksu S&P500 - porównanie pomiędzy siecią LSTM bez modułu FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym. Dla wygody otrzymane wyniki zostały zestawione wraz z wynikami z tabeli 5.4.

	Brak wiadomości		Pełna treść wiadomości	
Spółka	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
TSLA	-15.04	-3.17	-25.42	-3.54
BA	-9.35	-11.07	-10.16	-12.68
NKE	-8.86	-7.66	-12.85	-11.99
KO	-8.13	-7.86	-10.46	-10.33
CRM	-6.69	-7.99	-7.83	-11.55
JPM	-5.10	-6.71	-5.40	-7.65
GM	-4.16	-3.70	-5.55	-5.62
MSFT	-3.65	-6.01	-5.01	-6.21
GS	-3.57	-2.57	-2.27	-3.42
NVDA	-2.64	-8.56	-1.35	-9.51
F	-1.97	-3.96	-1.25	-2.44
C	-0.95	-2.76	1.58	-2.23
INTC	0.30	0.89	5.02	1.37
MS	1.43	-2.26	4.91	-3.84
VZ	4.06	-2.03	7.69	-3.27

Źródło: Obliczenia własne.

W tabeli 5.40 zebrane zostały zmiany błędu prognozy stopy zwrotu wybranych spółek indeksu S&P 500 dla modelu o architekturze LSTM w porównaniu z bazowymi modelami ARMA-GARCH. W tabeli 5.41 znajdują się adekwatne wyniki dla prognoz poziomów cen.

Otrzymane wyniki sugerują, że model ten ma pewną przewagę nad modelami bazowymi, jednak jest ona znacznie mniejsza niż w przypadku pełnego modelu LSTM-FENE lub nawet modelu LSTM-FENE wykorzystującego tylko nagłówki zamiast pełnej treści

Tabela 5.41. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych spółek indeksu S&P500 - porównanie pomiędzy siecią LSTM bez modułu FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym. Dla wygody otrzymane wyniki zostały zestawione wraz z wynikami z tabeli 5.6.

Spółka	Brak wiadomości			Pełna treść wiadomości		
	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]
TSLA	-8.30	-4.94	0.55	-11.80	-12.04	-7.00
KO	-7.13	-7.19	-7.85	-9.83	-9.90	-10.33
F	-4.19	-6.76	-3.94	-4.65	-6.68	-2.45
BA	-2.50	-3.57	-11.06	0.81	-2.16	-12.68
MSFT	-2.39	-4.76	-6.00	-0.85	-2.03	-6.23
GS	-2.25	-1.51	-2.56	-2.07	-3.17	-3.41
NKE	-2.09	-1.02	-7.65	-6.87	-5.57	-11.98
JPM	-1.97	-3.51	-6.71	0.12	-2.25	-7.65
GM	-1.81	-1.84	-3.70	-6.37	-6.86	-5.63
CRM	-0.95	-1.88	-7.99	-0.80	-4.74	-11.59
VZ	-0.55	-6.12	-2.02	8.79	-3.96	-3.26
C	0.44	-1.35	-2.75	2.15	-1.29	-2.20
NVDA	1.89	2.54	2.09	0.13	1.59	-0.74
INTC	1.99	3.75	0.90	5.90	1.50	1.39
MS	5.47	0.62	-2.24	8.50	-1.42	-3.83

Źródło: Obliczenia własne.

wiadomości. Mimo, że model LSTM bez modułu FENE pozwala osiągać lepsze prognozy dla części spółek (w porównaniu z modelami bazowymi), to otrzymane wyniki nie wykazują tak dużej poprawy, jak w przypadku modeli LSTM-FENE biorących pod uwagę dodatkowe informacje. Oznacza to, że sieci neuronowe są przynajmniej równie dobre w modelowaniu finansowych szeregów czasowych, co modele ARMA-GARCH. W tabelach z wynikami modelu bez wiadomości finansowych zostały powtórzone wyniki dla modelu wykorzystującego pełne treści wiadomości z tabel 5.4 i 5.6 w celu ułatwienia porównania rezultatów.

Wyniki testu Diebolda-Mariano dla RMSE (tabela 5.42 dla prognoz stóp zwrotu oraz tabela 5.43 dla prognoz poziomów cen) wskazują, że model nie uwzględniający treści wiadomości jest istotnie lepszy od modeli bazowych jedynie dla spółek TSLA i KO. W przypadku prognoz stóp zwrotu istotnie lepsze są również wyniki dla BA oraz dla NKE. Dla prognoz poziomów cen istotnie gorsze są wyniki otrzymywane dla spółki MS.

Analiza procentowej zmiany błędów prognoz modelu LSTM względem bazowych modeli ARMA-GARCH dla metali szlachetnych i kryptowalut znajduje się w tabe-

Tabela 5.42. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu dla badanych spółek indeksu S&P500 - porównanie pomiędzy siecią LSTM bez modułu FENE a modelem ARMA-GARCH.

Spółka	Wynik testu DM	Poziom istotności
TSLA	**-3.46**	**6.98E-04**
BA	**-2.39**	**.017**
NKE	**-2.73**	**.007**
KO	**-2.39**	**.017**
CRM	-1.90	.058
JPM	-1.77	.077
GM	-1.50	.134
MSFT	-1.47	.141
GS	-1.37	.172
NVDA	-1.22	.223
F	-1.11	.266
C	-0.68	.498
INTC	0.39	.694
MS	0.86	.390
VZ	1.75	.081

Źródło: Obliczenia własne.

Tabela 5.43. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen dla badanych spółek indeksu S&P500 - porównanie pomiędzy siecią LSTM bez modułu FENE a modelem ARMA-GARCH.

Spółka	Wynik testu DM	Poziom istotności
TSLA	**-2.86**	**.005**
KO	**-2.78**	**.006**
F	-1.88	.061
BA	-1.49	.136
MSFT	-1.30	.195
GS	-1.31	.192
NKE	-1.31	.192
JPM	-1.20	.230
GM	-1.28	.199
CRM	-0.87	.386
VZ	-0.55	.582
C	0.43	.670
NVDA	1.10	.271
INTC	1.26	.208
MS	**2.21**	**.027**

Źródło: Obliczenia własne.

li 5.44 dla prognoz stóp zwrotu i w tabeli 5.45 dla prognoz poziomów cen. Wyniki świadczą o minimalnej poprawie błędu RMSE dla kryptowalut i pogorszeniu wszystkich innych miar. Dla metali szlachetnych widoczne jest jedynie pogorszenie błędów. Są to wyniki znacznie gorsze niż dla pełnego modelu LSTM-FENE i porównywalnie złe

Tabela 5.44. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy siecią LSTM bez modułu FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym. Dla wygody otrzymane wyniki zostały zestawione wraz z wynikami z tabeli 5.10.

	Brak wiadomości		Pełna treść wiadomości	
Instrument	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
Dogecoin	-0.60	0.72	-0.63	0.47
Litecoin	-0.29	0.45	-0.43	0.24
Ethereum	-0.26	0.78	-0.30	0.53
Cardano	-0.21	0.92	-0.29	0.33
Bitcoin	-0.17	1.06	-0.23	0.54
Złoto	0.01	0.21	-0.16	-0.03
Srebro	0.06	0.26	-0.03	0.07

Źródło: Obliczenia własne.

Tabela 5.45. Procentowa zmiana błędów prognoz RMSE, MAE i MAPE dla prognoz poziomu cen dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy siecią LSTM bez modułu FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym. Dla wygody otrzymane wyniki zostały zestawione wraz z wynikami z tabeli 5.12.

	Brak wiadomości			Pełna treść wiadomości		
Instrument	ΔRMSE	ΔMAE	ΔMAPE	ΔRMSE	ΔMAE	ΔMAPE
Dogecoin	-0.64	0.64	0.71	-0.63	0.47	0.48
Litecoin	-0.36	0.42	0.43	-0.48	0.23	0.24
Ethereum	-0.31	0.74	0.76	-0.34	0.52	0.53
Bitcoin	-0.21	1.01	1.05	-0.26	0.51	0.54
Cardano	-0.14	0.84	0.90	-0.24	0.31	0.33
Złoto	0.01	0.21	0.21	-0.16	-0.03	-0.03
Srebro	0.05	0.25	0.26	-0.04	0.07	0.08

Źródło: Obliczenia własne.

dla modeli opartego wyłącznie o nagłówki. W tabelach z wynikami modelu bez wiadomości finansowych zostały powtórzone wyniki dla modelu wykorzystującego pełne treści wiadomości z tabel 5.10 i 5.12 w celu łatwiejszego porównania rezultatów).

Wyniki testu Diebolda-Mariano dla RMSE (tabele 5.46 i 5.47 dla odpowiednio: prognoz stóp zwrotu i prognoz poziomów cen) porównującego model LSTM nie wykorzystującego wiadomości finansowych z bazowymi modelami ARMA-GARCH dla metali szlachetnych i kryptowalut wskazują, że otrzymane wyniki są istotne statycznie wyłącznie dla kryptowaluty Dogecoin.

Model LSTM w niektórych przypadkach (głównie podczas modelowania spółek giełdowych) sprawdza się lepiej niż klasyczne modele ekonometryczne, jednak poprawa jest nieznaczna. Brak dodatkowych informacji płynących z wiadomości finansowych nie

Tabela 5.46. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy siecią LSTM bez modułu FENE a modelem ARMA-GARCH.

Instrument	Wynik testu DM	Poziom istotności
Dogecoin	**-2.67**	**.008**
Litecoin	-1.59	.111
Ethereum	-1.67	.095
Cardano	-1.27	.204
Bitcoin	-1.05	.295
Złoto	0.26	.798
Srebro	0.58	.563

Źródło: Obliczenia własne.

Tabela 5.47. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy siecią LSTM bez modułu FENE a modelem ARMA-GARCH.

Instrument	Wynik testu DM	Poziom istotności
Dogecoin	**-2.75**	**.006**
Litecoin	-1.94	.052
Ethereum	-1.90	.058
Bitcoin	-1.35	.177
Cardano	-1.01	.312
Złoto	0.31	.757
Srebro	0.57	.569

Źródło: Obliczenia własne.

pozwala wykorzystać pełnego potencjału sieci neuronowych. Powyższe wyniki częściowo potwierdzają postawioną we wstępie hipotezę: *modele oparte na głębokich sieciach neuronowych z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z modelami wykorzystującymi klasyczne metody sztucznej inteligencji.* Hipoteza ta jest potwierdzona dla prawie wszystkich badanych spółek. Dla kryptowalut oraz metali szlachetnych różnice w jakości modeli są niezauważalne (poza kryptowalutą Dogecoin).

5.8. Podsumowanie wyników badań

Na podstawie przeprowadzonych badań możliwe jest wyciągnięcie szeregu konkluzji:

— **Otrzymane wyniki świadczą o pozytywnym wpływie wykorzystania treści wiadomości pochodzących z portali finansowych na prognozowanie instrumentów finansowych przy dużej (5-minutowej) rozdzielczości czasowej.** Błędy prognozy stóp zwrotu i poziomów cen są mniejsze zarówno dla spółek

giełdowych, jak i (w mniejszym stopniu) metali szlachetnych oraz kryptowalut. Efekt ten jest szczególnie widoczny w okresach bezpośrednio po pojawieniu się nowych publikacji na temat tych instrumentów finansowych. Test Diebolda-Mariano dla RMSE dowodzi, że prognozy punktowe otrzymywane z autorskiego modelu LSTM-FENE są istotnie lepsze niż z klasycznych modeli ARMA-GARCH.

— **Wpływ wiadomości na jakość modeli różni się w zależności od niektórych czynników.** Największa poprawa jakości modeli widoczna jest w przypadku modelowania spółek giełdowych, zaś nieco mniejsza dla metali szlachetnych i kryptowalut. W przypadku spółek giełdowych mniejsza jest dzienna liczba publikowanych wiadomości. Dla metali szlachetnych i kryptowalut duża ilość publikacji może wręcz wprowadzać niepotrzebny szum.

— **Poprawa jakości prognoz stóp zwrotu i poziomów cen jest zależna od źródła, z którego pochodzą wiadomości.** Portale o większej renomie i popularności wywierają większy wpływ na notowania, co jest wykorzystywane przez model LSTM-FENE. Dla różnych źródeł i różnych instrumentów finansowych nie ma jednak korelacji między dzienną liczbą wiadomości (ani ich długością), a poprawą jakości prognoz.

— **Dobór hiperparametrów miał znaczenie w jakości otrzymywanych modeli.** Hiperparametry opisywanego modelu LSTM-FENE zostały dobrane poprzez częściowy *grid search*, ograniczony długim czasem treningu dużych modeli językowych. Dla wszystkich analizowanych typów instrumentów finansowych mały model BERT w części tekstowej modelu okazał się być najbardziej skuteczny. Różnice pojawiły się dopiero w doborze hiperparametrów części przetwarzającej finansowe szeregi czasowe.

— **Użycie wyłącznie nagłówków wiadomości również pozwala na podniesienie jakości prognoz względem klasycznych modeli ekonometrycznych, jednak dopiero wykorzystanie pełnych treści wiadomości umożliwia modelowi LSTM-FENE na zwiększenie potencjału.** Modele oparte tylko na nagłówkach otrzymują mniej dodatkowych informacji koniecznych do stworzenia skutecznych prognoz niż modele korzystające z pełnej treści wiadomości.

Badania przeprowadzone na modelach w pełni opartych na architekturze sieci neuronowych, tj. sieci LSTM przetwarzającej historyczne szeregi czasowe (stóp zwrotu i poziomów cen) połączonej z autorskim modułem FENE, pozwoliły dowieść większa

skuteczność tej metody w porównaniu z klasycznymi modelami ekonometrycznymi czy sieciami neuronowymi bez dostępu do informacji zawartych w wiadomościach finansowych. **Pierwsze cztery hipotezy postawione we wstępie niniejszej pracy zostały potwierdzone.**

Rozdział 6

Wykorzystanie modeli hybrydowych ARMA-GARCH-FENE oraz analiza ich efektywności

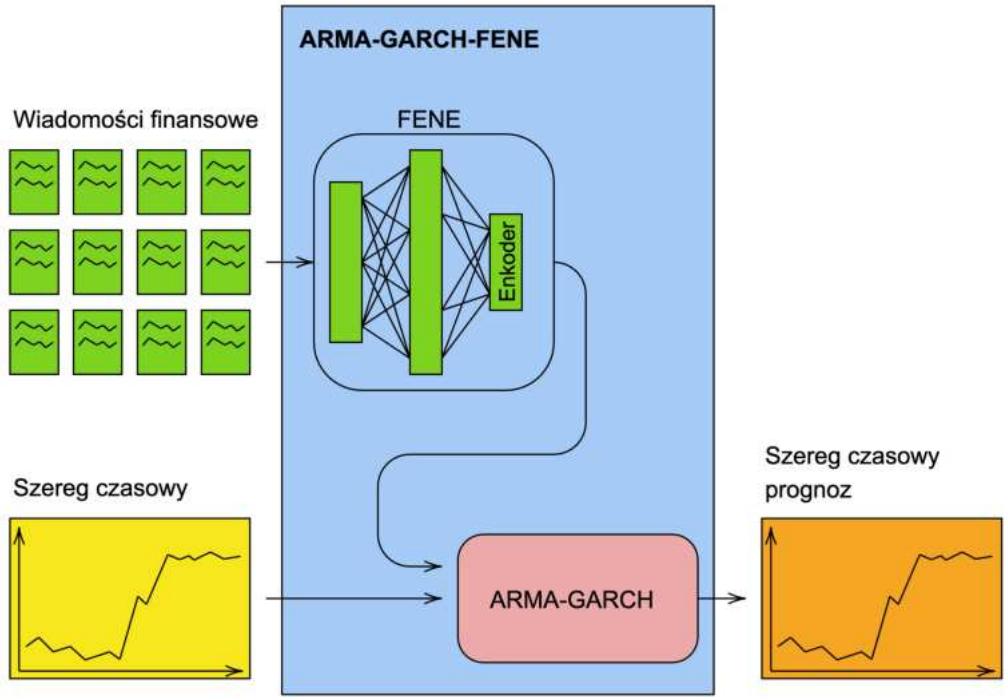
Wyniki osiągnięte dzięki zastosowaniu sieci neuronowych mogą być wykorzystane do stworzenia hybrydowych modeli typu ARMA-GARCH-FENE - klasycznych modeli ARMA-GARCH z modułem językowym FENE, służącym do analizy wiadomości finansowych. Zolfaghari M. i Gholami S. badali podejście polegające na wykorzystaniu prognoz z modeli ARMA-GARCH jako danych wejściowych różnego typu innych modeli [204]. Zbadali między innymi efekty wykorzystania różnych specyfikacji modeli ARMA-GARCH w połączeniu ze wskaźnikami analizy technicznej, które następnie poddano transformacji falkowej i podano na wejście rekurencyjnych sieci neuronowych.

W niniejszej pracy przyjęto założenie odwrotne - moduł FENE jest wykorzystywany do przetworzenia tekstów wiadomości dotyczących modelowanych instrumentów finansowych. Wyjście z tej części sieci jest następnie wykorzystywane jako wielowymiarowy szereg czasowy danych egzogenicznych dla modelu ARMA-GARCH. Podejście to pozwala wykorzystać dodatkowe informacje zawarte w tekstach przez klasyczne modele ekonometryczne. W rozdziale 6.1 przedstawiona została architektura hybrydowych modeli, zaś w rozdziale 6.3 wyniki otrzymane przy ich użyciu.

6.1. Propozycja architektury hybrydowych modeli

Klasyczne modele ARMA-GARCH mogą wykorzystywać dane egzogeniczne, których celem jest dostarczenie dodatkowych informacji, aby poprawić skuteczność prognoz. Uwzględnienie informacji zawartych w wiadomościach tekstowych wymaga przełożenia ich na wielkości liczbowe, posiadające wartość dla każdej chwili t_i z przypisaną modelowaną wartością stopy zwrotu.

Zautomatyzowane wyciąganie wniosków z dużej ilości takich wiadomości tekstowych jest możliwe przez wykorzystanie modułu FENE o odpowiedniej wielkości enkodera



Rysunek 6.1. Proponowana architektura hybrydowego modelu będącego połączeniem klasycznego modelu ARMA-GARCH z siecią neuronową przetwarzającą treść wiadomości. Wiadomości tekstowe analizowane są przy pomocy wytrenowanego wcześniej modułu FENE, a następnie w postaci wektora liczb trafiają do modelu ARMA-GARCH jako dane egzogeniczne. Źródło: Opracowanie własne.

$d_{encodefr}$. W ten sposób każdej chwili czasowej t_i przypisany zostaje wektor pojawiający się na wyjściu enkodera obliczony na podstawie treści wiadomości. Jeżeli nie było w tym czasie żadnej wiadomości, wektor obliczany jest na podstawie pustego ciągu znaków.

Rysunek 6.1 przedstawia schemat takiego modelu. Dla każdej chwili t_i dane wejściowe to szereg logarytmicznych stóp zwrotu oraz wektor wyjściowy enkodera, zatem model ARMA-GARCH otrzymuje $d_{encoder} + 1$ szeregów czasowych.

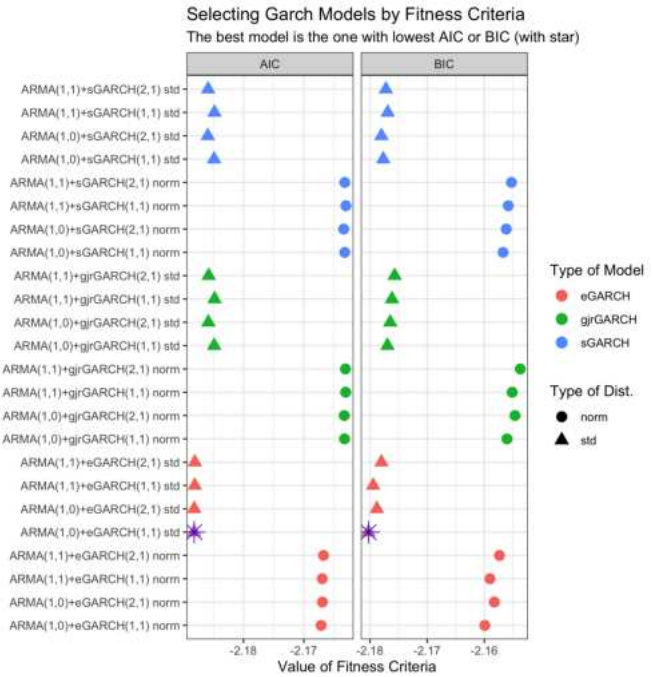
Wykorzystywany model to ARMA-GARCH wraz z modułem FENE o hiperparametrach zapewniajacych najlepsze osiągi (ich dobór został opisany w rozdziale 5.5.1). Moduł FENE wykorzystany w modelu hybrydowym zostaje "wycięty" już **po** procesie treningu pełnego modelu LSTM-FENE, aby był już wstępnie wytrenowany do swojego zadania.

Wielkość enkodera $d_{encoder}$ ma bezpośredni wpływ na architekturę modelu hybrydowego ze względu na ilość szeregów czasowych stanowiących dane wejściowe. Wobec tego, analiza wyników została przeprowadzona dla trzech rozmiarów enkodera: $d_{encoder} \in \{4, 8, 16\}$.

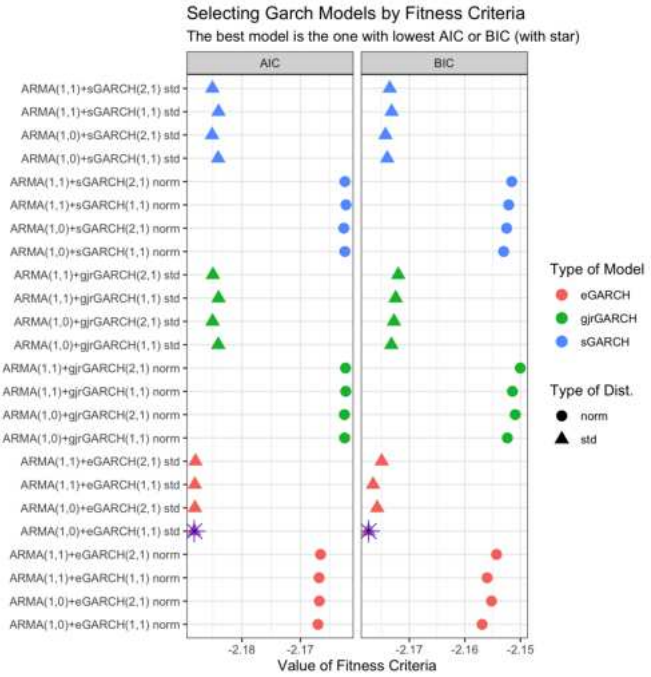
6.2. Estymacja i wybór specyfikacji hybrydowych modeli

Dla każdej z wybranych do badań spółek akcyjnych został estymowany hybrydowy model ARMA-GARCH-FENE, tj. model ARMA-GARCH na podstawie historycznych szeregów logarytmicznych stóp zwrotu oraz wektorów pochodzących z modułu FENE. Każdy z modeli powstał w 3 wersjach bazujących na trzech rozmiarach enkodera $d_{encoder} \in \{4, 8, 16\}$. Prognozy stóp zwrotu oraz poziomu cen wyznaczane były w schemacie rolowanym (patrz rozdział 4.5.2). We wszystkich przypadkach zostały wyliczone jednopunktowe prognozy logarytmicznych stóp zwrotu w horyzoncie $h = 1$.

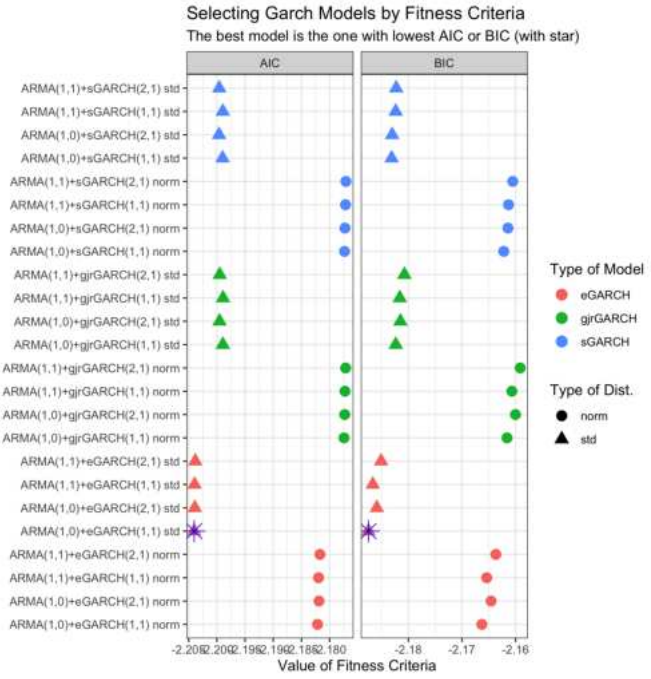
Tak jak w przypadku modeli bazowych, specyfikacja i rzędy opóźnień hybrydowych modeli ARMA-GARCH-FENE zostały ustalone dla pierwszego okna estymacji w schemacie rolowanym. W tym celu wykorzystane są kryteria AIC oraz BIC (patrz rozdział 2.1.1). Zostały one obliczone dla każdego badanego instrumentu finansowego dla trzech różnych wielkości enkodera w module FENE. Tak jak w przypadku modeli bazowych, po ustaleniu najlepszej specyfikacji hybrydowego modelu na pierwszym oknie czasowym (o długości 3 miesięcy, tj. 4914 obserwacji), była ona następnie utrzymana we wszystkich kolejnych oknach czasowych dla danego instrumentu finansowego.



Rysunek 6.2. Wybór najlepszej specyfikacji modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) na podstawie kryteriów AIC oraz BIC dla spółki Tesla Inc. (TSLA). Fioletowa gwiazdka oznacza, który typ modelu osiąga najlepszą wartość każdego z kryteriów. Różne specyfikacje modelu GARCH oznaczone są różnymi kolorami, zaś różne rozkłady kształtami znaczników. Źródło: Obliczenia własne, wizualizacja na podstawie [150].



Rysunek 6.3. Wybór najlepszej specyfikacji modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) na podstawie kryteriów AIC oraz BIC dla spółki Tesla Inc. (TSLA). Fioletowa gwiazdka oznacza, który typ modelu osiąga najlepszą wartość każdego z kryteriów. Różne specyfikacje modelu GARCH oznaczone są różnymi kolorami, zaś różne rozkłady kształtami znaczników. Źródło: Obliczenia własne, wizualizacja na podstawie [150].



Rysunek 6.4. Wybór najlepszej specyfikacji modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) na podstawie kryteriów AIC oraz BIC dla spółki Tesla Inc. (TSLA). Fioletowa gwiazdka oznacza, który typ modelu osiąga najlepszą wartość każdego z kryteriów. Różne specyfikacje modelu GARCH oznaczone są różnymi kolorami, zaś różne rozkłady kształtami znaczników. Źródło: Obliczenia własne, wizualizacja na podstawie [150].

Rysunki 6.2, 6.3 oraz 6.4 przedstawiają podsumowanie analizy rzędów opóźnień oraz specyfikacji modeli dla spółki Tesla Inc. (TSLA) z długością enkodera w module FENE równą odpowiednio $d_{encoder} = 4$, $d_{encoder} = 8$ i $d_{encoder} = 16$. Im niższa wartość kryteriów AIC oraz BIC tym lepsze dopasowanie modelu do danych. Dla każdego z kryteriów najlepszy model został oznaczony gwiazdką. Szczegółowe wyniki takiej analizy dla każdego z badanych instrumentów zostały zebrane w aneksie (rozdział A.3).

Tabela 6.1. Specyfikacje, warunkowe rozkłady prawdopodobieństwa oraz optymalne rzędy opóźnień ustalone na podstawie kryterium informacyjnego AIC dla modeli ARMA-GARCH-FENE ($d_{encoder} = 4$) dla badanych instrumentów finansowych.

	ARMA		GARCH			
Instrument	p	q	q	p	specyfikacja	rozkład
TSLA	1	0	1	1	EGARCH	t-Studenta
NKE	1	1	1	1	**GARCH**	t-Studenta
KO	1	1	2	1	EGARCH	t-Studenta
BA	1	0	1	1	**GARCH**	t-Studenta
CRM	1	1	2	1	EGARCH	t-Studenta
GM	1	0	1	1	**GJR-GARCH**	t-Studenta
JPM	1	0	2	1	EGARCH	t-Studenta
MSFT	1	0	2	1	EGARCH	t-Studenta
GS	1	1	1	1	**GJR-GARCH**	t-Studenta
NVDA	1	0	1	1	**GJR-GARCH**	t-Studenta
F	1	0	1	1	**GJR-GARCH**	t-Studenta
C	1	0	2	1	EGARCH	t-Studenta
MS	1	1	2	1	EGARCH	t-Studenta
INTC	1	1	2	1	**GARCH**	t-Studenta
VZ	1	0	2	1	EGARCH	t-Studenta
Złoto	1	1	2	1	EGARCH	t-Studenta
Srebro	1	0	2	1	EGARCH	t-Studenta
Bitcoin	1	1	2	1	EGARCH	t-Studenta
Cardano	1	1	2	1	EGARCH	t-Studenta
Ethereum	1	1	2	1	EGARCH	t-Studenta
Litecoin	1	1	2	1	EGARCH	t-Studenta
Dogecoin	1	1	2	1	EGARCH	t-Studenta

Źródło: Obliczenia własne.

Podsumowanie rzędów opóźnień, specyfikacji i rozkładów prawdopodobieństwa otrzymanych na podstawie kryterium informacyjnego AIC dla każdego instrumentu znajduje się w tabeli 6.1 przy założeniu wielkości enkodera $d_{encoder} = 4$. Analogiczne wyniki dla modeli z wielkościami enkodera równymi $d_{encoder} = 8$ oraz $d_{encoder} = 16$ znajdują się w tabeli 6.2 i tabeli 6.3.

Do dalszych analiz modele były wybierane na podstawie kryterium AIC. Tabele podsumowujące wartości kryterium BIC dla modeli hybrydowych znajdują się w

Tabela 6.2. Specyfikacje, warunkowe rozkłady prawdopodobieństwa oraz optymalne rzędy opóźnień ustalone na podstawie kryterium informacyjnego AIC dla modeli ARMA-GARCH-FENE ($d_{encoder} = 8$) dla badanych instrumentów finansowych.

	ARMA		GARCH			
Instrument	p	q	q	p	specyfikacja	rozkład
TSLA	1	0	1	1	EGARCH	t-Studenta
NKE	1	1	1	1	**GARCH**	t-Studenta
KO	1	1	2	1	EGARCH	t-Studenta
BA	1	0	2	1	EGARCH	t-Studenta
CRM	1	0	2	1	EGARCH	t-Studenta
GM	1	0	1	1	**GJR-GARCH**	t-Studenta
JPM	1	0	2	1	EGARCH	t-Studenta
MSFT	1	0	2	1	EGARCH	t-Studenta
GS	1	1	2	1	EGARCH	t-Studenta
NVDA	1	0	1	1	**GJR-GARCH**	t-Studenta
F	1	0	1	1	**GJR-GARCH**	t-Studenta
C	1	0	2	1	EGARCH	t-Studenta
MS	1	1	2	1	EGARCH	t-Studenta
INTC	1	0	1	1	**GARCH**	t-Studenta
VZ	1	0	2	1	EGARCH	t-Studenta
Złoto	1	1	2	1	EGARCH	t-Studenta
Srebro	1	0	2	1	EGARCH	t-Studenta
Bitcoin	1	1	2	1	EGARCH	t-Studenta
Cardano	1	1	2	1	EGARCH	t-Studenta
Ethereum	1	1	2	1	EGARCH	t-Studenta
Litecoin	1	1	2	1	EGARCH	t-Studenta
Dogecoin	1	1	2	1	EGARCH	t-Studenta

Źródło: Obliczenia własne.

aneksie (patrz rozdział A.3.4). Również w tym przypadku analizy zostały przeprowadzone dla modeli ARMA-GARCH-FENE o trzech wielkościach enkodera: $d_{encoder} = 4$, $d_{encoder} = 8$ i $d_{encoder} = 16$.

Optymalny wybór rzędów opóźnień oraz specyfikacji modelu ARMA-GARCH-FENE nie zawsze jest identyczny, co w przypadku bazowego modelu ARMA-GARCH dla tego samego instrumentu finansowego. Różnice zdarzają się również w przypadku, gdy zmienia się wielkość enkodera w module FENE. Dla większości instrumentów najlepsze wyniki osiągane są przez zastosowanie specyfikacji EGARCH. We wszystkich przypadkach przewagę osiągają modele wykorzystujące rozkład t-Studenta.

Tabela 6.3. Specyfikacje, warunkowe rozkłady prawdopodobieństwa oraz optymalne rzędy opóźnień ustalone na podstawie kryterium informacyjnego AIC dla modeli ARMA-GARCH-FENE ($d_{encoder} = 16$) dla badanych instrumentów finansowych.

	ARMA		GARCH			
Instrument	p	q	q	p	specyfikacja	rozkład
TSLA	1	0	1	1	EGARCH	t-Studenta
NKE	1	1	1	1	**GARCH**	t-Studenta
KO	1	1	2	1	EGARCH	t-Studenta
BA	1	0	1	1	**GARCH**	t-Studenta
CRM	1	0	2	1	EGARCH	t-Studenta
GM	1	0	1	1	**GJR-GARCH**	t-Studenta
JPM	1	0	2	1	EGARCH	t-Studenta
MSFT	1	0	2	1	EGARCH	t-Studenta
GS	1	0	1	1	**GJR-GARCH**	t-Studenta
NVDA	1	0	1	1	**GJR-GARCH**	t-Studenta
F	1	0	1	1	**GJR-GARCH**	t-Studenta
C	1	0	2	1	EGARCH	t-Studenta
MS	1	1	2	1	**GJR-GARCH**	t-Studenta
INTC	1	0	1	1	**GARCH**	t-Studenta
VZ	1	0	2	1	EGARCH	t-Studenta
Złoto	1	1	2	1	EGARCH	t-Studenta
Srebro	1	1	2	1	EGARCH	t-Studenta
Bitcoin	1	1	2	1	EGARCH	t-Studenta
Cardano	1	1	2	1	EGARCH	t-Studenta
Ethereum	1	1	2	1	EGARCH	t-Studenta
Litecoin	1	0	2	1	EGARCH	t-Studenta
Dogecoin	1	1	2	1	EGARCH	t-Studenta

Źródło: Obliczenia własne.

6.3. Analiza efektywności hybrydowych modeli

6.3.1. Porównanie modeli hybrydowych z modelami bazowymi

Mierniki wykorzystywane dla modeli hybrydowych zdefiniowane są w sposób analogiczny do mierników modeli LSTM-FENE. Obliczane są procentowe zmiany błędu prognozy stóp zwrotu względem bazowych modeli ARMA-GARCH dane wzorami 5.1, 5.2, 5.3. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Jakość hybrydowych modeli została oceniona poprzez obliczenie błędów prognozy stóp zwrotu RMSE oraz MAE, a następnie porównanie ich z błędami prognozy bazowych modeli ARMA-GARCH - tak jak w przypadku modelu LSTM-FENE opisywanego w rozdziale 5. W tabeli 6.4 przedstawione zostały procentowe zmiany błędów

Tabela 6.4. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Spółka	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
MSFT	-1.92	-2.35	-1.78	-2.23	-1.71	-2.10
GM	-0.73	-0.76	-0.76	-0.91	-0.87	-0.98
CRM	-0.42	-0.44	-0.47	-0.48	-0.47	-0.55
INTC	-0.40	-0.66	-0.39	-0.66	-0.51	-0.75
TSLA	-0.34	-3.46	-0.27	-2.74	-0.26	-2.57
NVDA	-0.17	-0.29	-0.14	-0.39	-0.20	-0.33
BA	-0.15	-0.42	-0.03	-0.23	-0.09	-0.21
NKE	0.04	0.61	-0.03	0.37	-0.02	0.31
KO	0.09	0.04	0.18	0.16	0.15	0.17
GS	0.54	0.72	0.44	0.76	0.65	0.89
VZ	0.67	1.02	0.70	1.05	0.64	1.04
JPM	0.74	1.08	0.68	1.03	0.62	0.87
F	1.05	2.10	1.02	2.03	1.00	2.05
MS	1.11	1.41	0.56	0.67	0.79	1.08
C	1.61	2.13	1.66	2.36	1.72	2.36

Źródło: Obliczenia własne.

względem modeli bazowych. Ujemna wartość oznacza poprawę jakości prognoz stóp zwrotu na korzyść modelu hybrydowego.

Poprawa obu błędów prognoz stóp zwrotu widoczna jest tylko w przypadku 7 spółek dla wszystkich trzech rozmiarów enkodera $d_{encoder}$. Dla spółki Nike Inc. (NKE) poprawa widoczna jest tylko w przypadku błędu RMSE dla dwóch mniejszych modeli $d_{encoder} = 8$ oraz $d_{encoder} = 4$. We wszystkich jednak przypadkach poprawa prognozy jest znacznie mniejsza niż dla modelu LSTM-FENE, opisanego w rozdziale 5.

W tabeli 6.5 znajduje się porównanie modeli hybrydowych ARMA-GARCH-FENE z bazowymi modelami ARMA-GARCH z wykorzystaniem błędów prognoz poziomów cen. Poprawa jest zauważalna wyłącznie dla 6 spółek. Dla spółek NVDA oraz NKE poprawa jest widoczna tylko dla niektórych rozmiarów enkodera.

Prognozy hybrydowego modelu zostały zbadane testem Diebolda-Mariano przez porównanie ich z bazowymi modelami ARMA-GARCH. Wyniki przedstawia tabela 6.6.

Wyniki testu świadczą o statystycznie znaczącej poprawie prognozy stóp zwrotu dla czterech ze spółek: MSFT, GM, CRM oraz INTC. W pozostałych przypadkach jakość prognoz jest lepsza (jednak niewystarczająco, aby miało to znaczenie statystyczne) lub wręcz gorsza (istotnie gorsza dla spółek GS, VS, JPM, F, MS, C).

Tabela 6.5. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz poziomu cen dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
Spółka	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
MSFT	-1.92	-2.30	-1.79	-2.17	-1.71	-2.04
GM	-0.68	-0.77	-0.70	-0.91	-0.81	-0.99
INTC	-0.44	-0.70	-0.43	-0.68	-0.51	-0.77
BA	-0.42	-0.62	-0.31	-0.43	-0.36	-0.40
CRM	-0.33	-0.38	-0.36	-0.40	-0.37	-0.47
TSLA	-0.27	-3.91	-0.28	-3.52	-0.26	-3.33
NVDA	-0.04	-0.20	0.01	-0.27	-0.07	-0.22
NKE	0.06	0.66	-0.01	0.41	-0.01	0.34
KO	0.09	0.04	0.18	0.16	0.15	0.16
GS	0.47	0.72	0.49	0.79	0.57	0.89
VZ	0.59	0.99	0.63	1.04	0.58	1.02
JPM	0.78	1.11	0.72	1.05	0.67	0.89
F	1.15	2.16	1.10	2.08	1.09	2.10
MS	1.17	1.45	0.57	0.66	0.84	1.11
C	1.56	2.11	1.64	2.36	1.69	2.35

Źródło: Obliczenia własne.

Tabela 6.6. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu przy wykorzystaniu obu metod dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH.

	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
Spółka	DM	p	DM	p	DM	p
MSFT	**-4.68**	**2.97E-06**	**-4.21**	**2.63E-05**	**-4.03**	**5.86E-05**
GM	**-2.42**	**.015**	**-3.10**	**.002**	**-2.79**	**.006**
CRM	**-1.97**	**.049**	**-2.12**	**.034**	**-2.14**	**.033**
INTC	-1.87	.062	-1.94	.052	**-2.15**	**.032**
TSLA	-1.69	.092	-1.51	.132	-1.42	.156
NVDA	-1.12	.262	-0.96	.338	-1.32	.188
BA	-0.90	.367	-0.28	.781	-0.85	.394
NKE	0.52	.601	-0.29	.770	-0.38	.702
KO	0.79	.432	1.23	.219	0.95	.343
GS	**2.32**	**.021**	**1.98**	**.048**	**2.55**	**.011**
VZ	**2.68**	**.007**	**2.66**	**.008**	**2.53**	**.012**
JPM	**2.64**	**.008**	**2.84**	**.005**	**2.56**	**.011**
F	**3.60**	**3.95E-04**	**3.61**	**3.74E-04**	**2.97**	**.004**
MS	**3.51**	**5.90E-04**	**2.28**	**.022**	**2.90**	**.005**
C	**4.55**	**5.53E-06**	**4.02**	**6.14E-05**	**4.99**	**6.07E-07**

Źródło: Obliczenia własne.

Wyniki testu Diebolda-Mariano dla RMSE przeprowadzonego dla prognoz poziomów cen zostały zebrane w tabeli 6.7. Zgodnie z wynikami testu, model hybrydowy jest lepszy w statystycznie istotny sposób dla tych samych czterech spółek: MSFT, GM, CRM, INTC, a także dla BA. Model jest gorszy w statystycznie istotny sposób dla wszystkich trzech wielkości enkodera w przypadku sześciu spółek: GS, VZ, JPM, F, MS oraz C.

Tabela 6.7. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen przy wykorzystaniu obu metod dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH.

	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
Spółka	DM	p	DM	p	DM	p
MSFT	**-6.13**	**1.05E-09**	**-6.15**	**9.71E-10**	**-6.08**	**1.26E-09**
GM	**-3.42**	**7.84E-04**	**-3.00**	**.004**	**-3.65**	**2.87E-04**
INTC	**-2.53**	**.012**	**-2.62**	**.009**	**-2.61**	**.009**
BA	**-2.17**	**.030**	**-2.11**	**.035**	**-2.17**	**.030**
CRM	-1.94	.052	**-2.03**	**.042**	**-1.99**	**.047**
TSLA	-1.82	.069	-1.87	.062	-1.78	.075
NVDA	-0.61	.544	0.30	.764	-0.88	.380
NKE	0.80	.425	-0.23	.822	-0.36	.719
KO	0.99	.322	1.52	.129	1.40	.162
GS	**2.83**	**.005**	**2.38**	**.017**	**2.79**	**.006**
VZ	**3.14**	**.002**	**3.11**	**.002**	**2.88**	**.005**
JPM	**3.62**	**3.52E-04**	**3.12**	**.002**	**3.44**	**7.41E-04**
F	**3.93**	**2.14E-04**	**4.52**	**6.28E-06**	**4.26**	**2.11E-05**
MS	**4.71**	**2.61E-06**	**2.68**	**.007**	**3.75**	**1.84E-04**
C	**5.09**	**3.80E-07**	**5.56**	**4.44E-08**	**5.80**	**1.99E-08**

Źródło: Obliczenia własne.

Dla metali szlachetnych i kryptowalut przeprowadzone zostały analogiczne obliczenia. Wyniki analiz dla prognoz stóp zwrotu przedstawia tabela 6.8, a dla poziomów cen tabela 6.9. Dla żadnego z tych instrumentów hybrydowe modele nie oferują wyników lepszych niż modele klasyczne. Również tutaj, zmiany względem modeli bazowych nie są duże.

Znaczna ilość dodatkowych informacji napływających w nieregularnych interwałach nie pozwala modelom na lepsze prognozowanie stóp zwrotu wyjątkowo zmiennych instrumentów. W celu zbadania tego zjawiska konieczne byłoby przeprowadzenie większej ilości badań oraz obliczeń na trudnych do pozyskania danych o większej rozdzielczości czasowej.

Wyniki testu Diebolda-Mariano dla RMSE dla kryptowalut i metali szlachetnych

Tabela 6.8. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Instrument	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
Ethereum	0.06	0.28	0.02	0.26	0.02	0.24
Złoto	0.12	0.12	0.26	0.24	0.27	0.22
Srebro	0.49	0.55	0.70	0.76	0.65	0.69
Cardano	1.83	1.74	1.45	1.41	1.37	1.36
Bitcoin	2.09	2.47	1.81	2.08	1.87	2.14
Litecoin	2.46	3.22	2.28	2.67	2.09	2.63
Dogecoin	3.32	3.09	3.70	3.33	3.70	3.33

Źródło: Obliczenia własne.

Tabela 6.9. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz poziomu cen dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

Instrument	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
Złoto	0.12	0.12	0.26	0.24	0.27	0.22
Ethereum	0.16	0.32	0.10	0.29	0.11	0.27
Srebro	0.50	0.56	0.70	0.77	0.66	0.70
Cardano	2.03	1.86	1.49	1.46	1.45	1.41
Litecoin	2.33	3.10	2.16	2.58	2.00	2.54
Bitcoin	2.35	2.58	1.88	2.10	1.95	2.16
Dogecoin	3.06	3.06	3.41	3.32	3.41	3.33

Źródło: Obliczenia własne.

przedstawiają tabele 6.10 (prognozy stóp zwrotu) oraz tabela 6.11 (prognozy poziomów cen). Hybrydowe modele zostały zestawione z modelami ARMA-GARCH. Otrzymane rezultaty wskazują, że dla wszystkich instrumentów można zauważyć statystycznie istotne pogorszenie jakości prognoz dla każdego instrumentu poza Ethereum i złotem (dla których wyniki są również na niekorzyść modeli hybrydowych).

Poprawa jakości błędów prognoz stóp zwrotu modeli hybrydowych, w rozbiciu na różne typy rynków została zaprezentowana w tabeli 6.12. Wyniki dla prognoz poziomów cen znajdują się w tabeli 6.13. Uśrednienie wszystkich osiągniętych wyników pokazuje, że nawet dla spółek giełdowych całkowity efekt zastosowania hybrydowych modeli wykorzystujących treść wiadomości tekstowych nie jest korzystny, jednak zmiana ta jest najmniejsza (część z badanych spółek otrzymywała lepsze wyniki). Znaczne pogorszenie

Tabela 6.10. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu przy wykorzystaniu obu metod dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH.

	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
Instrument	DM	p	DM	p	DM	p
Ethereum	0.66	.511	0.29	.770	0.23	.822
Złoto	0.81	.419	1.46	.145	1.65	.098
Srebro	**2.31**	**.021**	**2.96**	**.004**	**2.76**	**.006**
Cardano	**4.76**	**2.00E-06**	**4.44**	**8.92E-06**	**3.56**	**4.82E-04**
Bitcoin	**5.51**	**4.95E-08**	**4.31**	**1.69E-05**	**4.45**	**8.60E-06**
Litecoin	**5.87**	**1.27E-08**	**5.34**	**1.04E-07**	**4.75**	**2.12E-06**
Dogecoin	**6.63**	**8.29E-11**	**7.75**	**8.33E-14**	**7.82**	**6.37E-15**

Źródło: Obliczenia własne.

Tabela 6.11. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen przy wykorzystaniu obu metod dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH.

	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
Instrument	DM	p	DM	p	DM	p
Złoto	1.05	.295	1.75	.081	1.44	.149
Ethereum	1.27	.204	0.81	.419	0.99	.322
Srebro	**2.17**	**.030**	**2.57**	**.010**	**2.78**	**.006**
Cardano	**5.85**	**1.48E-08**	**4.41**	**1.08E-05**	**4.84**	**1.45E-06**
Litecoin	**5.75**	**2.50E-08**	**5.78**	**2.19E-08**	**5.66**	**3.42E-08**
Bitcoin	**6.51**	**1.56E-10**	**5.82**	**1.78E-08**	**4.89**	**1.13E-06**
Dogecoin	**6.77**	**1.51E-11**	**7.12**	**1.78E-12**	**7.07**	**3.68E-12**

Źródło: Obliczenia własne.

Tabela 6.12. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu w zależności od typu analizowanego rynku - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH.

	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
Instrumenty finansowe	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
Spółki akcyjne	0.11	0.05	0.09	0.05	0.10	0.08
Metale szlachetne	0.30	0.34	0.48	0.50	0.46	0.46
Kryptowaluty	1.95	2.16	1.85	1.95	1.81	1.94

Źródło: Obliczenia własne.

wyników widoczne jest dla metali szlachetnych i kryptowalut (żaden z instrumentów nie osiągał lepszych wyników).

Wybór rozmiaru enkodera $d_{encoder}$ dla prognozowania stóp zwrotu nie jest możliwy na podstawie uśrednionych wyników. Szczegółowe obliczenia w tabeli 6.8 pokazują, że

Tabela 6.13. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz poziomu cen w zależności od typu analizowanego rynku - porównanie pomiędzy modelem ARMA-GARCH-FENE (z różnymi rozmiarami enkodera $d_{encoder}$) a modelem ARMA-GARCH.

	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
Instrumenty finansowe	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
Spółki akcyjne	0.12	0.02	0.10	0.01	0.10	0.04
Metale szlachetne	0.31	0.34	0.48	0.50	0.46	0.46
Kryptowaluty	1.98	2.18	1.81	1.95	1.78	1.94

Źródło: Obliczenia własne.

dla każdego instrumentu finansowego parametr ten powinien być dobierany indywidualnie. Podobne wnioski można wyciągnąć dla prognoz poziomów cen (tabela 6.9).

Hipoteza, iż *klasyczne modele ekonometrii finansowej typu ARMA-GARCH z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z klasycznymi modelami typu ARMA-GARCH bez tego modułu* zostaje potwierdzona jedynie częściowo. Jakość otrzymanych modeli jest lepsza dla części ze spółek giełdowych, jednak w przypadku kryptowalut i metali szlachetnych dodanie modułu FENE do modeli ARMA-GARCH prowadzi jedynie do pogorszenia wyników prognoz.

6.3.2. Porównanie modeli hybrydowych z autorskim modelem LSTM-FENE

Modele hybrydowe ARMA-GARCH-FENE zostały porównane również z autorskim modelem LSTM-FENE. W niniejszym rozdziale modele hybrydowe zostały potraktowane jako modele bazowe, do których porównywany jest model autorski. Oznacza to, że ujemne wartości mierników $\Delta RMSE$, ΔMAE i ΔMAE, a także ujemne wartości testu Diebolda-Mariano dla RMSE świadczą o przewadze autorskiego modelu LSTM-FENE. Dla każdego instrumentu finansowego wybrany został model hybrydowy z najbardziej korzystnym rozmiarem enkodera $d_{encoder}$.

Wyniki procentowej zmiany błędów prognoz stóp zwrotów wybranych spółek S&P 500 zostały zebrane w tabeli 6.14. Dla większości z nich autorski model osiąga lepsze rezultaty (poza spółkami C, MS, INTC, VZ). W tabeli 6.15 znajdują się procentowe zmiany błędu prognoz poziomów cen (poza spółkami C, MS, INTC, VZ, a także BA, JPM i NVDA). Również w tym przypadku widać przewagę modelu w pełni opartego o sieci neuronowe.

Tabela 6.14. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH-FENE. Ujemna wartość oznacza przewagę modelu opartego wyłącznie o sieci neuronowe LSTM-FENE nad modelem hybrydowym ARMA-GARCH-FENE.

Spółka	ΔRMSE [%]	ΔMAE [%]	Wielkość enkodera
TSLA	-25.13	-3.32	16
NKE	-13.66	-12.27	8
BA	-10.37	-12.80	16
KO	-10.20	-10.26	16
CRM	-6.87	-10.92	4
GM	-5.90	-5.67	4
JPM	-5.37	-7.61	4
MSFT	-5.00	-6.27	16
GS	-2.43	-3.31	8
NVDA	-1.37	-9.65	4
F	-1.20	-2.42	4
C	1.76	-2.04	16
MS	4.31	-3.94	8
INTC	5.60	1.68	4
VZ	7.68	-3.27	4

Źródło: Obliczenia własne.

Tabela 6.15. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz poziomu cen dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH-FENE. Ujemna wartość oznacza przewagę modelu opartego wyłącznie o sieci neuronowe LSTM-FENE nad modelem hybrydowym ARMA-GARCH-FENE.

Spółka	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]	Wielkość enkodera
TSLA	-11.51	-11.76	-6.79	16
KO	-9.37	-9.69	-10.26	16
NKE	-7.31	-5.25	-12.26	8
GM	-6.89	-7.16	-5.68	4
F	-4.61	-6.67	-2.43	4
GS	-2.10	-2.96	-3.31	8
MSFT	-0.83	-2.07	-6.29	16
CRM	-0.39	-4.53	-10.97	4
NVDA	0.09	1.41	-0.89	4
JPM	0.15	-2.22	-7.61	4
BA	1.78	-1.27	-12.80	16
C	2.32	-1.11	-2.04	16
INTC	6.22	1.56	1.69	4
MS	7.92	-1.48	-3.93	8
VZ	8.78	-3.96	-3.26	4

Źródło: Obliczenia własne.

Tabela 6.16. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu przy wykorzystaniu obu metod dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH-FENE.

Spółka	Wynik testu DM	Poziom istotności
TSLA	**-4.55**	**5.53E-06**
NKE	**-3.77**	**1.69E-04**
BA	**-3.25**	**.001**
KO	**-3.05**	**.003**
CRM	**-2.35**	**.019**
GM	**-2.18**	**.030**
JPM	**-1.98**	**.048**
MSFT	-1.79	.073
GS	-1.08	.281
NVDA	-0.71	.479
F	-0.63	.529
C	0.98	.327
MS	1.85	.064
INTC	1.95	.052
VZ	**2.77**	**.006**

Źródło: Obliczenia własne.

Tabela 6.17. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen przy wykorzystaniu obu metod dla badanych spółek indeksu S&P 500 - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH-FENE.

Spółka	Wynik testu DM	Poziom istotności
TSLA	**-3.34**	**.001**
KO	**-2.92**	**.004**
NKE	**-2.53**	**.012**
GM	**-2.30**	**.022**
F	**-1.99**	**.047**
GS	-1.00	.316
MSFT	-0.48	.630
CRM	-0.25	.806
NVDA	0.12	.905
JPM	0.11	.914
BA	1.04	.297
C	1.15	.251
INTC	2.11	.035
MS	2.52	.012
VZ	2.86	.005

Źródło: Obliczenia własne.

Prognozy otrzymane przy pomocy obu typów modeli zostały również sprawdzone testem Diebolda-Mariano dla RMSE. Wyniki testu dla prognoz stóp zwrotu znajdują się w tabeli 6.16, zaś wyniki dla prognoz poziomów cen w tabeli 6.17. W obu tabelach spółki, dla których test był statystycznie istotny (tj. $P < .05$) zostały pogrubione.

Dla prognoz stóp zwrotu istotnie znaczącą poprawę można zaobserwować dla spółek TSLA, NKE, BA, KO, CRM, GM i JPM oraz istotne pogorszenie dla VZ. W przypadku prognoz poziomu cen istotnie znaczącą poprawę można zaobserwować dla spółek TSLA, KO, NKE, GM i F oraz statystycznie istotne pogorszenie dla INTC, MS i VZ.

Identyczna analiza została przeprowadzona poniżej dla kryptowalut i metali szlachetnych. Tabele 6.18 oraz 6.19 przedstawiają procentowe zmiany błędów prognoz dla odpowiednio: stóp zwrotu i poziomów cen. Poza błędem prognozy poziomów cen ΔMAE dla Litecoina, wszystkie wyniki wskazują na przewagę modelu LSTM-FENE nad modelami hybrydowymi. Błędy prognozy stóp zwrotu wypadają na korzyść modelu LSTM-FENE we wszystkich przypadkach. Różnice są nieco mniejsze niż dla spółek giełdowych (porównaj z tabelami 6.14 oraz 6.15).

Tabela 6.18. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz stóp zwrotu dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH-FENE. Ujemna wartość oznacza przewagę modelu opartego wyłącznie o sieci neuronowe LSTM-FENE nad modelem hybrydowym ARMA-GARCH-FENE.

Instrument	ΔRMSE [%]	ΔMAE [%]	Wielkość enkodera
Dogecoin	-6.82	-3.53	16
Srebro	-3.21	-3.21	16
Złoto	-2.76	-2.11	16
Ethereum	-1.66	-1.37	4
Bitcoin	-1.62	-1.26	8
Cardano	-1.25	-0.60	4
Litecoin	-1.00	-0.39	4

Źródło: Obliczenia własne.

Tabela 6.19. Procentowa zmiana błędów prognoz RMSE i MAE dla prognoz poziomu cen dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH-FENE. Ujemna wartość oznacza przewagę modelu opartego wyłącznie o sieci neuronowe LSTM-FENE nad modelem hybrydowym ARMA-GARCH-FENE.

Instrument	ΔRMSE [%]	ΔMAE [%]	ΔMAPE [%]	Wielkość enkodera
Dogecoin	-9.09	-6.62	-3.52	16
Srebro	-2.95	-3.03	-3.20	16
Złoto	-2.11	-1.42	-2.11	16
Ethereum	-1.76	-1.58	-1.36	4
Bitcoin	-1.72	-1.49	-1.26	8
Cardano	-1.58	-1.45	-0.60	4
Litecoin	-0.63	0.15	-0.38	4

Źródło: Obliczenia własne.

W tabeli 6.20 zebrane zostały wyniki testu Diebolda-Mariano przeprowadzonego dla RMSE dla prognoz stóp zwrotu. W tabeli 6.21 znajdują się analogiczne wyniki dla poziomów cen. Model LSTM-FENE osiąga istotnie lepsze wyniki dla wszystkich kryptowalut i metali szlachetnych w porównaniu z modelami hybrydowymi ARMA-GARCH-FENE.

Tabela 6.20. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz stóp zwrotu przy wykorzystaniu obu metod dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH-FENE.

Instrument	Wynik testu DM	Poziom istotności
Dogecoin	**-10.75**	**8.70E-25**
Srebro	**-6.97**	**7.47E-12**
Złoto	**-6.19**	**8.06E-10**
Ethereum	**-4.71**	**2.61E-06**
Bitcoin	**-4.52**	**6.28E-06**
Cardano	**-3.78**	**1.62E-04**
Litecoin	**-3.25**	**0.001**

Źródło: Obliczenia własne.

Tabela 6.21. Wyniki testu Diebolda-Mariano dla RMSE dla prognoz poziomu cen przy wykorzystaniu obu metod dla badanych metali szlachetnych i kryptowalut - porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH-FENE.

Instrument	Wynik testu DM	Poziom istotności
Dogecoin	**-11.77**	**3.00E-29**
Srebro	**-7.48**	**4.98E-13**
Złoto	**-5.16**	**2.74E-07**
Ethereum	**-4.94**	**8.04E-07**
Bitcoin	**-5.15**	**2.84E-07**
Cardano	**-4.49**	**7.30E-06**
Litecoin	**-3.04**	**0.003**

Źródło: Obliczenia własne.

Autorski model LSTM-FENE osiąga lepsze wyniki niż modele hybrydowe ARMA-GARCH-FENE niemal dla wszystkich badanych instrumentów i pod względem niemal wszystkich mierników. Jedynie w nielicznych przypadkach wykorzystanie modeli hybrydowych mogłoby być uzasadnione. Dodatkowo na niekorzyść modeli hybrydowych przemawia konieczność wcześniejszego stworzenia modelu LSTM-FENE, który osiąga lepsze wyniki. W takiej sytuacji tworzenie na jego podstawie gorszego modelu jest bezpodstawne.

Na podstawie powyższych wyników należy odrzucić hipotezę: *klasyczne modele ekonometrii finansowej typu ARMA-GARCH z autorskim modułem FENE generują lepsze*

(dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z modelami hybrydowymi opartymi na głębokich sieciach neuronowych z autorskim modułem FENE. Modele w pełni oparte o głębokie sieci neuronowe pozwalają otrzymać lepsze wyniki niż klasyczne modele ekonometryczne ARMA-GARCH nawet po połączeniu ich z wcześniej wytrenowanym modułem FENE.

6.4. Podsumowanie wyników badań

Przedstawione w tym rozdziale wyniki dotyczą możliwości zastosowania klasycznych modeli ekonometrycznych ARMA-EGARCH wzbogaconych informacjami pochodzącymi z sieci neuronowej, przetwarzającej tekst treści wiadomości pojawiających się w portalach informacyjnych. Dane te są przekazywane do modelu jako egzogeniczne szeregi czasowe o wymiarowości $d_{encoder}$ określanej już na etapie tworzenia sieci neuronowej.

Na podstawie niniejszych badań możliwe było częściowe potwierdzenie ostatniej z hipotez postawionych we wstępie: *zautomatyzowana analiza finansowych wiadomości i postów w mediach społecznościowych za pomocą algorytmów NLP w połączeniu z obecnie stosowanymi metodami prognozowania finansowych szeregów czasowych pozwala na poprawienie efektywności predykcji.*

Poprzez wykorzystanie modeli hybrydowych możliwe jest poprawienie jakości prognoz stóp zwrotu i poziomów cen dla niektórych z badanych spółek akcyjnych. Poprawa jest mniejsza niż w przypadku modelu LSTM-FENE opartego w całości o sieć neuronową, jednak dla pięciu ze spółek poprawa jest statystycznie znacząca.

Wykorzystanie modelu LSTM-FENE w niemal wszystkich przypadkach gwarantuje lepsze wyniki niż w przypadku hybrydowych modeli ARMA-GARCH-FENE. Aby móc wykorzystać moduł FENE, konieczne jest wcześniejsze wytrenowanie pełnego modelu LSTM-FENE. Powoduje to, że modelowanie przy pomocy modeli hybrydowych jest nieuzasadnione.

Hybrydowe modele dla kryptowalut i metali szlachetnych nie osiągnęły wyników lepszych niż modele bazowe. Różnice w jakości prognoz nie były jednak duże i wartościowe byłoby przeprowadzenie dalszych badań dla większej rozdzielczości czasowej.

Rozdział 7

Konkluzje i podsumowanie

W dzisiejszym dynamicznym i nieprzewidywalnym środowisku rynków finansowych, skuteczne prognozowanie cen akcji pozostaje jednym z kluczowych wyzwań dla zarówno inwestorów, jak i naukowców. Tradycyjne modele ekonometryczne, oparte na historycznych danych liczbowych, były przez lata głównym narzędziem analizy i prognozowania w tej dziedzinie. Jednakże, w ostatnich latach, postęp w dziedzinie NLP oraz uczenia maszynowego otwiera nowe perspektywy dla analizy finansowej poprzez wykorzystanie treści tekstowych dostępnych powszechnie w Internecie.

Celem niniejszej pracy doktorskiej było zbadanie potencjału wykorzystania technik NLP w procesie prognozowania cen akcji oraz porównanie wyników uzyskanych przy użyciu autorskiego modelu LSTM-FENE z tradycyjnymi modelami ekonometrycznymi typu ARMA-GARCH. W tym kontekście, praca skupiła się na analizie wiadomości pojawiających się na portalach informacyjnych i wzbogaceniu sieci neuronowej typu LSTM o te dane tekstowe.

W efekcie niniejsza praca nie tylko poszerza naszą wiedzę na temat zastosowania NLP w dziedzinie finansów, ale również wnosi praktyczny wkład w rozwój narzędzi prognozowania cen akcji. Wyniki eksperymentów wskazują na potencjał modelu LSTM-FENE w poprawie jakości predykcji w porównaniu z klasycznymi modelami ekonometrycznymi, otwierając nowe horyzonty w podejściach do analizy i prognozowania stóp zwrotu i cen instrumentów finansowych na rynkach finansowych.

W ramach niniejszej pracy doktorskiej szczegółowo omówiono proces implementacji, przedstawiono wyniki eksperymentalne oraz dokonano analizy ich implikacji teoretycznych i praktycznych. Praca ta nie tylko stanowi wkład w dziedzinie finansów i analizy rynków kapitałowych, lecz także stwarza podstawy do dalszych badań i eksploracji zastosowań technologii NLP w prognozowaniu cen akcji oraz innych obszarów finansowych.

7.1. Metodologia badawcza

W kontekście prognozowania stóp zwrotu i poziomów cen akcji, analiza wiadomości z internetowych źródeł za pomocą algorytmów NLP może dostarczyć cennych informacji, które tradycyjne modele oparte na danych liczbowych mogą pomijać. Główną zaletą wykorzystania NLP jest zdolność do ekstrakcji kontekstu, nastroju inwestorów oraz ukrytych informacji z tekstu, które mogą wpływać na zachowanie rynków finansowych.

Do przeprowadzenia eksperymentów wykorzystano zbiór danych zawierający wiadomości z różnych źródeł internetowych (portali informacyjnych). Każda wiadomość była opatrzona datą publikacji. Dane tekstowe zostały poddane procesowi wstępnej obróbki, obejmującej tokenizację, usuwanie znaków specjalnych oraz normalizację tekstu. Zbiór danych zawierał także historyczne dane cen akcji oraz związane z nimi miary ekonomiczne.

Bazą do wszelkich analiz było estymowanie modeli ARMA-GARCH wykorzystujących wzorce aktywności. Prognozy tych modeli posłużyły jako punkt odniesienia dla metod predykcji proponowanych przez autora.

Implementacja modelu LSTM-FENE opierała się głównie na bibliotece uczenia maszynowego TensorFlow, w języku Python. Sieć prognozująca stopy zwrotu oparta była głównie o klasyczną architekturę sieci typu LSTM, która jako stan pośredni otrzymywała wyniki obliczeń pochodzące z dużego pre-trenowanego modelu językowego typu BERT, który następnie podlegał procesowi *fine-tuningu* wraz z resztą modelu. Model został wytrenowany na danych historycznych, wykorzystując techniki regularyzacji, takie jak *dropout* w celu uniknięcia przeuczenia. Dobór hiperparametrów został przeprowadzony za pomocą technik walidacji, aby znaleźć optymalne wartości dla parametrów modelu.

Podjęta została również próba stworzenia modeli hybrydowych, tj. klasycznych modeli ARMA-GARCH, które wykorzystują dane zawarte w wiadomościach tekstowych. W tym celu moduł FENE (po wcześniejszym wytrenowaniu) używany był jako *enkoder* - pozwalając przetłumaczyć informacje tekstowe na wielowymiarowy szereg czasowy. Tak otrzymany szereg był następnie stosowany jako dane egzogeniczne dla klasycznych modeli ekonometrycznych.

W celu oceny jakości predykcji modelu LSTM-FENE i modelu hybrydowego ARMA-GARCH-FENE oraz porównania go z tradycyjnymi modelami ekonometrycznymi,

wykorzystano szereg miar trafności prognoz. Do tych miar należą m.in. RMSE i MAE. Poza tym, otrzymane szeregi prognoz zostały porównane ze sobą przy pomocy testu Diebolda-Mariano dla RMSE, który umożliwia prawidłową ocenę, który z modeli jest statystycznie znacząco lepszy.

7.2. Wyniki eksperymentów

W rozdziałach 5 i 6 opisane zostały szczegółowe wyniki eksperymentów przeprowadzonych z wykorzystaniem modelu LSTM-FENE (w całości opartego o sieć neuronową) oraz modelu hybrydowego (modelu ARMA-GARCH-FENE). W obu przypadkach analiza polegała na porównaniu jakości prognoz z prognozami klasycznych modeli ARMA-GARCH. Analiza przeprowadzona została dla stóp zwrotu oraz poziomów cen 15 najbardziej medialnych spółek giełdowych z indeksu S&P 500, 5 popularnych kryptowalu, a także złota i srebra.

7.2.1. Autorski model LSTM-FENE

Wyniki wskazują, że model LSTM-FENE oparty w całości o sieci neuronowe osiąga lepsze wyniki w porównaniu z modelami ARMA-GARCH, pod względem analizowanych metryk oceny dla większości wybranych spółek giełdowych. Poprawa jest widoczna w przypadku metryk RMSE oraz MAE. Test Diebolda-Mariano dla RMSE dowodzi też statystycznie znaczącą wyższość modelu dla ośmiu z wybranych spółek.

W przypadku metali szlachetnych i kryptowalut wyniki są tylko nieznacznie lepsze (w zaledwie dwóch przypadkach, kiedy test Diebolda-Mariano dla RMSE daje wynik statystycznie istotny). Może to wynikać z większej zmienności tych instrumentów finansowych, a także ogromnej liczby wiadomości i plotek, które znajdują się w Internecie.

Model LSTM-FENE daje zauważalną poprawę jakości prognoz stóp zwrotu i poziomów cen zwłaszcza w okresie bezpośrednio po wystąpieniu nowych wiadomości na temat danej spółki lub instrumentu. Wynika to z faktu, iż to właśnie publikacja wiadomości dostarcza uczestnikom rynku **nowych** informacji.

Tabela 7.1 przedstawia procentową zmianę błędów prognoz stóp zwrotu wynikającą z użycia autorskiego modelu LSTM-FENE zamiast klasycznych ekonometrycznych modeli ARMA-GARCH dla 8 spółek, dla których zmiana jest największa. Wykorzystanie modelu LSTM-FENE jest wyjątkowo korzystne.

Tabela 7.1. Procentowa zmiana błędów predykcji stóp zwrotu dla 5 spółek indeksu S&P 500, dla których różnica jest największa. Porównanie pomiędzy modelem LSTM-FENE a modelem ARMA-GARCH.

Spółka	ΔRMSE [%]	ΔMAE [%]
TSLA	-25.42	-3.54
NKE	-12.85	-11.99
KO	-10.46	-10.33
BA	-10.16	-12.68
CRM	-7.83	-11.55
GM	-5.55	-5.62
JPM	-5.40	-7.65
MSFT	-5.01	-6.21

Źródło: Obliczenia własne.

Autorski model LSTM-FENE osiągał szczególnie dobre wyniki dla spółek TSLA (Tesla Inc.), NIKE Inc. (NKE), Coca-Cola Company (KO), General Motors Company (GM), Boeing Company (BA), Salesforce.com Inc. (CRM), JPMorgan Chase & Co. (JPM) oraz Microsoft (MSFT). Są to jedne z najbardziej medialnych spółek indeksu S&P 500, o których wiadomości pojawiają sie często w wielu portalach o tematyce finansowej i nie tylko. Ustalenie cechy, która łączy te spółki, wymaga dalszych badań.

Stopień poprawy jakości prognoz jest uzależniony od źródła, z którego pochodzą wiadomości. Średnia dzienna liczba wiadomości ani ich długość nie są znacząco skorelowane z otrzymywanymi wynikami.

Model LSTM-FENE zachowywał się stabilnie w przypadku zmiany hiperparametrów w otoczeniu ich optimum. Ostateczne hiperparametry zostały dobrane dzięki wykorzystaniu techniki walidacji.

7.2.2. Hybrydowe modele wykorzystujące techniki NLP

Modele hybrydowe (ARMA-GARCH-FENE) również są w stanie uwzględniać dodatkowe informacje zawarte w wiadomościach o tematyce finansowej. Otrzymane wyniki świadczą jednak o dużo mniejszym wpływie tych informacji na wyniki niż w przypadku czystych sieci neuronowych.

Użycie hybrydowych modeli skutkuje ledwie mierzalną poprawą błędów prognoz w przypadku części spółek giełdowych i pogorszeniem jakości prognoz dla metali szlachetnych i kryptowalut. Test Diebolda-Mariano dla RMSE potwierdza poprawę w przypadku 3-5 (w zależności od wielkości enkodera i wielkości prognozowanej) z piętnastu

wybranych spółek giełdowych oraz ukazuje pogorszenie wyników dla innych instrumentów finansowych.

Wpływ wielkości enkodera tekstowej sieci neuronowej został zbadany dla trzech różnych opcji $d_{encoder} \in \{4, 8, 16\}$. Nie znaleziono odpowiedzi na to, który z nich jest najlepszy - parametr ten musi być dobierany indywidualnie w zależności od badanego instrumentu finansowego.

W tabeli 7.2 zebrane zostały wyniki zmiany wielkości błędu dla hybrydowych modeli ARMA-GARCH-FENE o różnej wielkości enkodera w odniesieniu do klasycznych modeli ARMA-GARCH.

Tabela 7.2. Procentowa zmiana błędów predykcji stóp zwrotu dla badanych spółek indeksu S&P 500. Porównanie pomiędzy modelem ARMA-GARCH-FENE a modelem ARMA-GARCH. Ujemna wartość oznacza przewagę badanego modelu nad modelem bazowym.

	$d_{encoder} = 16$		$d_{encoder} = 8$		$d_{encoder} = 4$	
Spółka	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
MSFT	-1.92	-2.35	-1.78	-2.23	-1.71	-2.10
GM	-0.73	-0.76	-0.76	-0.91	-0.87	-0.98
CRM	-0.42	-0.44	-0.47	-0.48	-0.47	-0.55
INTC	-0.40	-0.66	-0.39	-0.66	-0.51	-0.75
TSLA	-0.34	-3.46	-0.27	-2.74	-0.26	-2.57

Źródło: Obliczenia własne.

W niektórych przypadkach zmiana jest korzystna, jednak uśredniając błędy dla wszystkich badanych instrumentów, wyniki nie są lepsze niż osiągane przez modele klasyczne. W tabeli 7.3 zebrane zostały procentowe zmiany błędów (względem modeli ARMA-GARCH) dla modelu LSTM-FENE oraz ARMA-GARCH-FENE. Niniejsza tabela potwierdza, że autorski model ma przewagę nad modelami hybrydowymi, zwłaszcza użyty do prognozowania stóp zwrotu spółek akcyjnych.

Tabela 7.3. Procentowa zmiana błędów predykcji stóp zwrotu dla różnych typów rynków. Tabela porównuje, o ile zmienia się błąd względem klasycznych modeli ARMA-GARCH przez wykorzystanie modelu LSTM-FENE lub ARMA-GARCH-FENE.

	LSTM-FENE		**ARMA-GARCH-FENE**	
Instrumenty finansowe	ΔRMSE [%]	ΔMAE [%]	ΔRMSE [%]	ΔMAE [%]
Spółki akcyjne	-4.56	-6.19	0.09	0.05
Metale szlachetne	-0.10	0.02	0.30	0.34
Kryptowaluty	-0.38	0.42	1.81	1.94

Źródło: Obliczenia własne.

Mimo iż wyniki osiągane przez modele hybrydowe nie są tak dobre, jak wyniki modelu LSTM-FENE opartego wyłącznie na architekturze sieci neuronowej, również one mogą być poprawione przez odpowiednie wykorzystanie informacji pochodzących z treści wiadomości na temat spółek. Świadczy to o znacznym potencjale wiedzy zawartej w takich publikacjach.

7.2.3. Implikacje teoretyczne i praktyczne

Jedną z głównych różnic między modelem LSTM-FENE, a modelami ekonometrycznymi jest elastyczność. Modele ekonometryczne opierają się na matematycznym opisie relacji między zmiennymi i przyjmują pewne założenia dotyczące struktury danych. W przeciwieństwie do takiego podejścia, model LSTM-FENE, oparty o ateoretyczne sieci neuronowe, jest zdolny do wykrywania złożonych wzorców w danych, nawet jeśli struktura nie jest dobrze zdefiniowana. Dzięki temu, może on uwzględniać nieliniowe zależności między danymi tekstowymi a cenami akcji.

Sieci neuronowe są zdolne do uwzględniania kontekstu poprzez zautomatyzowaną analizę tekstu (NLP). Informacje zawarte w wiadomościach, takie jak wiadomości sektorowe, zmiany w zarządzaniu firmą czy wydarzenia globalne, mogą mieć znaczący wpływ na ceny akcji. Klasyczne modele ekonometryczne opierają się głównie na danych liczbowych, co może prowadzić do pominięcia kluczowych informacji.

Rynki finansowe są dynamiczne i często reagują na szybko zmieniające się warunki. Model LSTM-FENE ma przewagę w reagowaniu na nowości, ponieważ jest zdolny do analizy wiadomości w czasie rzeczywistym. Modele ekonometryczne, które często opierają się na historycznych danych, mogą mieć trudności w radzeniu sobie z nagłymi zmianami warunków rynkowych.

Modele ekonometryczne często charakteryzują się wysoką interpretowalnością, ponieważ są oparte na matematycznych równaniach. W przypadku modelu LSTM-FENE, interpretowalność może być trudniejsza, ponieważ działanie sieci neuronowych jest bardziej skomplikowane. Niemniej jednak, narzędzia ekstrakcji informacji mogą pomóc zrozumieć, które aspekty tekstu wpływają na prognozy.

Modele oparte o głębokie sieci neuronowe (w tym autorski model LSTM-FENE) wymagają większej mocy obliczeniowej niż klasyczne modele ekonometryczne. Ponadto precyzyjna analiza tekstu wymaga odpowiednio przetworzonych i zrozumiałych danych tekstowych. Modele ekonometryczne są mniej wymagające obliczeniowo i mogą działać dobrze nawet na mniej zaawansowanych komputerach.

Przeprowadzenie eksperymentów pozwoliło na weryfikację hipotez badawczych postawionych we wstępie niniejszej pracy:

***Hipoteza*: Możliwe jest efektywne zautomatyzowanie analizy finansowych wiadomości oraz postów w mediach społecznościowych poprzez użycie współczesnych metod przetwarzania języka naturalnego – głównie głębokich sieci neuronowych.**

Wynik weryfikacji: Hipoteza została potwierdzona. Wyniki porównania prognoz modelu LSTM-FENE z modelami bazowymi okazały się korzystne dla tego pierwszego. Wykorzystanie danych tekstowych powoduje zmniejszenie błędów prognoz (RMSE oraz MAE) dla jedenastu z piętnastu wybranych spółek, a także błędu RMSE wszystkich kryptowalut i metali szlachetnych. Test Diebolda-Mariano dla RMSE potwierdza, że wyniki te są statystycznie znaczące dla siedmiu z piętnastu spółek i dla jednej z kryptowalut.

***Hipoteza*: Wykorzystanie treści całej wiadomości w analizie pozwala na osiągnięcie lepszych wyników prognozowania niż w przypadku wykorzystania wyłącznie treści nagłówków**.

Wynik weryfikacji: Hipoteza została potwierdzona. Model LSTM-FENE, który został wytrenowany na danych zawierających jedynie nagłówki wiadomości pozwala otrzymywać prognozy lepsze niż klasyczne modele ekonometryczne. Poprawa błędów jednak nie jest tak znaczna, jak w przypadku modelu wykorzystującego pełną treść wiadomości. Model mający dostęp do danych z pełnymi treściami publikacji pozwala osiągnąć większą poprawę błędów prognoz, a jego wyniki są statystycznie znaczące dla większej liczby spółek.

***Hipoteza*: Modele oparte na głębokich sieciach neuronowych z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z modelami wykorzystującymi klasyczne metody sztucznej inteligencji.**

Wynik weryfikacji: Hipoteza została częściowo potwierdzona. Głębokie sieci neuronowe są w stanie przechwycić dodatkowe informacje zawarte w tekście wiadomości dzięki modułowi FENE. Wyniki otrzymane z wykorzystaniem autorskiego modułu dla spółek giełdowych są lepsze niż bez niego. Dla kryptowalut oraz metali szlachetnych

zmiany nie są jednak widoczne i hipoteza nie może być w pełni potwierdzona.

***Hipoteza*: Wykorzystanie szeregu czasowego notowań o 5 minutowej częstotliwości jest wystarczające, aby zaobserwować zmienioną charakterystykę zwrotów spowodowaną publikacją wiadomości.**

Wynik weryfikacji: Hipoteza została potwierdzona. Modelowanie śródziennych logarytmicznych stóp zwrotu o 5-minutowej rozdzielczości wraz z danymi tekstowymi pochodzącymi z wiadomości finansowych publikowanych w Internecie pozwala na zaobserwowanie podniesionej skuteczności prognoz. Tak duża rozdzielczość umożliwia modelom wyłapanie efektów widocznych w okresach bezpośrednio po wystąpieniu nowych publikacji na temat różnych instrumentów finansowych.

***Hipoteza*: Zautomatyzowana analiza finansowych wiadomości i postów w mediach społecznościowych za pomocą algorytmów NLP w połączeniu z obecnie stosowanymi metodami prognozowania finansowych szeregów czasowych pozwala na poprawienie efektywności predykcji.**

Wynik weryfikacji: Hipoteza została częściowo potwierdzona. Stworzenie modeli hybrydowych bazujących na architekturze ARMA-GARCH wzbogaconych o dane egzogeniczne wygenerowane przez moduł FENE prowadzi do otrzymywania lepszej jakości prognoz tylko dla części z wybranych spółek. Jednak nie poprawia jakości prognoz bardziej zmiennych instrumentów (metali szlachetnych lub kryptowalut). W przypadku tego typu modeli, zmiany są jednak niewielkie, a statystycznie znacząca poprawa prognoz była zaobserwowana wyłącznie w przypadku jednej spółki.

***Hipoteza*: Klasyczne modele ekonometrii finansowej typu ARMA-GARCH z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z klasycznymi modelami typu ARMA-GARCH bez tego modułu.**

Wynik weryfikacji: Hipoteza została częściowo potwierdzona. Wykorzystanie modułu FENE do generowania dodatkowych egzogenicznych danych dla modeli typu ARMA-GARCH prowadzi do otrzymywania dokładniejszych prognoz dla większości spółek giełdowych. Wyniki osiągane dla kryptowalut i metali szlachetnych nie są jednak w znaczący sposób lepsze od bazowych modeli i hipoteza nie może być w pełni potwierdzona.

***Hipoteza*: Klasyczne modele ekonometrii finansowej typu ARMA-GARCH z autorskim modułem FENE generują lepsze (dokładniejsze) punktowe prognozy cen i stóp zwrotu aktywów finansowych w porównaniu z modelami hybrydowymi opartymi na głębokich sieciach neuronowych z autorskim modułem FENE.**

Wynik weryfikacji: Hipoteza została odrzucona. Modele w pełni oparte o głębokie sieci neuronowe LSTM-FENE osiągają wyniki lepsze niż hybrydowe modele ARMA-GARCH-FENE dla wszystkich badanych klas instrumentów finansowych.

Wprowadzenie modelu LSTM-FENE wykorzystującego techniki NLP do analizy finansowej niesie za sobą szereg implikacji teoretycznych. Dowodzi to, że zaawansowane technologie przetwarzania języka naturalnego mają potencjał do poprawy jakości prognoz cen akcji. Wykorzystanie ekstrakcji kluczowych informacji z tekstów wiadomości pozwala na uchwycenie nieuwzględnianych dotąd aspektów, które mogą wpływać na zachowanie rynków finansowych. Implikacje teoretyczne rozciągają się również na pole nauki o danych, potwierdzając, że hybrydowe modele łączące dane liczbowe z tekstem, mogą zwiększyć zdolność modeli do prognozowania.

Wprowadzenie modelu LSTM-FENE do praktyki inwestycyjnej może przynieść rewolucyjne zmiany. Inwestorzy i menedżerowie portfeli mogą wykorzystać bardziej precyzyjne prognozy cen akcji w procesach decyzyjnych. Możliwość uwzględniania kluczowych informacji z wiadomości pozwala na szybkie i celne reagowanie na istotne wydarzenia związane z firmami, sektorami czy globalnymi trendami ekonomicznymi. W praktyce oznacza to oparte na lepszych informacjach i trafne decyzje inwestycyjne.

7.3. Ograniczenia i perspektywy dalszych badań

Autorski model LSTM-FENE, mimo swoich zalet, posiada pewne ograniczenia. Większość z nich jest możliwa do pokonania, jednak warto je uwzględnić i świadomie im zapobiegać. Poniżej wymieniono podstawowe ograniczenia i potencjalne sposoby ich uniknięcia.

— **Analiza tekstu jest wrażliwa na jakość danych.** Nieregularności, błędy lub niedoskonałości w tekście źródłowym mogą wpływać na jakość ekstrakcji informacji. Szczególnie problematyczna może okazać się analiza danych tekstowych pobieranych automatycznie z wielu różnorodnych stron z wiadomościami. Należy przyłożyć

szczególną uwagę do oczyszczenia takich zbiorów danych z fragmentów reklam lub nagłówków innych artykułów, które mogły przez przypadek zostać pobrane przez *crawler*.

— **Interpretowalność głębokich sieci neuronowych może być trudniejsza w porównaniu z klasycznymi modelami ekonometrycznymi.** Może to stanowić wyzwanie dla osób zarządzających inwestycjami. Stosowanie modeli typu "czarna skrzynka", której zachowania nie można być pewnym, jest niebezpieczne ze względu na ryzyko nieprzewidzianych zachowań. Należy korzystać z metod umożliwiających zrozumienie wewnętrznej struktury modelu. Rozwój popularności głębokich sieci neuronowych oraz dużych modeli językowy doprowadził do stworzenia metodologii pozwalającej lepiej wyjaśnić przepływ danych w takich sieciach [18, 82, 203].

— **Duże wymagania sprzętowe.** Wytrenowanie (a nawet uruchomienie) modeli opartych o głębokie sieci neuronowe wymaga znacznie większej mocy obliczeniowej niż klasyczne modele ekonometryczne. Najbardziej korzystnym sposobem jest obecnie wykorzystanie rozwiązań chmurowych dostarczanych np. przez Google, Microsoft i Amazon, które nie wymagają samodzielnego utrzymania infrastruktury obliczeniowej.

— **Potrzeba dużych zbiorów danych.** Głębokie sieci neuronowe składają się z olbrzymiej liczby parametrów. Aby (skuteczne) wytrenowanie tak dużego modelu było możliwe, konieczne jest użycie bardzo dużych zbiorów danych. Rozwiązaniem może być wstępne wytrenowanie modelu na danych łatwiej dostępnych, a następnie wykorzystanie techniki *transfer learningu*. Jeżeli to możliwe, jeszcze lepszym rozwiązaniem jest użycie pre-trenowanego modelu, jeśli taki jest dostępny.

Rynki finansowe są dynamiczne i podlegają zmianom związanym z nowościami ekonomicznymi, politycznymi oraz innymi czynnikami. Model LSTM-FENE wymaga ciągłego uaktualniania w celu uwzględniania nowych danych oraz wydarzeń. Zaniedbanie aktualizacji może prowadzić do utraty skuteczności modelu w przewidywaniu ruchów cen akcji.

W niektórych przypadkach ekstrakcja informacji może być utrudniona ze względu na złożone wzorce tekstu. Współzależności między słowami i kontekstem, a także zmienne znaczenie terminów, mogą wpływać na trafność analizy. Wymaga to zastosowania bardziej zaawansowanych technik przetwarzania języka naturalnego, aby zwiększyć precyzję analizy. Korzystne byłoby m.in. wykorzystanie bazy wiedzy ze zgroma-

dzonymi faktami na temat analizowanych spółek lub innych instrumentów. Dzięki temu model otrzymywałby dodatkowy kontekst, który nie zawsze jest zawarty w tekście artykułu. Do najbardziej wartościowych informacji, które nie zawsze znajdują się w tekście wiadomości, można zaliczyć:

— listę konkurencyjnych firm - sukcesy i niepowodzenia konkurencji mogą wpływać na kurs badanej spółki,
— listę produktów i usług - artykuły opisujące produkty i usługi nie zawsze mają jasne odniesienia do spółki, która je dostarcza,
— lista powiązanych osób - wielu dyrektorów i inwestorów wyższego szczebla jest powiązanych z kilkoma spółkami i wiadomości na ich temat (np. plotki o skandalach) mogą mieć wpływ na nie wszystkie.

Tego typu fakty mogą być dodawane do wejścia części tekstowej przez wykorzystanie *prompt engineeringu*, czyli zapisanie ich w postaci tekstu, choć niekoniecznie w sformalizowany sposób.

Mimo ograniczeń, wprowadzenie modelu korzystającego z technik NLP w analizie finansowej otwiera wiele perspektyw dalszych badań. Istnieje potencjał do rozwinięcia bardziej zaawansowanych technik ekstrakcji informacji, co może poprawić jakość prognoz. Hybrydowe modele łączące dane tekstowe z innymi danymi finansowymi mogą również znacząco rozszerzyć zastosowania tej technologii w dziedzinie analizy finansowej. Ponadto badania nad optymalizacją hiperparametrów modelu oraz testy na różnych rynkach i częstotliwościach danych, mogą dostarczyć bardziej wszechstronne i ogólne wyniki.

Dodatkowe badania, które mogłyby znacząco poprawić prezentowane w niniejszej pracy wyniki, obejmują głównie pracę nad danymi wejściowymi modeli, w tym:

— Zastosowanie większej (a także mniejszej) rozdzielczości. Prowadzenie obliczeń na danych o 1 minutowym skoku nie było możliwe w tych badaniach ze względu na znacznie większą złożoność obliczeniową oraz dostępność danych.
— Wykorzystanie źródeł wiadomości innych niż tylko portale informacyjne - w tym między innymi Twitter oraz inne media społecznościowe.
— Wykorzystanie publikacji, które nie wspominają bezpośrednio danej spółki, np. informacje o podniesieniu stóp procentowych przez banki centralne czy politycznych wydarzeniach mających wpływ na całą gospodarkę.

Wprowadzenie technik NLP w analizie finansowej może również wiązać się z pewnymi wyzwaniami etycznymi. Analiza tekstu może prowadzić do wykrywania informacji, które są poufne lub które mogą wpłynąć na manipulację rynkami. Konieczne jest odpowiednie zachowanie zasad poufności i etyki w procesie przetwarzania i wykorzystywania danych tekstowych.

7.4. Podsumowanie

W niniejszej pracy doktorskiej przeprowadzona została kompleksowa analiza wykorzystania technik przetwarzania języka naturalnego w prognozowaniu finansowych szeregów czasowych (logarytmicznych stóp zwrotu) dla spółek akcyjnych, metali szlachetnych i kryptowalut. Autorski model LSTM-FENE, który łączy analizę tekstu z danymi liczbowymi, został zaimplementowany i przetestowany na danych historycznych. Wyniki eksperymentów potwierdziły przewagę tego typu modelu nad klasycznymi modelami ekonometrycznymi, zarówno pod względem mierników trafności prognoz, jak i zdolności do uwzględniania kontekstu.

Wprowadzenie modeli opartych o techniki NLP do praktyki inwestycyjnej może przynieść znaczące korzyści. Inwestorzy, analitycy i menedżerowie portfeli mogą korzystać z bardziej precyzyjnych prognoz cen akcji opartych na ekstrakcji kluczowych informacji z wiadomości. To z kolei może prowadzić do decyzji inwestycyjnych opartych o konkretne informacje, zwiększając potencjał generowania zysków i minimalizując ryzyko.

Niniejsza praca otwiera drzwi do wielu kierunków dalszych badań. Dalsza optymalizacja modelu LSTM-FENE, badania nad bardziej zaawansowanymi technikami ekstrakcji informacji, a także rozwinięcie hybrydowych modeli łączących różne źródła danych, mogą przyczynić się do dalszego zwiększenia skuteczności prognoz. Rozszerzenie modelu na inne instrumenty finansowe oraz analizę ryzyka finansowego stanowi kolejne intrygujące wyzwania badawcze.

Wprowadzenie technologii przetwarzania języka naturalnego do analizy finansowej ma potencjał do zmiany sposobu, w jaki podchodzi się do prognozowania cen akcji. Model LSTM-FENE, który łączy dane tekstowe z danymi liczbowymi, pokazuje, że ekstrakcja informacji z wiadomości może dostarczyć istotnych informacji, które wpływają na zachowanie rynków finansowych. Choć istnieją ograniczenia i wyzwania, dalsze badania oraz rozwój technologii mogą przynieść znaczący postęp w tej dziedzinie.

Niniejsza praca jest ukierunkowana na stworzenie mostu między dziedzinami finansów i przetwarzania języka naturalnego, otwierając nowe horyzonty w analizie finansowej. Otrzymane wyniki dają nadzieję na przyszłość, w której zaawansowane technologie będą stały się integralną częścią narzędzi inwestycyjnych, przyczyniając się do lepszego zrozumienia i prognozowania zachowań rynków kapitałowych.

Bibliografia

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015. `http://tensorflow.org/`.

[2] S. Agarwal, S. Kumar, and U. Goel, *Stock Market Response to Information Diffusion through Internet Sources: A literature review and research agenda in Indian Context*, 2017.

[3] Alpha Vantage Inc., *Alpha Vantage Website*. `https://www.alphavantage.co/`.

[4] W. Antweiler and M. Z. Frank, *Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards*, The Journal of Finance **59** (2004), no. 3, 1259–1294.

[5] R. T Baillie, T. Bollerslev, and H. O. Mikkelsen, *Fractionally integrated generalized autoregressive conditional heteroskedasticity*, Journal of econometrics **74** (1996), no. 1, 3–30.

[6] N. Barberis, A. Shleifer, and R. Vishny, *A model of investor sentiment*, Journal of financial economics **49** (1998), no. 3, 307–343.

[7] L. Bauwens, W. B. Omrane, and P. Giot, *News announcements, market activity and volatility in the euro/dollar foreign exchange market*, Journal of International Money and Finance **24** (2005), no. 7, 1108–1125.

[8] I. Beltagy, K. Lo, and A. Cohan, *SciBERT: A pretrained language model for scientific text*, arXiv preprint arXiv:1903.10676 (2019).

[9] C. Bergmeir and J. M Benítez, *On the use of cross-validation for time series predictor evaluation*, Information Sciences **191** (2012), 192–213.

[10] E. Birbeck and D. Cliff, *Using Stock Prices as Ground Truth in Sentiment Analysis to Generate Profitable Trading Signals*, 2018 ieee symposium series on computational intelligence (ssci), 2018, pp. 1868–1875.

[11] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, *Enriching Word Vectors with Subword Information*, Transactions of the Association for Computational Linguistics **5** (2017), 135–146.

[12] T. Bollerslev, *Generalized autoregressive conditional heteroskedasticity*, Journal of Econometrics **31** (1986), no. 3, 307–327.

[13] K. Boudt, C. Croux, and S. Laurent, *Robust estimation of intraweek periodicity in volatility and jump detection*, Journal of Empirical Finance **18** (2011), no. 2, 353–367.

[14] C. Brooks, *Introductory Econometrics for Finance*, 4th ed., Cambridge University Press, 2019.

[15] Raymond M. Brooks, A. Patel, and T. Su, *How the Equity Market Responds to Unanticipated Events*, The Journal of Business **76** (2003), no. 1, 109–133.

[16] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, *Language Models are Few-Shot Learners*, arXiv preprint arXiv:2005.14165 (2020).

[17] M. Buczynski and M. Chlebus, *GARCHNet: Value-at-Risk Forecasting with GARCH Models Based on Neural Networks*, Computational Economics (2023), 1–31.

[18] N. Burkart and M. F Huber, *A survey on the explainability of supervised machine learning*, Journal of Artificial Intelligence Research **70** (2021), 245–317.

[19] J. Y Campbell and S. B Thompson, *Predicting excess stock returns out of sample: Can anything beat the historical average?*, The Review of Financial Studies **21** (2008), no. 4, 1509–1531.

[20] S. Casola, I. Lauriola, and A. Lavelli, *Pre-trained transformers: an empirical comparison*, Machine Learning with Applications **9** (2022), 100334.

[21] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, *LEGAL-BERT: The muppets straight out of law school*, arXiv preprint arXiv:2010.02559 (2020).

[22] W. S. Chan, *Stock price reaction to news and no-news: drift and reversal after headlines*, Journal of Financial Economics **70** (2003), no. 2, 223–260.

[23] Q. Chen, *Stock movement prediction with financial news using contextualized embedding from BERT*, arXiv preprint arXiv:2107.08721 (2021).

[24] M. Chlebus, M. Dyczko, and M. Woźniak, *Nvidia's stock returns prediction using machine learning techniques for time series forecasting problem*, Technical Report 2020-22, Faculty of Economic Sciences, University of Warsaw, 2020.

[25] N. Chomsky, *Syntactic Structures*, Mouton and Co., The Hague, 1957.

[26] C. Chua and S. Tsiaplias, *Information flows and stock market volatility*, Journal of Applied Econometrics **34** (2018).

[27] City Falcon Limited, *CityFALCON Website*. `https://www.cityfalcon.com`.

[28] T. E Clark and K. D West, *Approximately normal tests for equal predictive accuracy in nested models*, Journal of econometrics **138** (2007), no. 1, 291–311.

[29] K. M. Colby, F. D. Hilf, S. Weber, and H. C Kraemer, *Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes*, Artificial Intelligence **3** (1972), 199–221.

[30] E. Colombatto and V. Tavormina, *Regulating information flows: Is it just? Insider trading and mandatory-disclosure rules from a free-market perspective*, European Journal of Law and Economics **46** (2018).

[31] Common Crawl Foundation, *Common Crawl*. `https://commoncrawl.org/`.

[32] P. Coulombe, M. Leroux, D. Stevanovic, and S. Surprenant, *How is Machine Learning Useful for Macroeconomic Forecasting?*, arXiv preprint arXiv:2008.12477 (2020).

[33] K. Cuthbertson and D. Nitzsche, *Quantitative Financial Economics*, 2nd ed., Wiley, 2004.

[34] J. Czekaj, *Rynki, instrumenty i instytucje finansowe*, Wydawnictwo Naukowe PWN, 2017.

[35] M. Czerwonka and B. Gorlewski, *Finanse behawioralne. Zachowania inwestorów i rynku.*, Szkoła Główna Handlowa, 2008.

[36] H. Dai, G. Huang, H. Zeng, and F. Yang, *PM2. 5 Concentration prediction based on spatio-temporal feature selection using XGBoost-MSCNN-GA-LSTM*, Sustainability **13** (2021), no. 21, 12071.

[37] W. F. De Bondt and R. Thaler, *Does the stock market overreact?*, The Journal of finance **40** (1985), no. 3, 793–805.

[38] X. Deng, V. Bashlovkina, F. Han, S. Baumgartner, and M. Bendersky, *What do LLMs Know about Financial Markets? A Case Study on Reddit Market Sentiment Analysis*, 2023 acm web conference (www), 2023.

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv preprint arXiv:1810.04805 (2018). `https://github.com/google-research/bert`.

[40] F. Diebold and R. Mariano, *Comparing Predictive Accuracy*, Journal of Business & Economic Statistics **13** (1995), no. 3, 253–63.

[41] R. Diedrigkeit, *Rynek pieniądza i papierów wartościowych*, Zarządzanie i Bankowość, 1993.

[42] X. Ding, Y. Zhang, T. Liu, and J. Duan, *Using Structured Events to Predict Stock Price Movement: An Empirical Investigation*, Proceedings of the 2014 conference on empirical methods in natural language processing ("emnlp"), 2014, pp. 1415–1425.

[43] X. Ding, Y. Zhang, T. Liu, and J. Duan, *Knowledge-Driven Event Embedding for Stock Prediction*, Proceedings of "coling" 2016, the 26th international conference on computational linguistics: Technical papers, 2016, pp. 2133–2142.

[44] M. Doman, *Prognozowanie zmiennosci polskich indeksów giełdowych za pomocą modeli GARCH przy uzyciu danych wysokiej czestotliwosci* **177 Rynki finansowe, prognozy a decyzje** (2004), 291–309.

[45] M. Doman and R. Doman, *Modelowanie zmienności i ryzyka: metody ekonometrii finansowej*, Oficyna a Wolters Kluwer business, 2009.

[46] Y. Dong, D. Yan, A. I. Almudaifer, S. Yan, Z. Jiang, and Y. Zhou, *Belt: A pipeline for stock price prediction using news*, 2020 ieee international conference on big data (big data), 2020, pp. 1137–1146.

[47] W. Dębski, *Rynek finansowy i jego mechanizmy: podstawy teorii i praktyki*, Wydawnictwo Naukowe PWN, 2014.

[48] A. Einstein, *Zur Elektrodynamik bewegter Körper*, Annalen der Physik **14** (1905), no. S1, 194–224.

[49] A. ElBahrawy, L. Alessandretti, A. Kandler, R. Pastor-Satorras, and A. Baronchelli, *Evolutionary dynamics of the cryptocurrency market*, Royal Society open science **4** (2017), no. 11, 170623.

[50] R. Engle, *Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models*, Journal of Business & Economic Statistics **20** (2002), no. 3, 339–350.

[51] R. F. Engle, *Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation*, Econometrica **50** (1982), no. 4, 987–1007.

[52] R. F Engle and T. Bollerslev, *Modelling the persistence of conditional variances*, Econometric reviews **5** (1986), no. 1, 1–50.

[53] R. F Engle, D. M Lilien, and R. P Robins, *Estimating time varying risk premia in the term structure: The ARCH-M model*, Econometrica: journal of the Econometric Society (1987), 391–407.

[54] R. F Engle and V. K Ng, *Measuring and testing the impact of news on volatility*, The journal of finance **48** (1993), no. 5, 1749–1778.

[55] R. F Engle and Z. Sun, *Forecasting volatility using tick by tick data* (2005).

[56] F. J. Fabozzi and F. Modigliani, *Capital Markets: institutions and instruments*, Prentice Hall, 1996.

[57] E. F. Fama, *Efficient Capital Markets: A Review of Theory and Empirical Work*, The Journal of Finance **25** (1970), no. 2, 383–417.

[58] R. Farell, *An analysis of the cryptocurrency industry*, Wharton Research Scholars **130** (2015), 1–23.

[59] S. Feuerriegel and J. Gordon, *Long-term stock index forecasting based on text mining of regulatory disclosures*, Decision Support Systems **112** (2018), 88 –97.

[60] P. Fiszeder, *Modele klasy GARCH w empirycznych badaniach finansowych*, Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika, 2009.

[61] P. Fiszeder et al., *Forecasting the volatility of the polish stock index–WIG20*, Forecasting financial markets. theory and applications (2005), 29–42.

[62] A. Galanos, *rugarch: Univariate GARCH models.*, 2023. R package version 1.5-1.

[63] J. D. Garrett, *garrettj403/SciencePlots* (2021). `http://doi.org/10.5281/zenodo.4106649`.

[64] P. Giot et al., *Time transformations, intraday data and volatility models*, Citeseer, 1999.

[65] P. Glasserman and H. Mamaysky, *Does Unusual News Forecast Market Stress?*, Journal of Financial and Quantitative Analysis **54** (2019), no. 5, 1937–1974.

[66] A. Glodd and D. Hristova, *Extraction of Forward-looking Financial Information for Stock Price Prediction from Annual Reports Using NLP Techniques* (2023).

[67] L. R Glosten, R. Jagannathan, and D. E Runkle, *On the relation between the expected value and the volatility of the nominal excess return on stocks*, The journal of finance **48** (1993), no. 5, 1779–1801.

[68] Y. Goldberg, *A primer on neural network models for natural language processing*, Journal of Artificial Intelligence Research **57** (2016), 345–420.

[69] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, 2016.

[70] Google LLC, *Google Colab Website*. `https://colab.research.google.com/`.

[71] Google LLC, *Google Compute Engine Website*. `https://cloud.google.com/compute`.

[72] John R. Graham, Jennifer L. Koski, and U. Loewenstein, *Information Flow and Liquidity around*

Anticipated and Unanticipated Dividend Announcements, The Journal of Business **79** (2006), no. 5, 2301–2336.

[73] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, *Learning word vectors for 157 languages*, arXiv preprint arXiv:1802.06893 (2018).

[74] G. Grefenstette, *Tokenization*, Syntactic Wordclass Tagging (1999), 117–133.

[75] S. Grossman and J. Stiglitz, *On The Impossibility of Informationally Efficient Markets*, American Economic Review **70** (1980), 393–408.

[76] E. Guresen, G. Kayakutlu, and T. U Daim, *Using artificial neural network models in stock market index prediction*, Expert Systems with Applications **38** (2011), no. 8, 10389–10397.

[77] P. R Hansen, A. Lunde, and J. M Nason, *The model confidence set*, Econometrica **79** (2011), no. 2, 453–497.

[78] P. R. Hansen, *A test for superior predictive ability*, Journal of Business & Economic Statistics **23** (2005), no. 4, 365–380.

[79] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, *Array programming with NumPy*, Nature **585** (2020), no. 7825, 357–362.

[80] Z. S Harris, *Distributional structure*, Word **10** (1954), no. 2-3, 146–162.

[81] D. Harvey, S. Leybourne, and P. Newbold, *Testing the equality of prediction mean squared errors*, International Journal of forecasting **13** (1997), no. 2, 281–291.

[82] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, *Explainability in deep reinforcement learning*, Knowledge-Based Systems **214** (2021), 106685.

[83] J. Z. G. Hiew, X. Huang, H. Mou, D. Li, Q. Wu, and Y. Xu, *BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability*, arXiv preprint arXiv:1906.09024 (2019).

[84] S. Hochreiter and J. Schmidhuber, *Long Short-term Memory*, Neural computation **9** (1997), 1735–80.

[85] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, *spaCy: Industrial-strength natural language processing in python* (2020).

[86] J. Howard and S. Ruder, *Universal language model fine-tuning for text classification*, arXiv preprint arXiv:1801.06146 (2018).

[87] D. A Hsieh, *Modeling heteroscedasticity in daily foreign-exchange rates*, Journal of Business & Economic Statistics **7** (1989), no. 3, 307–317.

[88] D. A Hsieh, *Testing for nonlinear dependence in daily foreign exchange rates*, Journal of Business (1989), 339–368.

[89] Hugging Face, *Hugging Face Website*. `https://huggingface.co/`.

[90] J. Hutchins, *Two precursors of machine translation : Artsrouni and Trojanskij*, 2002.

[91] J. Hutchins, *The first public demonstration of machine translation : the Georgetown-IBM system, 7th January 1954*, 2006.

[92] W J. Hutchins, L. Dostert, and P. Garvin, *The Georgetown-IBM Experiment*, 1955.

[93] R. J Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, OTexts, 2018.

[94] R. J Hyndman and A. B Koehler, *Another look at measures of forecast accuracy*, International journal of forecasting **22** (2006), no. 4, 679–688.

[95] J. J. ibn Ishak al Kindi, M. Mrayati, Y. M. Alam, and M. H. at Tayyan, *Al-Kindi's Treatise on Cryptanalysis*, Al-Kindi's Treatise on Cryptanalysis, King Faisal Center for Research and Islamic Studies, 2003.

[96] Z. Ivković and S. Weisbenner, *Information Diffusion Effects in Individual Investors' Common Stock Purchases: Covet Thy Neighbors' Investment Choices*, The Review of Financial Studies **20** (2007), no. 4, 1327–1357, available at `https://academic.oup.com/rfs/article-pdf/20/4/1327/24429181/hhm009.pdf`.

[97] R. Jennings and L. Starks, *Information Content and the Speed of Stock Price Adjustment*, Journal of Accounting Research **23** (1985), no. 1, 336–350.

[98] J. Jiang, *Information extraction from text*, Mining text data (2012), 11–41.

[99] M. I. Jordan, *Chapter 25 - Serial Order: A Parallel Distributed Processing Approach*, Neural-network models of cognition, 1997, pp. 471–495.

[100] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, *Exploring the limits of language modeling*, arXiv preprint arXiv:1602.02410 (2016).

[101] D. Jurafsky and J. H Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.*

[102] Kaggle, *Kaggle Website*. `https://kaggle.com/`.

[103] J. Kaplan, S. McCandlish, T. Henighan, T. B Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, *Scaling laws for neural language models*, arXiv preprint arXiv:2001.08361 (2020).

[104] G.-r. Kim and H. M. Markowitz, *Investment rules, margin, and market volatility*, The Journal of Portfolio Management **16** (1989), no. 1, 45–52, available at `https://jpm.pm-research.com/content/16/1/45.full.pdf`.

[105] Y. Kim, S. Jeong, and I. Ghani, *Text Opinion Mining to Analyze News for Stock Market Prediction*, International journal of advances in soft computing and its applications **6** (2014), no. 1.

[106] A. Kliber and B. Bedowska-Sojka, *Realized Volatility Versus GARCH and Stochastic Volatility Models. The Evidence from the WIG20 index and the EUR/PLN Foreign Exchange Market*, Przegląd statystyczny/Statistical Review **4** (2010), 105–127.

[107] J. Knowlson, *Universal language schemes in England and France 1600-1800*, University of Toronto Press, 1975.

[108] P. Kokoszka and X. Zhang, *Functional prediction of intraday cumulative returns*, Statistical Modelling **12** (2012), no. 4, 377–398.

[109] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, *AlBERT: A lite BERT for Self-supervised Learning of Language Representations*, arXiv preprint arXiv:1909.11942 (2019).

[110] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, *BioBERT: a pre-trained biomedical language representation model for biomedical text mining*, Bioinformatics **36** (2020), no. 4, 1234–1240.

[111] S. S. Lee, *Jumps and Information Flow in Financial Markets*, The Review of Financial Studies **25** (2012), no. 2, 439–479.

[112] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, arXiv preprint arXiv:1910.13461 (2019).

[113] J. Li, Z. Xu, L. Yu, and L. Tang, *Forecasting oil price trends with sentiment of online news articles*, Procedia Computer Science **91** (2016), 1081–1087.

[114] T. Li, D. Shin, and B. Wang, *Cryptocurrency Pump-and-Dump Schemes*, SSRN Electronic Journal (2018).

[115] X. Li, L. Bing, W. Zhang, and W. Lam, *Exploiting BERT for end-to-end aspect-based sentiment analysis*, arXiv preprint arXiv:1910.00883 (2019).

[116] C. M Liapis and S. Kotsiantis, *Temporal Convolutional Networks and BERT-Based Multi-Label Emotion Analysis for Financial Forecasting*, Information **14** (2023), no. 11, 596.

[117] S.-M. Liu and B W. Brorsen, *Maximum likelihood estimation of a GARCH-stable model*, Journal of Applied Econometrics **10** (1995), no. 3, 273–285.

[118] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, arXiv preprint arXiv:1907.11692 (2019).

[119] B. Llacay and G. Peffer, *Using realistic trading strategies in an agent-based stock market model*, Computational and Mathematical Organization Theory **24** (2018), no. 3, 308–350.

[120] E. Loper and S. Bird, *NLTK: The Natural Language Toolkit*, arXiv preprint cs (2002).

[121] W. Louhichi, *Adjustment of stock prices to earnings announcements: evidence from Euronext Paris*, Review of Accounting and Finance **7** (2008), no. 1, 102–115.

[122] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model*, arXiv preprint arXiv:2211.02001 (2022).

[123] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler, *The accuracy of extrapolation (time series) methods: Results of a forecasting competition*, Journal of forecasting **1** (1982), no. 2, 111–153.

[124] H. Maqsood, I. Mehmood, M. Maqsood, M. Yasir, S. Afzal, F. Aadil, M. M. Selim, and K. Muhammad, *A local and global event sentiment based efficient stock exchange forecasting using deep learning*, International Journal of Information Management **50** (2020), 432–451.

[125] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).

[126] T. Mikolov, I. Sutskever, K. Chen, G. S Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, Advances in neural information processing systems **26** (2013).

[127] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heinz, and D. Roth, *Recent advances in natural language processing via large pre-trained language models: A survey*, arXiv preprint arXiv:2111.01243 (2021).

[128] M. L Mitchell and J H. Mulherin, *The impact of public information on the stock market*, The Journal of Finance **49** (1994), no. 3, 923–950.

[129] T. Mitchell, *Machine Learning*, McGraw Hill, 1997. `http://www.cs.cmu.edu/~tom/mlbook.html`.

[130] A. Mittal and A. Goel, *Stock prediction using twitter sentiment analysis*, Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf) **15** (2012), 2352.

[131] M. L. Morgia, A. Mei, F. Sassi, and J. Stefa, *The Doge of Wall Street: Analysis and Detection of Pump and Dump Cryptocurrency Manipulations*, ArXiv **abs/2105.00733** (2021).

[132] K. P. Murphy, *Probabilistic Machine Learning: An introduction*, MIT Press, 2022.

[133] P. Muthukumar and J. Zhong, *A stochastic time series model for predicting financial trends using NLP*, arXiv preprint arXiv:2102.01290 (2021).

[134] M. Nagao, *A framework of a mechanical translation between Japanese and English by analogy principle*, na, 1984.

[135] D. B Nelson, *Conditional heteroskedasticity in asset returns: A new approach*, Econometrica: Journal of the econometric society (1991), 347–370.

[136] I. K. Nti, A. F. Adekoya, and B. A. Weyori, *A systematic review of fundamental and technical analysis of stock market predictions*, Artificial Intelligence Review **53** (2020), no. 4, 3007–3057.

[137] M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, and G. Wills, *A Survey on Machine Learning for Stock Price Prediction: Algorithms and Techniques*, Proceedings of the 2nd international conference on finance, economics, management and it business, 2020.

[138] R. Olaniyan, D. Stamate, and I. Pu, *A Two-Step Optimised BERT-Based NLP Algorithm for Extracting Sentiment from Financial News*, Ifip international conference on artificial intelligence applications and innovations, 2021, pp. 745–756.

[139] Open AI, *Chat GPT*. `https://chat.openai.com/chat`.

[140] Open AI, *GPT 4*. `https://openai.com/gpt-4`.

[141] OpenAI and Team, *GPT-4 Technical Report*, 2023.

[142] J Osiewalski and M Pipieri, *Bayesian Forecasting of Exchange Rates Using GARCH Models with Skewed t Conditional Distributions, MACROMODELS'98*, Proceedings of the conference, jurata, 1998.

[143] M. Osińska, *Ekonometria finansowa*, PWE Polskie Wydawnictwo Ekonomiczne, 2005.

[144] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, *Sentiment analysis of Twitter data for predicting stock market movements*, 2016 international conference on signal processing, communication, power and embedded system (scopes), 2016, pp. 1345–1350.

[145] A. Pajor, *Wielowymiarowe procesy wariancji stochastycznej w ekonometrii finansowej: ujęcie bayesowskie*, Zeszyty Naukowe/Uniwersytet Ekonomiczny w Krakowie. Seria Specjalna, Monografie **195** (2010).

[146] Papers With Code, *Linguistics Acceptability on CoLA*, 2022. `https://paperswithcode.com/sota/linguistic-acceptability-on-cola`.

[147] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, Advances in neural infor-

mation processing systems 32, 2019, pp. 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[148] J. M. Patell and M. A. Wolfson, *The intraday speed of adjustment of stock prices to earnings and dividend announcements*, Journal of Financial Economics **13** (1984), no. 2, 223–252.

[149] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, *Carbon emissions and large neural network training*, arXiv preprint arXiv:2104.10350 (2021).

[150] M. S. Perlin, M. Mastella, D. F. Vancin, and H. P. Ramos, *A GARCH tutorial with R*, Revista de Administração Contemporânea **25** (2020), e200088.

[151] M. Pipień, *Wnioskowanie bayesowskie w ekonometrii finansowej*, Zeszyty Naukowe - Akademia Ekonomiczna w Krakowie: Monografie, Wydawn. Akad. Ekonomicznej w Krakowie, 2006.

[152] R. Rabemananjara and J.-M. Zakoian, *Threshold ARCH models and asymmetries in volatility*, Journal of applied econometrics **8** (1993), no. 1, 31–49.

[153] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., *Improving language understanding by generative pre-training* (2018).

[154] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., *Language models are unsupervised multitask learners*, OpenAI blog **1** (2019), no. 8, 9.

[155] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, Journal of Machine Learning Research **21** (2020), no. 140, 1–67.

[156] P. Rajpurkar, R. Jia, and P. Liang, *Know what you don't know: Unanswerable questions for SQuAD*, arXiv preprint arXiv:1806.03822 (2018).

[157] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, *Squad: 100,000+ questions for machine comprehension of text*, arXiv preprint arXiv:1606.05250 (2016).

[158] D. E Rumelhart, G. E Hinton, and R. J Williams, *Learning internal representations by error propagation*, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[159] E. Samanidou, E. Zschischang, D. Stauffer, and T. Lux, *Agent-based Models of Financial Markets*, Reports on Progress in Physics **70** (2007), 409–450.

[160] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter*, arXiv preprint arXiv:1910.01108 (2019).

[161] R. E. Schapire, *The strength of weak learnability*, 30th annual symposium on foundations of computer science, 1989, pp. 28–33.

[162] R. P. Schumaker and H. Chen, *A quantitative stock prediction system based on financial news*, Information Processing & Management **45** (2009), no. 5, 571 –583.

[163] V. Shankar and S. Parsana, *An overview and empirical comparison of natural language processing (NLP) models and an introduction to and empirical application of autoencoder models in marketing*, Journal of the Academy of Marketing Science **50** (2022), no. 6, 1324–1350.

[164] S. Singh, *Natural language processing for information extraction*, arXiv preprint arXiv:1807.02383 (2018).

[165] J. Socha, *Rynek papierów wartościowych w Polsce*, Olympus, 2003.

[166] P. Soni, Y. Tewari, and D. Krishnan, *Machine Learning Approaches in Stock Price Prediction: A Systematic Review*, Journal of physics: Conference series, 2022, pp. 012065.

[167] A. Sopoćko, *Rynkowe instrumenty finansowe*, Wydawnictwo Naukowe PWN, 2005.

[168] F. Souza, R. Nogueira, and R. Lotufo, *Portuguese named entity recognition using BERT-CRF*, arXiv preprint arXiv:1909.10649 (2019).

[169] T. T. P. Souza and T. Aste, *Predicting future stock market structure by combining social and financial network information*, Physica A: Statistical Mechanics and its Applications **535** (2019), 122343.

[170] P. J Stone, D. C Dunphy, and M. S Smith, *The general inquirer: A computer approach to content analysis.* (1966).

[171] A. Sławiński, *Rynki finansowe*, Polskie Wydawnictwo Ekonomiczne, 2006.

[172] R. Tadeusiewicz, *Sieci neuronowe*, Vol. 110, Akademicka Oficyna Wydawnicza RM Warszawa, 1993.

[173] R. Tadeusiewicz and M. Szaleniec, *Leksykon sieci neuronowych*, Projekt Nauka. Fundacja na rzecz promocji nauki polskiej, 2015.

[174] P. Talar, *A new GARCH process with hyperbolic noise*, Przegląd Statystyczny **2** (2007), no. 54, 28–46.

[175] P. C Tetlock, M. Saar-Tsechansky, and S. Macskassy, *More than words: Quantifying language to measure firms' fundamentals*, The journal of finance **63** (2008), no. 3, 1437–1467.

[176] The pandas development team, *pandas-dev/pandas: Pandas*, Zenodo, 2020. `https://pandas.pydata.org`.

[177] J. Thomas and K. Sycara, *Integrating Genetic Algorithms and Text Learning for Financial Prediction*, Proceedings of the GECCO-2000 Workshop on Data Mining with Evolutionary Algorithms (2000).

[178] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, *Well-Read Students Learn Better: On the Importance of Pre-training Compact Models*, arXiv preprint arXiv:1908.08962v2 (2019).

[179] A. M Turing and J Haugeland, *Computing machinery and intelligence*, The Turing Test: Verbal Behavior as the Hallmark of Intelligence (1950), 29–56.

[180] P. D Turney, *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*, arXiv preprint cs (2002).

[181] G. Van Rossum and F. L Drake Jr, *Python tutorial*, Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[182] S. Vanfossan, C. H Dagli, and B. Kwasa, *An agent-based approach to artificial stock market modeling*, Procedia Computer Science **168** (2020), 161–169.

[183] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, Ł u. Kaiser, and I. Polosukhin, *Attention is All You Need*, Proceedings of the 31st international conference on neural information processing systems, 2017, pp. 6000–6010.

[184] I. Venezia (ed.), *Behavioral Finance:A Novel Approach*, World Scientific Publishing Co. Pte. Ltd., 2020.

[185] N. Vo and R. Ślepaczuk, *Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on SP500 Index*, Entropy **24** (2022), no. 2.

[186] P. P. Vora, *Clogged information flow and stock-market sluggishness*, PLOS ONE **15** (2020), 1–13.

[187] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, *Superglue: A stickier benchmark for general-purpose language understanding systems*, Advances in neural information processing systems **32** (2019).

[188] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R Bowman, *GLUE: A multi-task benchmark and analysis platform for natural language understanding*, arXiv preprint arXiv:1804.07461 (2018).

[189] A. Warstadt, A. Singh, and S. R Bowman, *Neural Network Acceptability Judgments*, arXiv preprint arXiv:1805.12471 (2018).

[190] W. Weaver, *Translation*, Proceedings of the conference on mechanical translation, 1952.

[191] J. J Webster and C. Kit, *Tokenization as the initial phase in NLP*, Coling 1992 volume 4: The 14th international conference on computational linguistics, 1992.

[192] K. Weiss, T. M Khoshgoftaar, and D. Wang, *A survey of transfer learning*, Journal of Big data **3** (2016), no. 1, 1–40.

[193] J. Weizenbaum, *Computer power and human reason: From judgment to calculation.* (1976).

[194] W. Welfe, *Gospodarka oparta na wiedzy*, Polskie Wydawnictwo Ekonomiczne, 2007.

[195] R. Weron, *Electricity price forecasting: A review of the state-of-the-art with a look into the future*, International journal of forecasting **30** (2014), no. 4, 1030–1081.

[196] S. Wheelwright, S. Makridakis, and R. J Hyndman, *Forecasting: methods and applications*, John Wiley & Sons, 1998.

[197] H. White, *A reality check for data snooping*, Econometrica **68** (2000), no. 5, 1097–1126.

[198] Wikimedia Foundation, *Wikipedia Wolna Encyklopedia*. `https://www.wikipedia.org/`.

[199] X. Wu, *Event-Driven Learning of Systematic Behaviours in Stock Markets*, arXiv preprint arXiv:2010.15586 (2020).

[200] B. Wuthrich, V. Cho, S. Leung, D. Permunetilleke, K. Sankaran, and J. Zhang, *Daily stock market forecast from textual web data*, SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218), 1998, pp. 2720–2725 vol.3.

[201] Y. Yang, M. C. S. Uy, and A. Huang, *FinBERT: A pretrained language model for financial communications*, arXiv preprint arXiv:2006.08097 (2020).

[202] A. Zeliaś, B. Pawełek, and S. Wanat, *Prognozowanie ekonomiczne: teoria, przykłady, zadania*, Wydawnictwo Naukowe PWN, 2013.

[203] J. E. Zini and M. Awad, *On the explainability of natural language processing deep models*, ACM Computing Surveys **55** (2022), no. 5, 1–31.

[204] M. Zolfaghari and S. Gholami, *A hybrid approach of adaptive wavelet transform, long short-term memory and ARIMA-GARCH family models for the stock index prediction*, Expert Systems with Applications **182** (2021), 115149.

Aneks

A.1. Autorskie kody źródłowe programów

A.1.1. Pobieranie treści wiadomości finansowych

Poniższa klasa w języku Python służy do pobierania nagłówków oraz treści wiadomości finansowych z serwisu CityFalcon [27].

```
import pandas as pd
from datetime import datetime, timedelta
from readability import Document
import requests
import re
import json
import random
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support import ui
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.keys import Keys
import ast

def cleanhtml(raw_html):
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, '', raw_html)
    return cleantext

def get_article(url):
    try:
```

```
        headers = {
            'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X '
            '10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) '
            'Chrome/56.0.2924.87 Safari/537.36',
        }
        response = requests.get(url,headers=headers, timeout=5)
        doc = Document(response.text)

        return cleanhtml(doc.summary()).strip()
    except:
        return "SCRAPER ERROR - Some HTML error"

def get_js_articles(js_articles_df):
    driver = webdriver.Safari(
        executable_path = '/usr/bin/safaridriver')

    processed_rows = []
    for i in range(len(js_articles_df)):
        row = js_articles_df.iloc[i]
        source_name = row["source_name"]
        url = row["url"]

        driver.get(url)

        try:
            if source_name == "cnn.com":
                accept_all= ui.WebDriverWait(driver,10).until(
                    EC.element_to_be_clickable((By.XPATH,
                        '//*[@id="onetrust-accept-btn-handler"]')))
            elif source_name == "investors.com":
                accept_all= ui.WebDriverWait(driver,10).until(
                    EC.element_to_be_clickable((By.ID,
                        "gdpr-accept-button")))
            else:
                print("BAD SOURCE NAME")

            accept_all.click()
        except:
            pass

        time.sleep(random.randint(3,4))
```

```
        article_content = driver.page_source

        doc = Document(article_content)
        clean_page = cleanhtml(doc.summary()).strip()
        row["full_article"] = clean_page
        processed_rows.append(row)

    driver.delete_all_cookies()
    driver.quit()
    del driver

    js_articles_processed_df = pd.DataFrame(processed_rows)

    return js_articles_processed_df

def get_sa_articles(sa_articles_df, max_num_retries = 8,
    verbose = False, skip=True):

    if skip:
        clean_page = f'SCRAPER ERROR - Skipping Seeking Alpha'

        pass_df = sa_articles_df.copy()
        pass_df["full_article"] = clean_page
        return pass_df

    sa_articles_df = sa_articles_df.sort_values("uuid",
        ascending = True)

    log_pass = [('d2020@student.uek.krakow.pl', '1qA2wS3eD4rF'),
                ('bednarrski@gmail.com', '2wS3eD4rF5tG'),
                ("piotr.exmachina@gmail.com", "3eD4rF5tG6yH"),
                ("piotr@dodo.work", "4rF5tG6yH7uJ")]

    log_pass_idx = random.randint(0,len(log_pass)-1)
    login, password = log_pass[log_pass_idx]

    url_login = 'https://seekingalpha.com/login'

    login_xpath    = '//*[@id="root"]/div[1]/main/div[2]/form/div/'
    'div[1]/label/div/input'
```

```
password_xpath = '//*[@id="signInPasswordField"]'

signin_button_xpath = '//*[@id="root"]/div[1]/main/div[2]/form/'
'button'

start_signin_xpath = '/html/body/div[2]/div[2]/div/div[2]/'\
    'section/div/div[2]/div/button'


driver = webdriver.Safari(
    executable_path = '/usr/bin/safaridriver')
print(f'Log in as {login}')
driver.get(url_login)

login_box = ui.WebDriverWait(driver,10).until(
    EC.element_to_be_clickable((By.XPATH, login_xpath)))
login_box.send_keys(login)

password_box = ui.WebDriverWait(driver,10).until(
    EC.element_to_be_clickable((By.XPATH, password_xpath)))
password_box.send_keys(password)
password_box.send_keys(Keys.ENTER)

time.sleep(random.randint(5,7))

num_relogins = 0
processed_rows = []
for i in range(len(sa_articles_df)):
    row = sa_articles_df.iloc[i]

    if not skip:
        url = row["url"]

        driver.get(url)
        time.sleep(random.randint(3,5))

        try:
            article_content = driver.find_element(By.XPATH,
                "//*[@data-test-id='article-content']")
            doc = Document(
```

```
                article_content.get_attribute('innerHTML'))
            clean_page = cleanhtml(doc.summary()).strip()
        except:

            for j in range(max_num_retries):
                if j > 4:
                    print(f'\t{j}, {url}')
                driver.delete_all_cookies()
                driver.quit()
                del driver
                time.sleep(6*j+random.randint(6,9))

                log_pass_idx = (log_pass_idx + 1)%len(log_pass)
                login, password = log_pass[log_pass_idx]
                driver = webdriver.Safari(
                    executable_path = '/usr/bin/safaridriver')
                if verbose:
                    print(f'Re-log in as {login}')
                num_relogins += 1
                driver.get(url_login)

                try:
                    login_box = ui.WebDriverWait(driver,10)\
                        .until(EC.element_to_be_clickable(
                            (By.XPATH, login_xpath)))
                    login_box.send_keys(login)

                    password_box = ui.WebDriverWait(driver,10)\
                        .until(EC.element_to_be_clickable(
                            (By.XPATH, password_xpath)))
                    password_box.send_keys(password)
                    password_box.send_keys(Keys.ENTER)

                    time.sleep(random.randint(3,5))

                    driver.get(url)
                    time.sleep(random.randint(3,5))
                    try:
                        article_content = driver.find_element(
                            By.XPATH,
                            "//*[@data-test-id='article-content']"
```

```
                            )
                            doc = Document(article_content\
                                .get_attribute('innerHTML'))
                            clean_page = cleanhtml(doc.summary())\
                                .strip()
                            success = True
                            break
                        except:
                            success = False
                    except:
                        success = False

            if success == False:
                clean_page = f'SCRAPER ERROR - Error after '\
                    f'{max_num_retries+1} retries on Seeking Alpha'
                print(f'\tError after {max_num_retries+1} retries'
                    f'with URL:\n\t\t{url}')

        else:
            clean_page = f'SCRAPER ERROR - Skipping Seeking Alpha'

        row["full_article"] = clean_page
        processed_rows.append(row)

    driver.quit()
    del driver

    print(f'For SeekingAlpha {num_relogins} re-logins were needed.')
    sa_articles_processed_df = pd.DataFrame(processed_rows)

    return sa_articles_processed_df


class NewsGetter():
    def __init__(self, config):
        self._config = config["news_getter"]
        self.news = pd.DataFrame()

    def get_news(self, date_start = "beginning", date_end = "now",
        symbols = [], verbose = False):
        source = self._config["source"]
```

```
        if date_end == "now":
            date_end = datetime.utcnow().strftime('%Y-%m-%d-%H:%M:%S')
        if date_start == "beginning":
            date_start = "1970-01-01-00:00:00"

        if source == "yahoo_scraper":

            self.news = self.load_yahoo_scraped_news(
                date_start, date_end, symbols, verbose = verbose)

        if source == "city_falcon_api":

            self.news = self.load_city_falcon_api_news(
                date_start, date_end, symbols, verbose = verbose)

        if source == "city_falcon_scraper":

            self.news = self.city_falcon_scraped_news(
                date_start, date_end, symbols, verbose = verbose)

        else:
            if verbose:
                print("No such news data source.")

    def load_city_falcon_api_news(self, date_start, date_end, symbols,
        verbose = False):

        get_full_articles = self._config["get_full_articles"]
        symbols_search = self._config["symbols_search"]
        city_falcon_api_key = self._config["city_falcon_api_key"]
        city_falcon_timeframe = self._config["city_falcon_timeframe"]

        if len(symbols) == 0:
            symbols = ["AAPL", "IBM", "TSLA", "AMZN"]

        identifier_type = "full_tickers"

        symbols_fixed = []
        for symbol in symbols:
            if symbol[0] == "#":
```

```
            symbol = symbol[1:]
            identifier_type = 'assets'
        else:
            symbol = symbol+"_US"
        symbols_fixed.append(symbol)

    url = 'https://api.cityfalcon.com/v0.2/stories?'+\
          f'identifier_type={identifier_type}&'+\
          f'identifiers={",".join(symbols_fixed)}&'+\
          'categories=mp&'+\
          'min_cityfalcon_score=20&'+\
          'order_by=latest&'+\
          'languages=en&'+\
          f'access_token={city_falcon_api_key}&'+\
          f'time_filter={city_falcon_timeframe}'


    if verbose:
        print(f'URL:\n{url}\n')

    response = requests.get(url)
    articles_dict = json.loads(response.text)

    try:
        old_news_df = pd.DataFrame(articles_dict['stories'])
    except:
        print("URL:", url)
        print("Symbols:", symbols)
        print(response.text)
        old_news_df = pd.DataFrame([])

    if verbose:
        print(f'There are {len(old_news_df)} news items'
            'loaded from City Falcon API.')

    if len(old_news_df) == 0:
        return old_news_df

    old_news_df["published"] = pd.to_datetime(
        old_news_df["publishTime"]).apply(
        lambda x: x.strftime('%Y-%m-%d-%H:%M:%S'))
```

```
        old_news_df = old_news_df[
            (old_news_df["published"] > date_start) &\
            (old_news_df["published"] <= date_end)]

        del old_news_df["publishTime"]

        if verbose:
            print(f'There are {len(old_news_df)} news items left'
                'after filtering.')

        return old_news_df

    def city_falcon_scraped_news(self, datetime_start_str,
        datetime_end_str, symbols, verbose = False):
        if datetime_start_str == "beginning":
            datetime_start_str = "1970-01-01-00:00:00"
        if datetime_end_str == "now":
            datetime_end_str = datetime.utcnow()\
            .strftime('%Y-%m-%d-%H:%M:%S')

        date_start = datetime.strptime(datetime_start_str,
            '%Y-%m-%d-%H:%M:%S').date()
        date_end = datetime.strptime(datetime_end_str,
            '%Y-%m-%d-%H:%M:%S').date()

        date_list = [
            (date_start + timedelta(days=x)).strftime("%Y-%m-%d")\
            for x in range(0, (date_end-date_start).days+1)]

        symbols_fixed = []
        for symbol in symbols:
            if symbol[0] == "#":
                symbols_fixed.append(symbol[1:])
            else:
                symbols_fixed.append(symbol+"_US")

        # Get the headlines and URLs

        def any_symbol_in_search_tags(search_tags, symbols):
            search_tags = ast.literal_eval(search_tags)
            for search_tag in search_tags:
```

```
                for symbol in symbols:
                    if search_tag == symbol:
                        return True
            return False

        temp_df_list = []
        for date in date_list:
            file_path = self._config['city_falcon_headlines_path'] +\
                f'{date[0:4]}/news_{date}.csv'

            temp_df = pd.read_csv(file_path)

            temp_df = temp_df[temp_df["search_tags"]\
                .apply(lambda search_tags : any_symbol_in_search_tags(
                    search_tags, symbols_fixed))]
            temp_df_list.append(temp_df)

        old_news_df = pd.concat(temp_df_list)
        del old_news_df['Unnamed: 0']

        old_news_df = old_news_df[
            (old_news_df["published_utc"] > datetime_start_str) &\
            (old_news_df["published_utc"] <= datetime_end_str)]
        old_news_df["full_article"] = ''
        old_news_df["description"] = old_news_df["description"]\
            .fillna('')

        if self._config["get_full_articles"]:
            corrupted_dates = []
            for date in date_list:
                try:
                    file_path = \
                        self._config['city_falcon_full_news_path'] +\
                        f'{date[0:4]}/full_articles_{date}.csv'

                    temp_df = pd.read_csv(file_path, sep=';')

                    old_news_df = pd.merge(old_news_df, temp_df,
                        how="left", on="uuid",
                        suffixes=('', '_added'))
                    old_news_df["full_article_added"] =\
```

```
                    old_news_df["full_article_added"].fillna('')
                    old_news_df["full_article"] =\
                        old_news_df["full_article"] +\
                        old_news_df["full_article_added"]
                    del old_news_df["full_article_added"]
                except:
                    corrupted_dates.append(date)

                if len(corrupted_dates) > 0:
                    print(f'\tThere were {len(corrupted_dates)}\
                        corrupted news files for dates:\
                        \n\t\t{corrupted_dates}')

        return old_news_df

    def load_yahoo_scraped_news(self, date_start, date_end, symbols,
        verbose = False):
        file_path = self._config["yahoo_file_path"]
        earliest_parsed_date = self._config["yahoo_earliest"]
        get_full_articles = self._config["get_full_articles"]
        symbols_search = self._config["symbols_search"]

        old_news_df = pd.read_csv(file_path)
        del old_news_df["Unnamed: 0"]

        old_news_df = old_news_df[old_news_df["title"] != "title"]
        old_news_df = old_news_df[
            old_news_df["parsed_datetime"] > earliest_parsed_date]

        earliest_df = old_news_df.groupby("link")\
            .agg({"published" : min}).reset_index()
        old_news_df = earliest_df.merge(old_news_df, how="inner")


        if verbose:
            print(f'There are {len(old_news_df)} news items loaded\
                from Yahoo scraped file.')

        old_news_df = old_news_df[
            (old_news_df["published"] > date_start) &\
            (old_news_df["published"] <= date_end)]
```

```
        old_news_df["website"] = old_news_df["link"].apply(
            lambda x : (x.split("/"))[2])

        if get_full_articles:
            old_news_df["full_article"] = old_news_df["link"]\
            .apply(get_article)

        if len(symbols) > 0:
            symbols_keywords = self.get_symbols_keywords(symbols)
            if verbose:
                print(f'Keywords used for search: {symbols_keywords}')

            if symbols_search == "title":
                old_news_df = old_news_df[old_news_df["title"]\
                    .apply(lambda x : any(keyword in x\
                        for keyword in symbols_keywords))]

            if get_full_articles:
                if symbols_search == "full_article":
                    old_news_df = old_news_df[
                        old_news_df["full_article"].apply(lambda x :\
                            any(keyword in x for keyword in\
                                symbols_keywords))]

                elif symbols_search == "everywhere":
                    old_news_df = old_news_df[
                        (old_news_df["title"].apply(lambda x :\
                            any(keyword in x for\
                                keyword in symbols_keywords))) |
                        (old_news_df["full_article"].apply(lambda x :\
                            any(keyword in x for\
                                keyword in symbols_keywords)))]

            else:
                if (symbols_search == "full_article") |\
                    (symbols_search == "everywhere"):
                    if verbose:
                        print("You need to set get_full_articles to\
                            True to search in full articles.")

        old_news_df = old_news_df.rename(columns={"link" : "url"})
```

```
        if verbose:
            print(f'There are {len(old_news_df)} news items left\
                after filtering.')

        return old_news_df

    def get_symbols_keywords(self, symbols, verbose = False):
        keywords = []
        for symbol in symbols:
            temp_keywords = [symbol, symbol.lower()]

            if symbol.upper() in SYMBOLS_TEMP_DICT:
                temp_keywords = temp_keywords +\
                    SYMBOLS_TEMP_DICT[symbol.upper()]

            keywords = keywords + temp_keywords

        keywords = list(set(keywords))
        return keywords
```

A.1.2. Pobieranie historycznych notowań giełdowych

Klasa *StocksGetter* została stworzona do pobierania historycznych notowań giełdowych i łączenia ich z pobranymi wiadomościami w sposób zapewniający odpowiednią chronologię. Dane pobierane są z serwisu Alpha Vantage [3].

```
import pandas as pd
from datetime import datetime
import requests
import json
from io import StringIO
from data.news_getter import NewsGetter
import time


class StocksGetter():
    def __init__(self, config):
        self._config = config["stocks_getter"]
        self.quotes = pd.DataFrame()
        self.news = pd.DataFrame()
        self.symbol = ""
        self.metadata = {}


    def get_historical_quotes_from_file(self, symbol,
        date_start = "beginning", date_end = "now"):
        self.symbol = symbol
        if date_end == "now":
            date_end = datetime.utcnow().strftime('%Y-%m-%d-%H:%M:%S')
        if date_start == "beginning":
            date_start = "1970-01-01-00:00:00"

        filepath = f'{self._config["stocks_dir"]}{self.symbol}_\
            {self._config["interval"]}.csv'
        print(f'Loading data from file: {filepath}')
        raw_df = pd.read_csv(filepath)
        del raw_df["Unnamed: 0"]

        self.metadata = {
            "2. Symbol": symbol,
            "4. Interval": self._config["interval"],
            "6. Time Zone": self._config["forced_timezone"]
```

```
        }
        print(f'Metadata loaded from job config:\n{self.metadata}')

        raw_df = raw_df[
            (pd.to_datetime(raw_df["time"]) <=\
                pd.to_datetime(date_end)) & \
            (pd.to_datetime(raw_df["time"]) >=\
                pd.to_datetime(date_start))]

        raw_df["time_dt"] = pd.to_datetime(raw_df["time"])
        raw_df.set_index("time_dt")
        raw_df["time_utc"] = raw_df["time_dt"]\
            .dt.tz_localize(tz=self.metadata['6. Time Zone'])\
            .dt.tz_convert("UTC").apply(lambda x: x.strftime(
                '%Y-%m-%d-%H:%M:%S'))
        del raw_df["time_dt"]

        raw_df["date"] = pd.to_datetime(raw_df["time"])\
            .apply(lambda x: x.strftime('%Y-%m-%d'))

        self.quotes = raw_df


    def get_historical_quotes(self, symbol, date_start = "beginning",
        date_end = "now", get_metadata = True):
        self.symbol = symbol
        if date_end == "now":
            date_end = datetime.utcnow().strftime('%Y-%m-%d-%H:%M:%S')
        if date_start == "beginning":
            date_start = "1970-01-01-00:00:00"

        api_key = self._config["av_api_key"]
        interval = self._config["interval"]
        adjusted = self._config["adjusted"]

        function = "TIME_SERIES_INTRADAY_EXTENDED"
        output_size = "full"
        data_type = "csv"

        if get_metadata:
            print("Getting Metadata from Alpha Vantage.\nWARNING:\
```

```
            This is using one additional API call!")
        # Load JSON to get metadata
        url = f'https://www.alphavantage.co/query?'+\
        f'function=TIME_SERIES_INTRADAY'+\
        f'&symbol={self.symbol.replace(".", "-")}'+\
        f'&interval={interval}'+\
        f'&outputsize=compact'+\
        f'&apikey={api_key}'+\
        f'&datatype=json'

        time.sleep(12)
        response = requests.get(url)
        response_dict = json.loads(response.text)

        if "Meta Data" in response_dict:
            self.metadata = response_dict["Meta Data"]
        else:
            success = False
            counter = 0
            while not success and counter < 10:
                counter += 1
                time.sleep(12)
                response = requests.get(url)
                response_dict = json.loads(response.text)
                if "Meta Data" in response_dict:
                    self.metadata = response_dict["Meta Data"]
                    success = True
                else:
                    print(response.text)
                    print(counter, url)
    else:
        self.metadata = {
            "2. Symbol": symbol,
            "4. Interval": interval,
            "5. Output Size": output_size,
            "6. Time Zone": self._config["forced_timezone"]
        }
        print(f'Metadata from job config:\n{self.metadata}')

    # Load actual data, iterating over API responses
    time_delta = datetime.utcnow() - datetime.strptime(date_start,
```

```
        '%Y-%m-%d-%H:%M:%S')
    months = int(time_delta.days/30)+1

    if months > 24:
        months = 24
    if months < 1:
        months = 1
    slices = ["year"+str(int(month/12)+1)+"month"+str(
        month%12+1) for month in range(months)]

    temp_df_list = []
    print(f'Parsing {len(slices)} slices for {symbol}:')
    for s in reversed(slices):
        print(f'Getting {s}...')
        url = f'https://www.alphavantage.co/query?'+\
        f'function={function}'+\
        f'&symbol={self.symbol.replace(".", "-")}'+\
        f'&interval={interval}'+\
        f'&slice={s}'+\
        f'&adjusted={adjusted}'+\
        f'&apikey={api_key}'

        time.sleep(12)
        response = requests.get(url)
        raw_data = StringIO(response.text)
        temp_df = pd.read_csv(raw_data, sep=",")

        if "time" in list(temp_df.columns):
            temp_df_list.append(temp_df)
        else:
            success = False
            counter = 0
            while not success and counter < 10:
                counter += 1
                time.sleep(12)
                raw_data = StringIO(response.text)
                temp_df = pd.read_csv(raw_data, sep=",")
                if "time" in list(temp_df.columns):
                    temp_df_list.append(temp_df)
                    success = True
                else:
```

```
                    print(response.text)
                    print(counter, url)

        if len(temp_df_list) > 0:
            raw_df = pd.concat(temp_df_list)


            raw_df = raw_df[
                (pd.to_datetime(raw_df["time"]) <=\
                    pd.to_datetime(date_end)) & \
                (pd.to_datetime(raw_df["time"]) >=\
                    pd.to_datetime(date_start))]

            raw_df["time_dt"] = pd.to_datetime(raw_df["time"])
            raw_df.set_index("time_dt")
            raw_df["time_utc"] = raw_df["time_dt"]\
                .dt.tz_localize(tz=self.metadata['6. Time Zone'])\
                .dt.tz_convert("UTC").apply(lambda x: x.strftime(
                    '%Y-%m-%d-%H:%M:%S'))
            del raw_df["time_dt"]

            raw_df["date"] = pd.to_datetime(raw_df["time"])\
            .apply(lambda x: x.strftime('%Y-%m-%d'))

            self.quotes = raw_df
        else:
            self.quotes = pd.DataFrame()

    def get_recent_quotes(self, symbol):
        self.symbol = symbol

        api_key = self._config["av_api_key"]
        interval = self._config["interval"]
        adjusted = self._config["adjusted"]

        function = "TIME_SERIES_INTRADAY"
        output_size = "full"
        data_type = "csv"

        # Load JSON to get metadata
        url = f'https://www.alphavantage.co/query?'+\
```

```
        f'function=TIME_SERIES_INTRADAY'+\
        f'&symbol={self.symbol.replace(".", "-")}'+\
        f'&interval={interval}'+\
        f'&outputsize=compact'+\
        f'&apikey={api_key}'+\
        f'&datatype=json'

        response = requests.get(url)
        self.metadata = json.loads(response.text)["Meta Data"]

        # Load actual data
        url = f'https://www.alphavantage.co/query?'+\
        f'function={function}'+\
        f'&symbol={self.symbol.replace(".", "-")}'+\
        f'&interval={interval}'+\
        f'&adjusted={adjusted}'+\
        f'&outputsize={output_size}'+\
        f'&apikey={api_key}'+\
        f'&datatype={data_type}'

        response = requests.get(url)

        raw_data = StringIO(response.text)

        raw_df = pd.read_csv(raw_data, sep=",")

        self.quotes = raw_df


    def augment_with_news(self, news_getter,
        adjusting_method="nearest", market_open_only = False):
        date_min = self.quotes["time_utc"].min()
        date_max = self.quotes["time_utc"].max()

        print(date_min)
        print(date_max)

        if market_open_only:
            self.limit_to_market_hours()

        news_getter.get_news(
```

```
            date_min, date_max, symbols=[self.symbol])

        temp_news = news_getter.news

        temp_news["published_dt"] = pd.to_datetime(
            temp_news["published_utc"])
        self.quotes["time_utc_dt"] = pd.to_datetime(
            self.quotes["time_utc"])

        temp_news.sort_values('published_dt', inplace=True)
        self.quotes.sort_values('time_utc_dt', inplace=True)

        temp = self.quotes.set_index('time_utc_dt')\
            .reindex(temp_news.set_index('published_dt').index,
                method=adjusting_method).reset_index()
        adjusted_new = pd.merge(temp_news, temp, on='published_dt')\
            [list(temp_news.columns)+["time_utc"]]\
            .rename(columns = {"time_utc" : "published_adjusted_utc"})

        del self.quotes["time_utc_dt"]
        del adjusted_new["published_dt"]

        self.news = adjusted_new

    def limit_to_market_hours(self):
        market_opening_time = self._config["market_hours"][0]
        market_closing_time = self._config["market_hours"][1]

        self.quotes["time_dt"] = pd.to_datetime(self.quotes["time"])
        self.quotes = self.quotes.set_index("time_dt")
        self.quotes = self.quotes.between_time(market_opening_time,
            market_closing_time).reset_index()

        del self.quotes["time_dt"]
```

A.1.3. Generowanie zmiennych modelowanych

Klasa *FeatureGenerator* jest używana do przygotowania surowych danych do dalszego modelowania wykorzystując klasy *NewsGetter* oraz *StocksGetter*.

```
from data.news_getter import NewsGetter
from data.stocks_getter import StocksGetter
from datetime import timedelta
import numpy as np
import pandas as pd
import re

def make_substitutes(article, substitutes_dict, order=None):
    categories = substitutes_dict.keys()

    order_set = set(order)
    categories_set = set(categories)
    remaining_categories = list(categories_set.difference(order_set))

    ordered_categories = order + remaining_categories

    for category in ordered_categories:
        for named_entity_pattern in substitutes_dict[category]:
            article = re.sub(named_entity_pattern, category, article)

    return article

def backward_then_forward(arr):
    df = pd.DataFrame(arr)
    df.fillna(method='ffill', axis=0, inplace = True)
    df.fillna(method='bfill', axis=0, inplace = True)
    out = df[0].values
    return out

class FeatureGenerator():
    def __init__(self, config):
        self._config = config
        self._sg_ng_config = {
                "news_getter" : {
                    "source"                    : "city_falcon_scraper",
                    "yahoo_file_path"                   : "",
                    "city_falcon_headlines_path"        : "",
```

```
                "city_falcon_full_news_path"      : "",
                "yahoo_file_path"                 : "",
                "yahoo_earliest"                  : "",
                "city_falcon_api_key"             : "",
                "city_falcon_timeframe"           : "w1",
                "symbols_search"                  : "everywhere",
                "get_full_articles"               : True,
            },
            "stocks_getter" : {
                "interval"              : self._config["interval"],
                "adjusted"              : "true",
                "av_api_key"            : "",
                "forced_timezone"       : "US/Eastern",
                "market_hours"          : ("09:30:00", "16:05:00"),
                "stocks_dir"            : ""
            },
        }
    self.quotes_news = pd.DataFrame()
    self.sg = StocksGetter(self._sg_ng_config)
    self.ng = NewsGetter(self._sg_ng_config)
    self.corrupted_symbols_list = []

def load_all_symbols(self):
    for i, symbol in enumerate(self._config["symbols"]):
        print(f'\nSymbol: {i}/{len(self._config["symbols"])}')
        symbol_data_with_news =\
            self.load_symbol_data_with_news(symbol)
        if symbol_data_with_news is not None:
            symbol_data_with_news[
                symbol_data_with_news["volume"] < 0.0 ] = np.nan
            symbol_data_with_news_grouped =\
                symbol_data_with_news.groupby("time_utc").agg({
                    "time" : "min",
                    "open" : "min",
                    "high" : "min",
                    "low" : "min",
                    "close" : "min",
                    "volume" : "min",
                    "date" : "min",
                    "uuid" : list,
                    "title" : list,
```

```
                    "description" : list,
                    "asset_tags" : list,
                    "search_tags" : list,
                    "url" : list,
                    "source_name" : list,
                    "published_utc" : list,
                    "date_utc" : min,
                    "full_article" : list,
                    "published_adjusted_utc" : "count",
                    "symbol" : min,
                }).reset_index().sort_values("time_utc").rename(
                columns={"published_adjusted_utc":"article_count"})

                distinct_dates = list(symbol_data_with_news["date"]\
                    .drop_duplicates())

                for date in distinct_dates:
                    single_date_df =self.generate_timeseries_features(
                        symbol_data_with_news_grouped, date)
                    self.quotes_news = pd.concat(
                        [self.quotes_news, single_date_df])

        if self._config["timeseries_fill_na"] ==\
            "backward_then_forward":
                for feature in self._config["timeseries_features"]:
                    self.quotes_news[feature+"_past"] =\
                        self.quotes_news[feature+"_past"]\
                            .apply(backward_then_forward)

        print("Corrupted symbols in this run: ",
            self.corrupted_symbols_list)

    def load_symbol_data_with_news(self, symbol):
        if self._config["load_quotes_from_files"]:
            try:
                self.sg.get_historical_quotes_from_file(symbol,
                    date_start=self._config["date_start"]+"-00:00:00",
                    date_end = self._config["date_end"]+"-23:59:59")
            except:
                print("\tFile does not exist.")
                return None
```

```
        else:
            self.sg.get_historical_quotes(symbol,
                date_start = self._config["date_start"]+"-00:00:00",
                date_end = self._config["date_end"]+"-23:59:59")

        print("\tLen quotes:", len(self.sg.quotes))

        if len(self.sg.quotes) > 0:
            self.sg.augment_with_news(self.ng,
                market_open_only=self._config["limit_to_market_open"])
            print("\tLen news:", len(self.sg.news))

            merged_df = pd.merge(self.sg.quotes, self.sg.news,
                how='left', left_on = "time_utc",
                right_on="published_adjusted_utc")
            merged_df["symbol"] = symbol
            return merged_df

        else:
            print("\tLen news:", 0)
            return None

    def time_window(self, time_utc, col_name, full_time_df,
        window_len_mins, interval_mins):
        backward = (window_len_mins < 0)
        window_len_mins = abs(window_len_mins)

        expected_length = window_len_mins//interval_mins

        if backward:
            window_values = full_time_df[
                (full_time_df["time_utc"] >=\
                    time_utc-timedelta(minutes=window_len_mins)) &\
                (full_time_df["time_utc"] <=\
                    time_utc)][col_name].values
            expected_length += 1
            window_values = np.pad(window_values,
                (expected_length-len(window_values),0),
                'constant', constant_values=(np.nan))

        else:
```

```
            window_values = full_time_df[
                (full_time_df["time_utc"] > time_utc) &\
                (full_time_df["time_utc"] <= time_utc+\
                    timedelta(minutes=window_len_mins))
            ][col_name].values

            window_values = np.pad(window_values,
                (0, expected_length-len(window_values)),
                'constant', constant_values=(np.nan))

        return window_values

    def generate_timeseries_features(self,
        symbol_data_with_news_grouped, date):
        one_day_df = symbol_data_with_news_grouped[
            symbol_data_with_news_grouped["date"] == date]

        idx = pd.date_range(one_day_df.time_utc.min(),
            one_day_df.time_utc.max(), freq=self._config["interval"])

        one_day_df.index =\
            pd.DatetimeIndex(pd.to_datetime(one_day_df.time_utc))

        one_day_df = one_day_df.reindex(idx).reset_index()
        del one_day_df["time_utc"]
        one_day_df = one_day_df.rename(columns={"index" : "time_utc"})
        one_day_df["date"] = date
        one_day_df["article_count"] = one_day_df["article_count"]\
            .fillna(value=0.0)
        one_day_df["symbol"] = one_day_df["symbol"]\
            .fillna(method='ffill')
        one_day_df["uuid"] = one_day_df["uuid"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)
        one_day_df["title"] = one_day_df["title"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)
        one_day_df["description"] = one_day_df["description"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)
        one_day_df["asset_tags"] = one_day_df["asset_tags"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)
        one_day_df["search_tags"] = one_day_df["search_tags"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)
```

```
        one_day_df["url"] = one_day_df["url"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)
        one_day_df["source_name"] = one_day_df["source_name"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)
        one_day_df["published_utc"] = one_day_df["published_utc"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)
        one_day_df["full_article"] = one_day_df["full_article"]\
            .apply(lambda x: x if x != [np.nan] else np.nan)

        one_day_df = one_day_df.set_index(one_day_df['time_utc'])

        interval_mins = int(self._config["interval"][:-3])

        window_len_past_mins =\
            -self._config["timeseries_input_len_mins"]
        window_len_future_mins =\
            self._config["timeseries_output_len_mins"]

        only_news_df = one_day_df[~one_day_df["url"].isna()].copy()

        for feature in self._config["timeseries_features"]:
            only_news_df[feature+"_past"] = only_news_df["time_utc"]\
                .apply(lambda x: self.time_window(x, feature,
                    one_day_df, window_len_past_mins, interval_mins))
            only_news_df[feature+"_future"]= only_news_df["time_utc"]\
                .apply(lambda x: self.time_window(x, feature,
                    one_day_df,window_len_future_mins, interval_mins))

        required_cols =\
            [x + "_past" for x in\
                self._config["timeseries_features"]] +\
            [x + "_future" for x in\
                self._config["timeseries_features"]] +\
                self._config["scalar_features"]

        for col in list(only_news_df.columns):
            if col not in required_cols:
                del only_news_df[col]

        return only_news_df
```

A.1.4. Kod TensorFlow autorskiego modelu

Funkcja *create_model* jest wykorzystywana do stworzenia sieci neuronowej z wykorzystaniem biblioteki TensorFlow [1].

```
def create_model(
    input_shape,
    output_shape,
    n_hidden,
    encoder_dim,
    text_dropout,
    seq_dropout,
    loss,
    learning_rate,
    embedding_source_size,
    embedding_symbol_size,
    tfhub_handle_encoder,
    tfhub_handle_preprocess,
    nlp_trainable
):

    input_train = Input(shape=input_shape)
    output_train = Input(shape=output_shape)

    # ----- Static input -----
    static_input = tf.keras.layers.Input(
        shape=(),
        dtype=tf.string,
        name='static_features'
    )
    source_lookup = tf.keras.layers.StringLookup(
        vocabulary=vocab
    )(static_input)
    source_embedding = tf.keras.layers.Embedding(
        len(vocab),
        embedding_source_size,
        input_length = 1,
        name="source_embedding"
    )(source_lookup)

    symbol_input = tf.keras.layers.Input(
        shape=(),
```

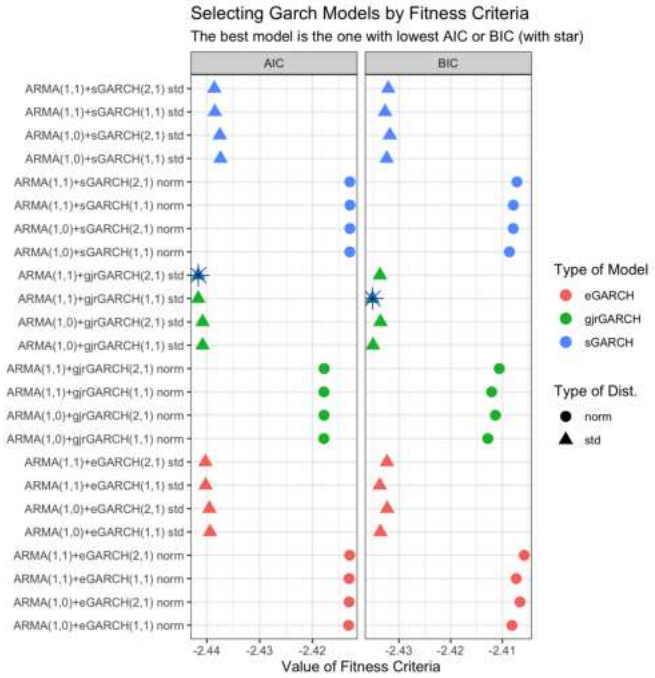
```
        dtype=tf.string,
        name='symbol_features'
    )
    symbol_lookup = tf.keras.layers.StringLookup(
        vocabulary=vocab
    )(symbol_input)
    symbol_embedding = tf.keras.layers.Embedding(
        len(vocab_symbol),
        embedding_symbol_size,
        input_length = 1,
        name="symbol_embedding"
    )(symbol_lookup)

    # ---- Text input -----
    text_input = tf.keras.layers.Input(
        shape=(),
        dtype=tf.string,
        name='text'
    )
    preprocessing_layer = hub.KerasLayer(
        tfhub_handle_preprocess,
        name='preprocessing'
    )
    encoder_inputs = preprocessing_layer(text_input)
    encoder = hub.KerasLayer(
        tfhub_handle_encoder,
        trainable=nlp_trainable,
        name='NLP_encoder'
    )
    outputs = encoder(encoder_inputs)
    net = outputs['pooled_output']
    net = tf.keras.layers.Dropout(text_dropout)(net)

    concat_static = tf.keras.layers.Concatenate()([
        net, source_embedding, symbol_embedding
    ])
    dense_layer_1 = Dense(
        encoder_dim,
        name='news_encoder'
    )(concat_static)
```

```
    # -------- Seq encoder -----
    encoder_last_h1, encoder_last_h2, encoder_last_c = LSTM(
        n_hidden,
        activation='elu',
        dropout=seq_dropout,
        recurrent_dropout=seq_dropout,
        return_sequences=False,
        return_state=True
    )(input_train)

    encoder_last_h1 = tf.keras.layers.BatchNormalization(
        momentum=0.6
    )(encoder_last_h1)
    encoder_last_h1 = tf.keras.layers.Concatenate()([
        encoder_last_h1, dense_layer_1
    ])
    encoder_last_h1 = Dense(n_hidden)(encoder_last_h1)

    encoder_last_c = tf.keras.layers.BatchNormalization(
        momentum=0.6
    )(encoder_last_c)
    encoder_last_c = tf.keras.layers.Concatenate()([
        encoder_last_c, dense_layer_1
    ])
    encoder_last_c = Dense(n_hidden)(encoder_last_c)

    # --------- Seq decoder -----
    decoder = tf.keras.layers.RepeatVector(
        output_shape[1]
    )(encoder_last_h1)

    decoder = LSTM(
        n_hidden,
        activation='elu',
        dropout=seq_dropout,
        recurrent_dropout=seq_dropout,
        return_state=False,
        return_sequences=True
    )(decoder, initial_state=[encoder_last_h1, encoder_last_c])

    out = TimeDistributed(Dense(output_shape[2]))(decoder)
```

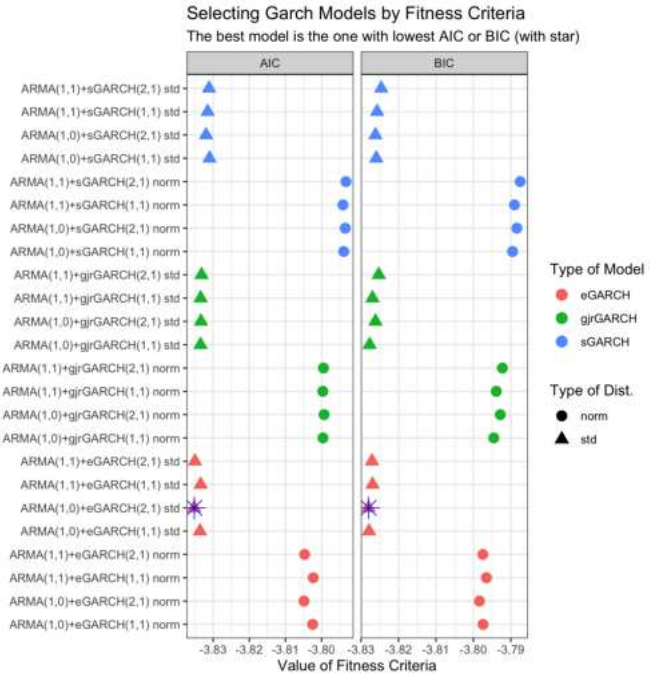
```
    model = Model(
        inputs=[input_train, text_input, static_input, symbol_input],
        outputs=out
    )
    opt = tf.keras.optimizers.Adam(lr=learning_rate)
    model.compile(loss=loss, optimizer=opt, metrics=['mae'])
    model.summary()

    return model
```

A.2. Wybór specyfikacji i rzędów opóźnień bazowych modeli ARMA-GARCH

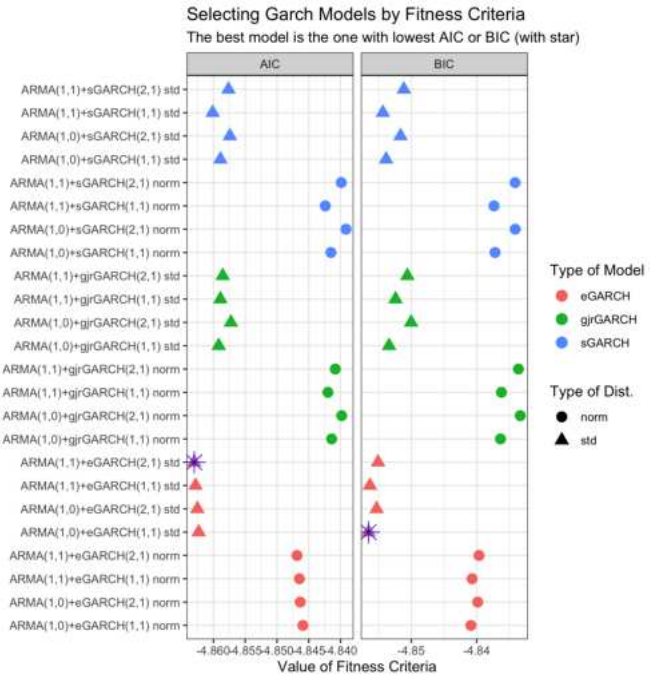
Poniżej przedstawione zostały wyniki doboru specyfikacji, rzędów opóźnień oraz rozkładu prawdopodobieństw modeli ARMA-GARCH dla każdego z badanych instrumentów. Analiza została przeprowadzona przy pomocy programu w języku R z repozytorium udostępnionego wraz z publikacją Perlina M. et al [150].



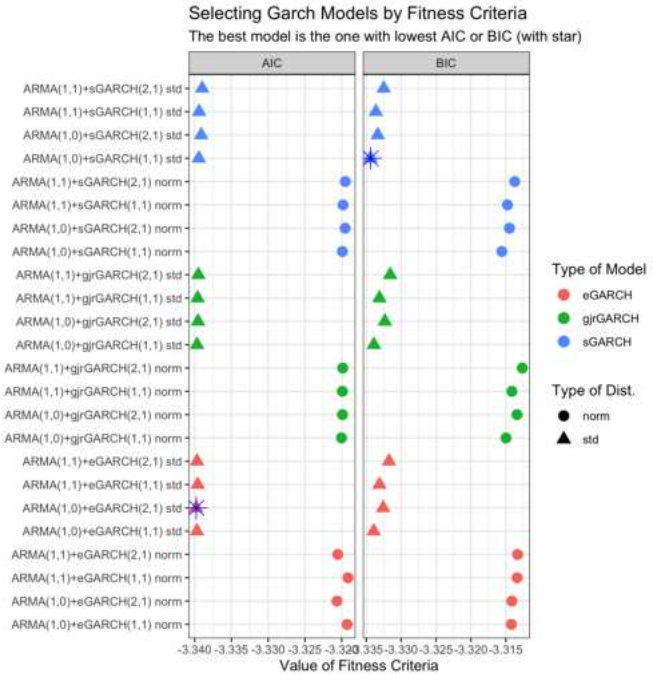
Rysunek A.1. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki TSLA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



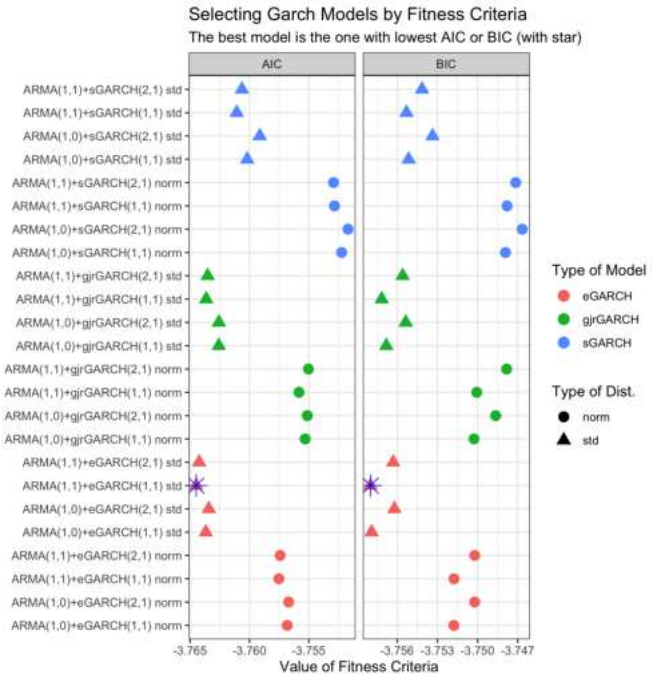
Rysunek A.2. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki NKE. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



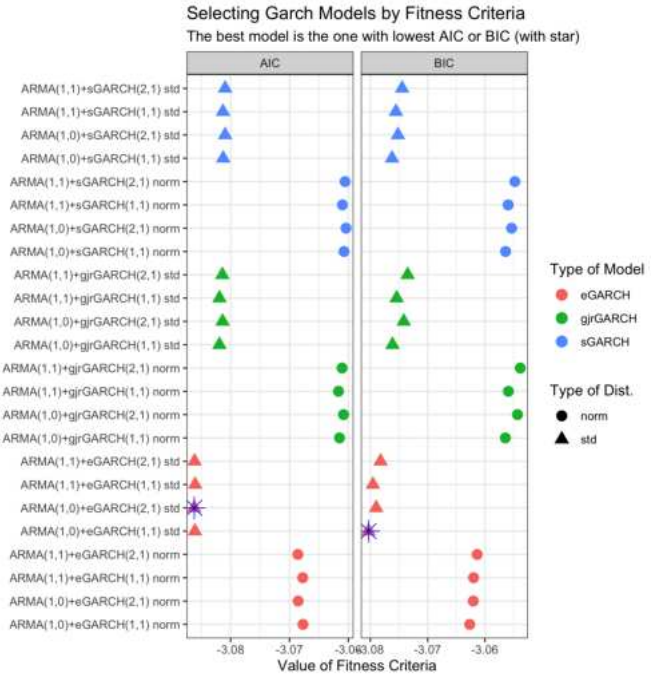
Rysunek A.3. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki KO. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



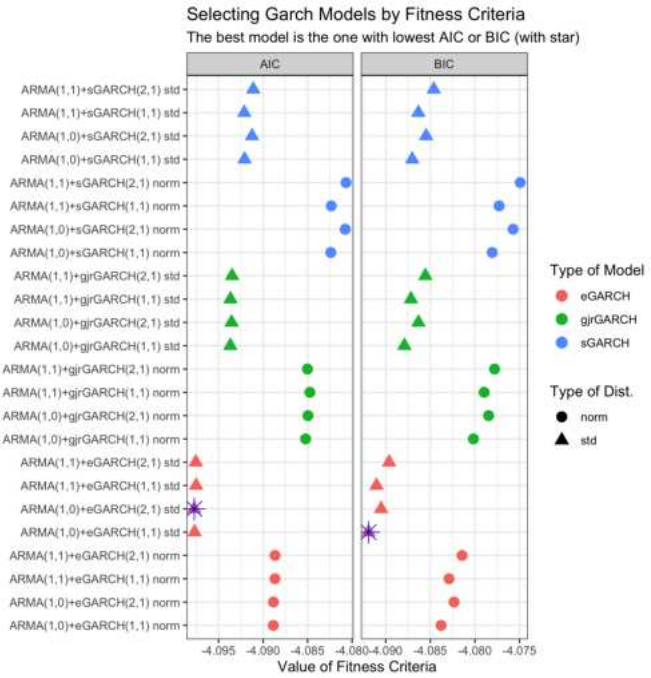
Rysunek A.4. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki BA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



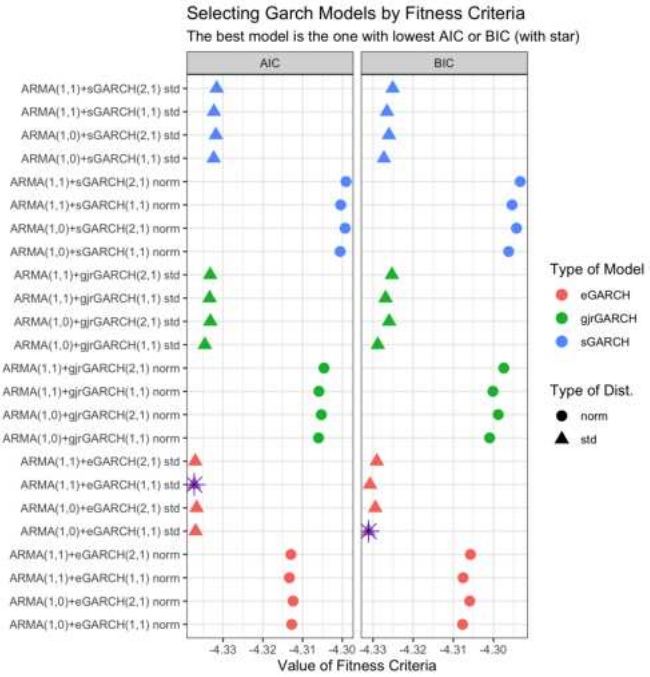
Rysunek A.5. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki CRM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



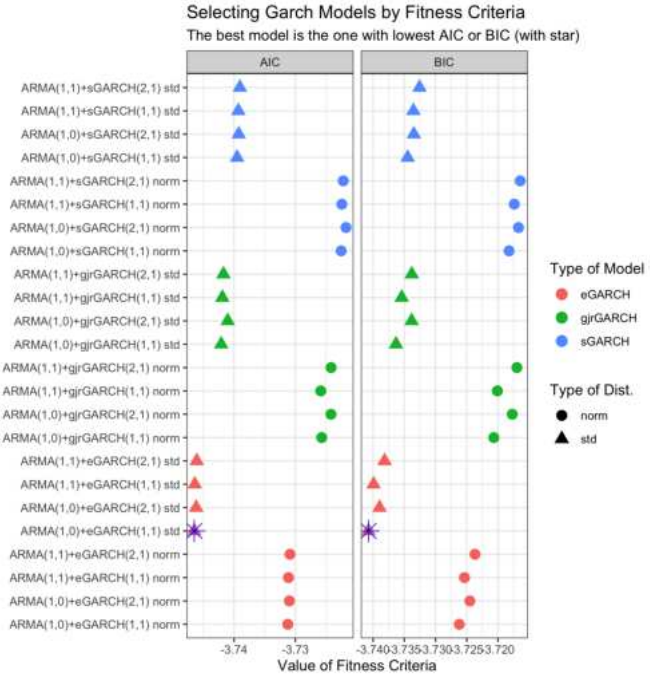
Rysunek A.6. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki GM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



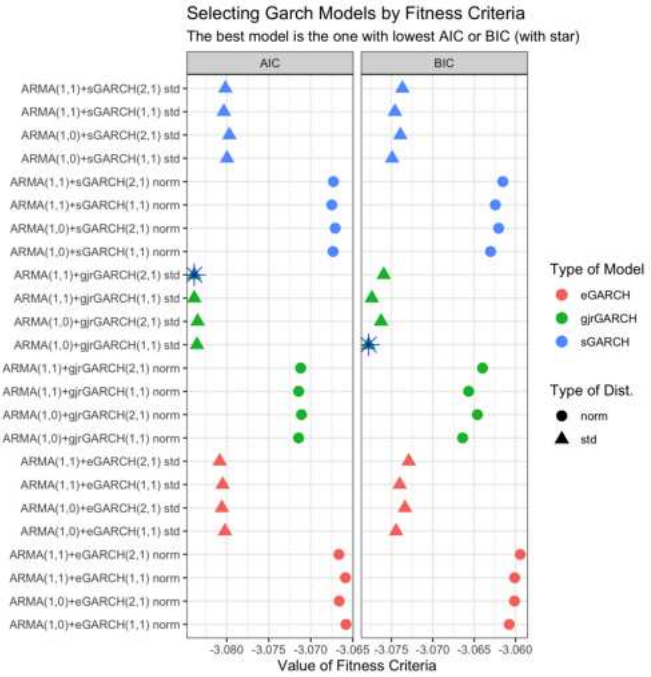
Rysunek A.7. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki JPM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



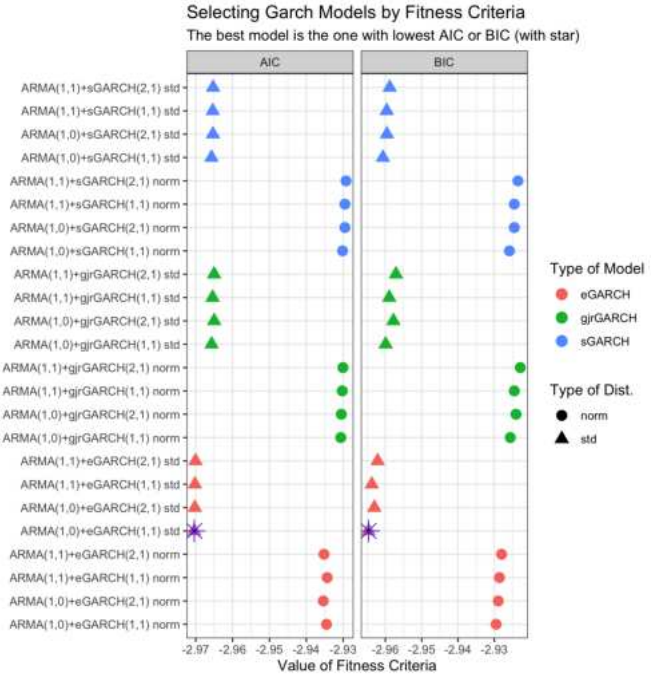
Rysunek A.8. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki MSFT. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



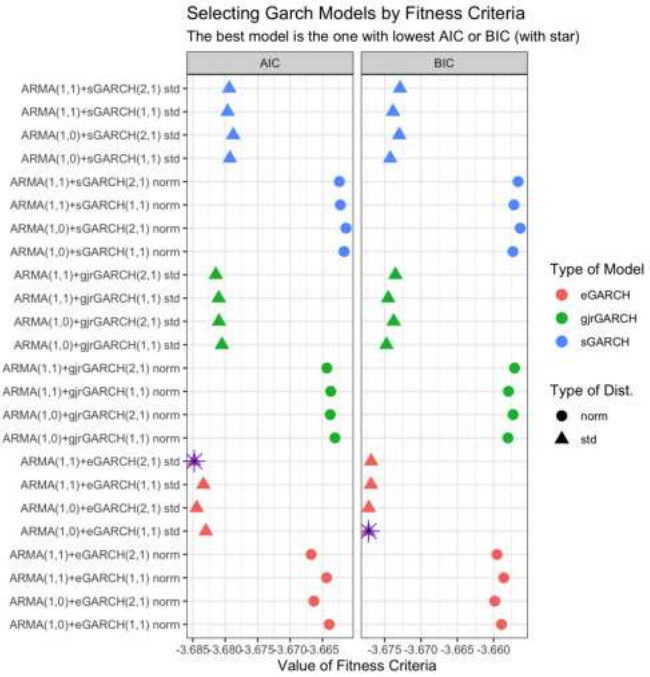
Rysunek A.9. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki GS. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



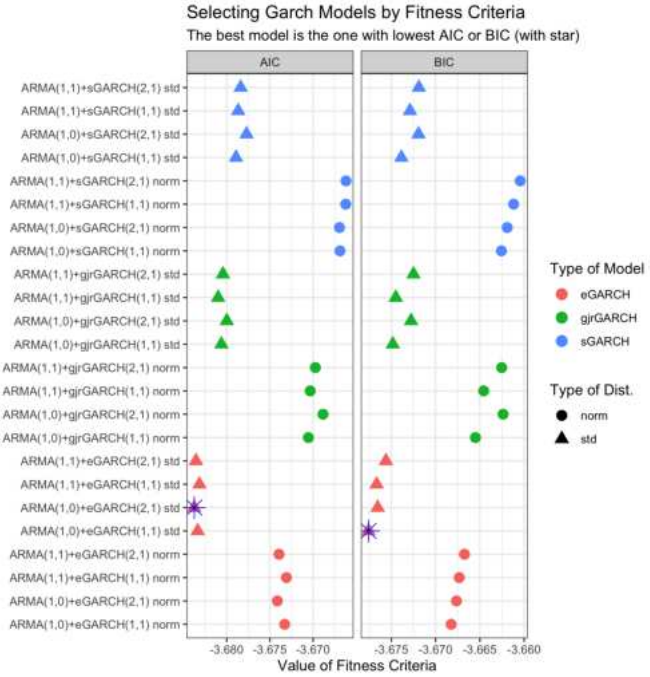
Rysunek A.10. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki NVDA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



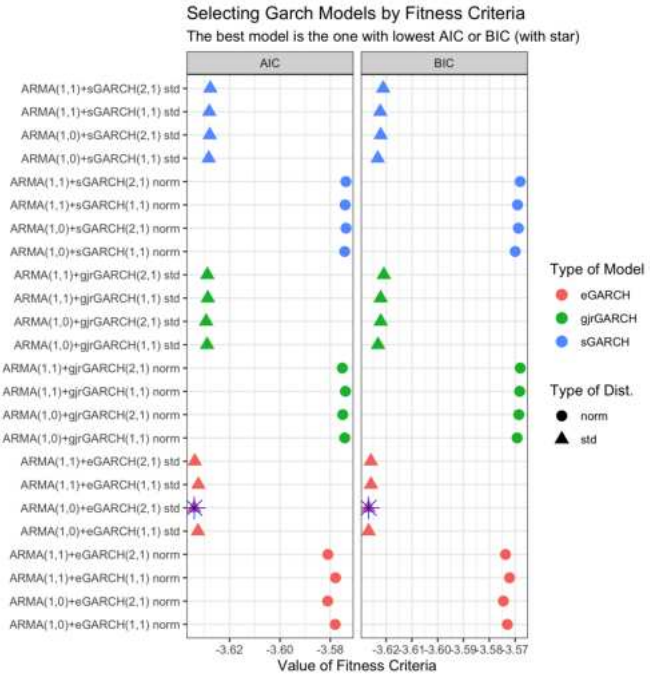
Rysunek A.11. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki F. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



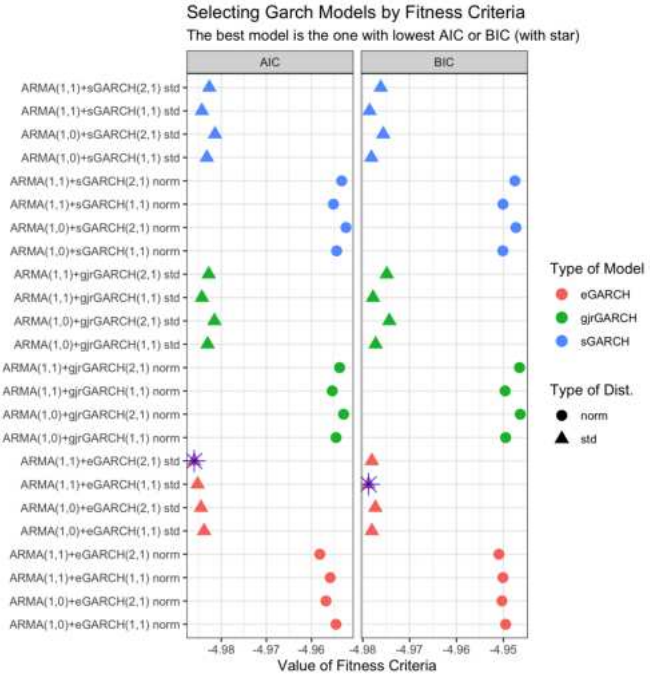
Rysunek A.12. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki C. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



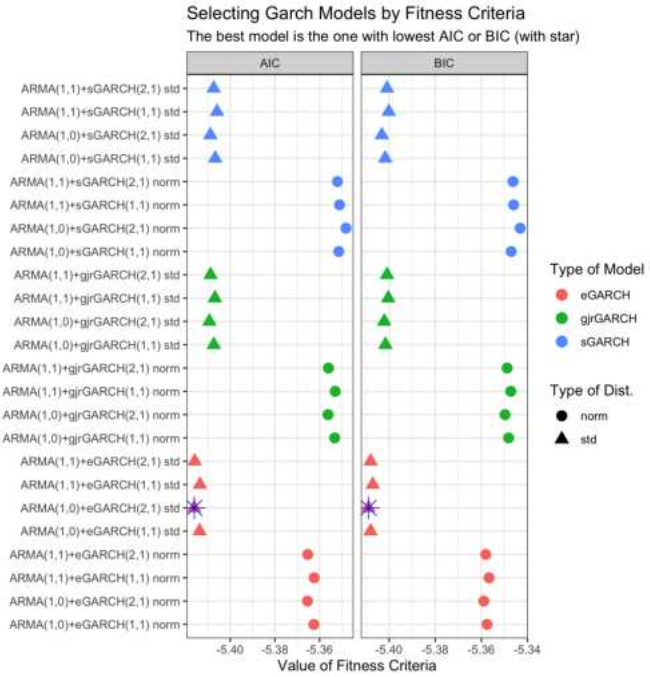
Rysunek A.13. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki MS. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



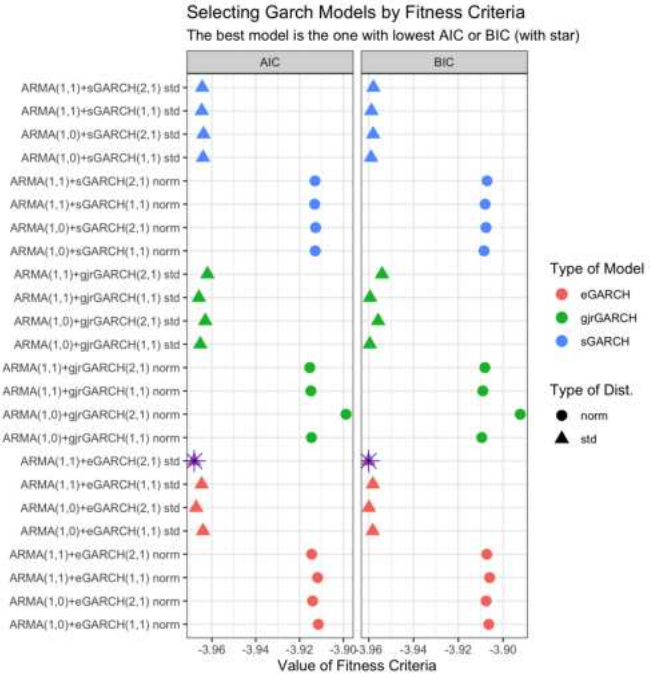
Rysunek A.14. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki INTC. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



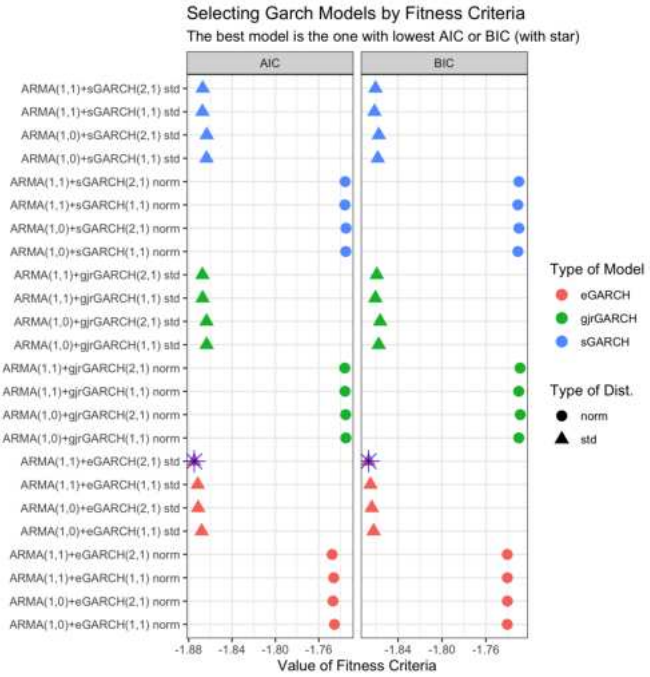
Rysunek A.15. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki VZ. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



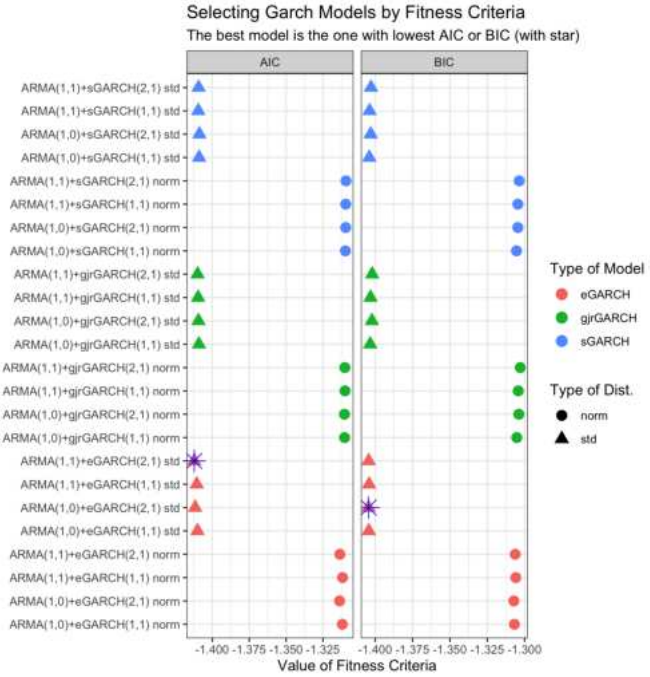
Rysunek A.16. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu złota. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



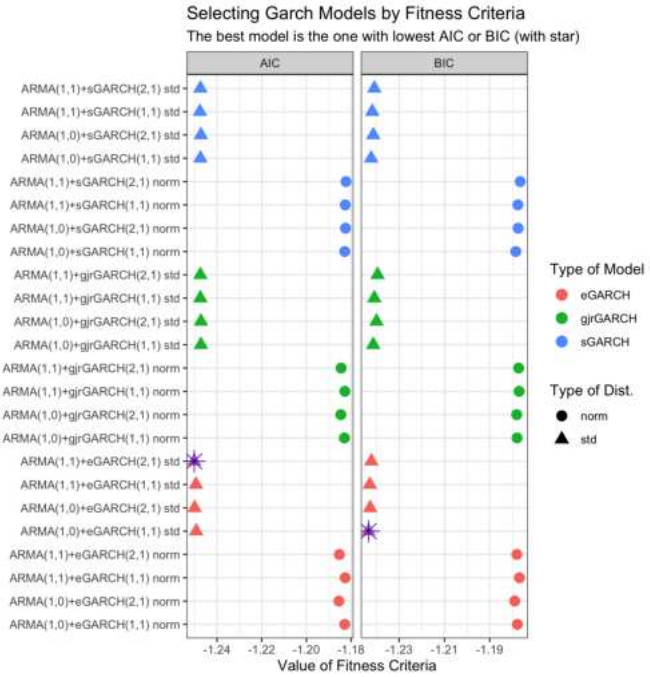
Rysunek A.17. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu srebra. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



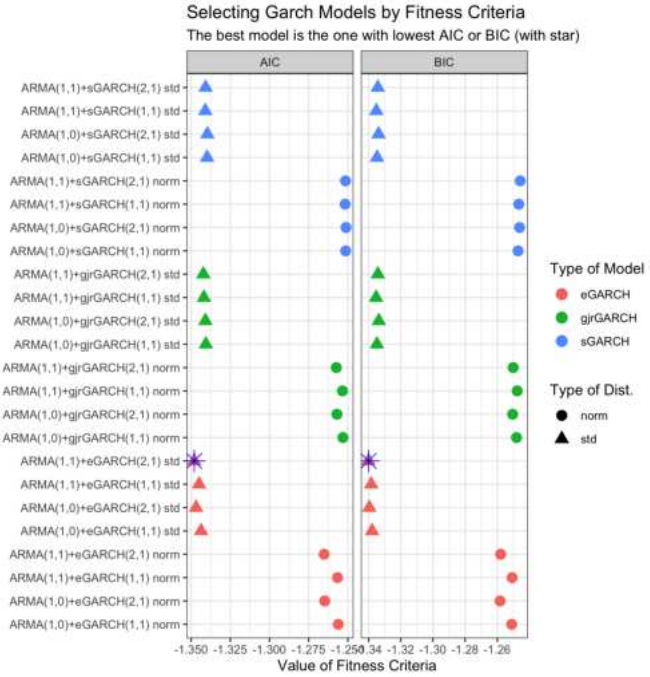
Rysunek A.18. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu Bitcoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



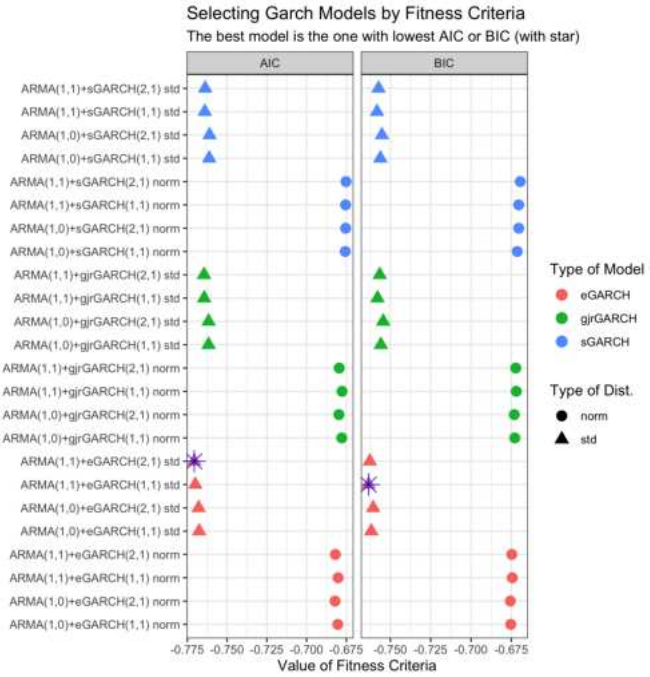
Rysunek A.19. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu Ethereum. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.20. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu Cardano. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



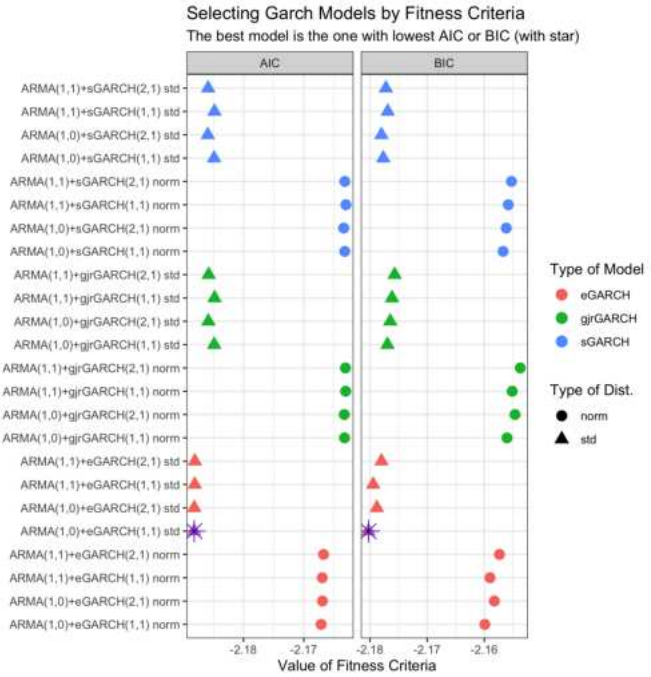
Rysunek A.21. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu Litecoin. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



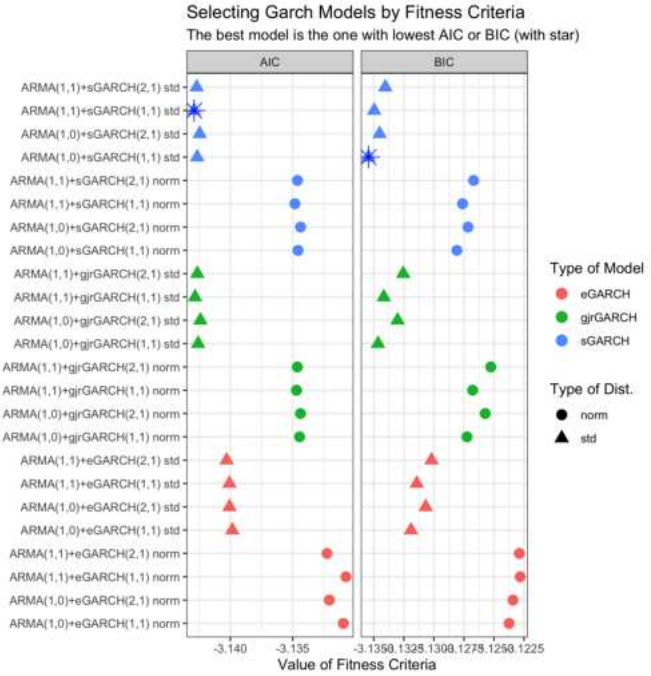
Rysunek A.22. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH dla logarytmicznych stóp zwrotu spółki Dogecoin. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

A.3. Wybór specyfikacji i rzędów opóźnień modeli ARMA-GARCH-FENE

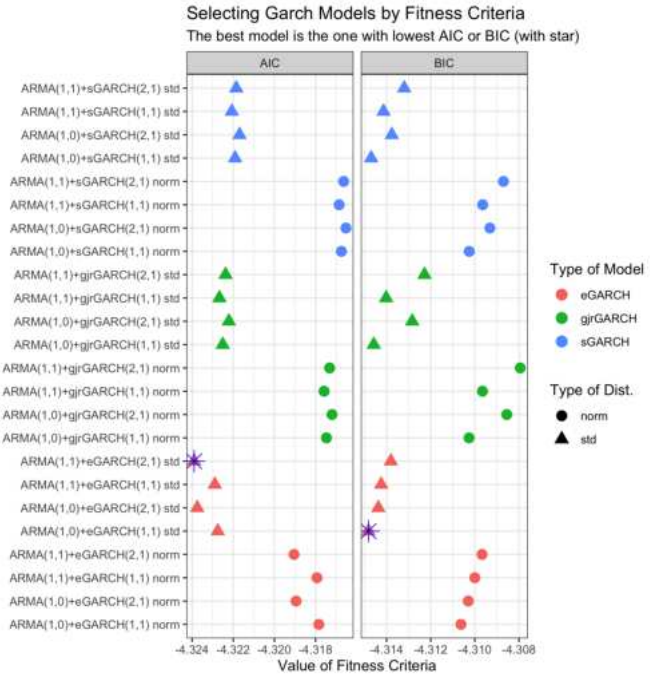
Poniżej przedstawione zostały wyniki doboru specyfikacji, rzędów opóźnień oraz rozkładu prawdopodobieństw modeli ARMA-GARCH-FENE dla każdego z badanych instrumentów. Analiza została przeprowadzona przy pomocy programu w języku R z repozytorium udostępnionego wraz z publikacją Perlina M. et al [150].



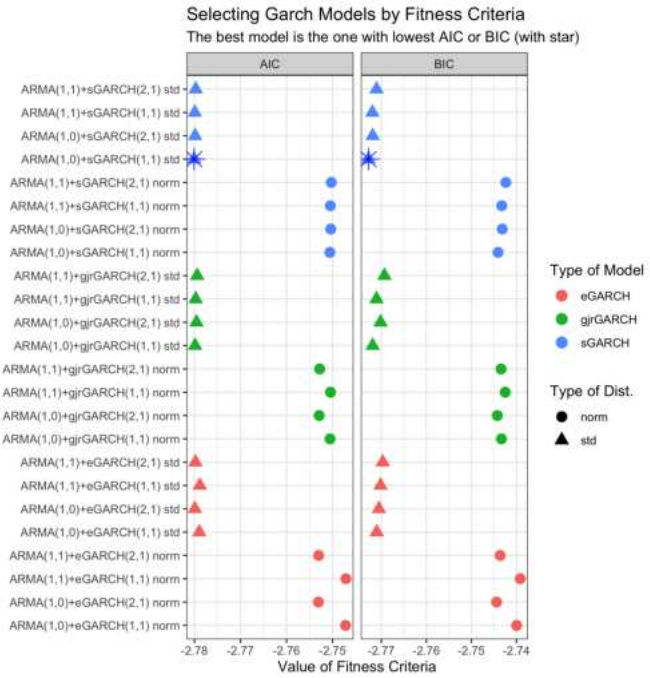
Rysunek A.23. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki TSLA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



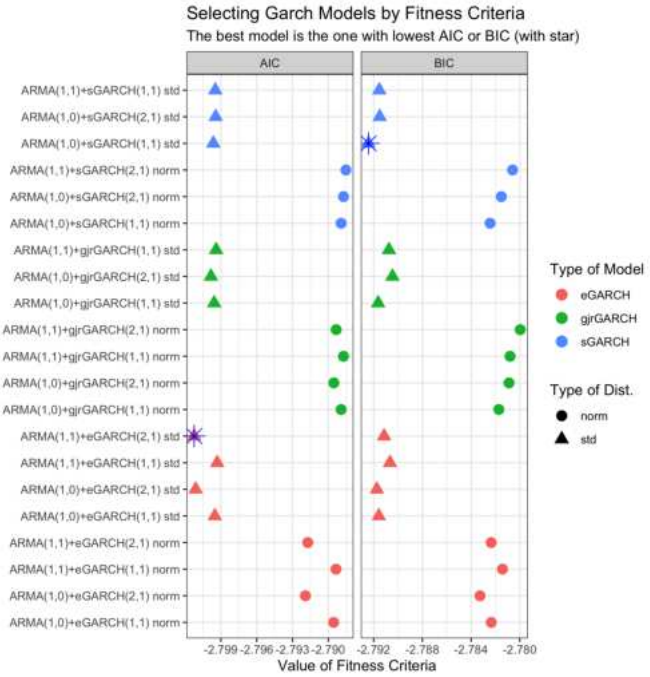
Rysunek A.24. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki NKE. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



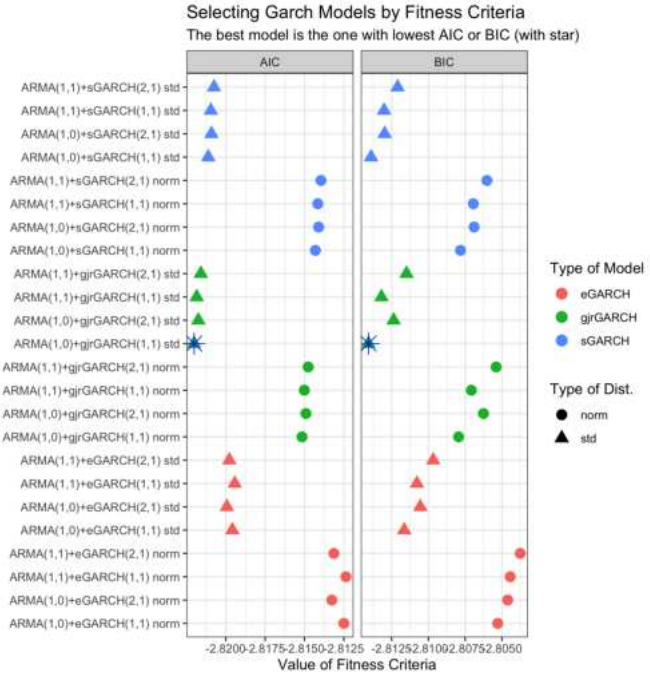
Rysunek A.25. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki KO. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



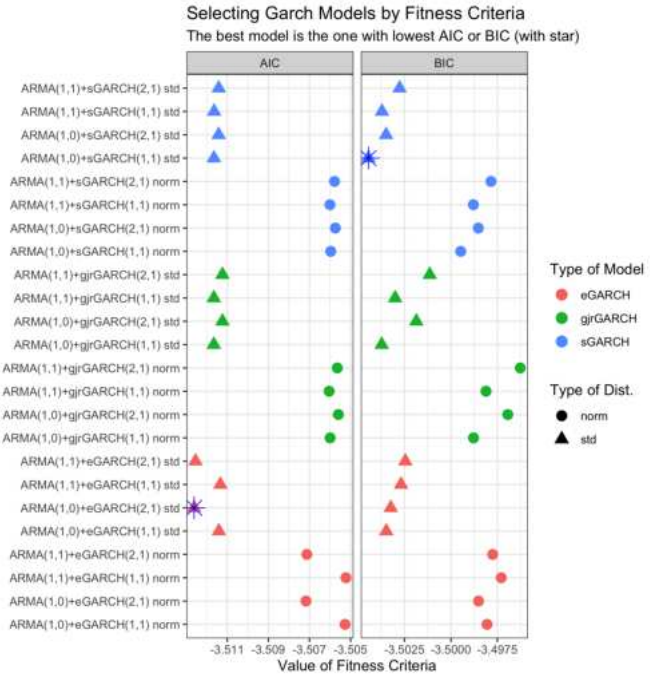
Rysunek A.26. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki BA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



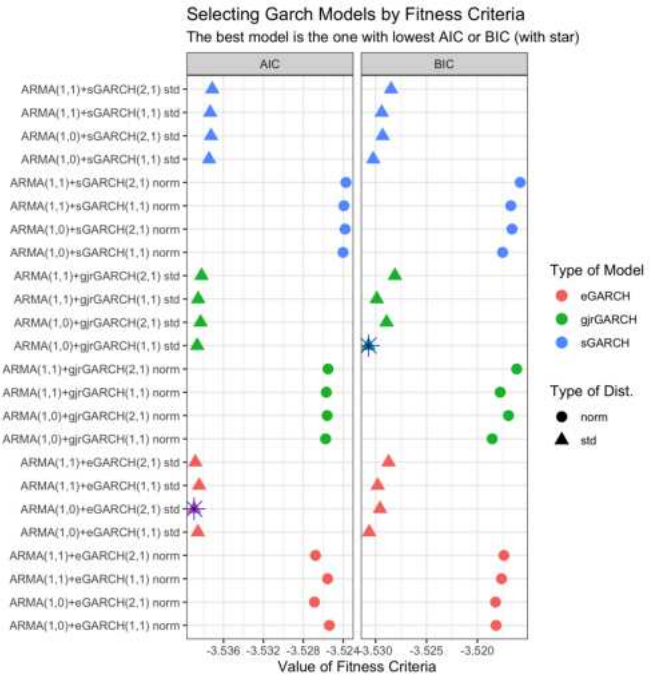
Rysunek A.27. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki CRM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.28. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki GM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

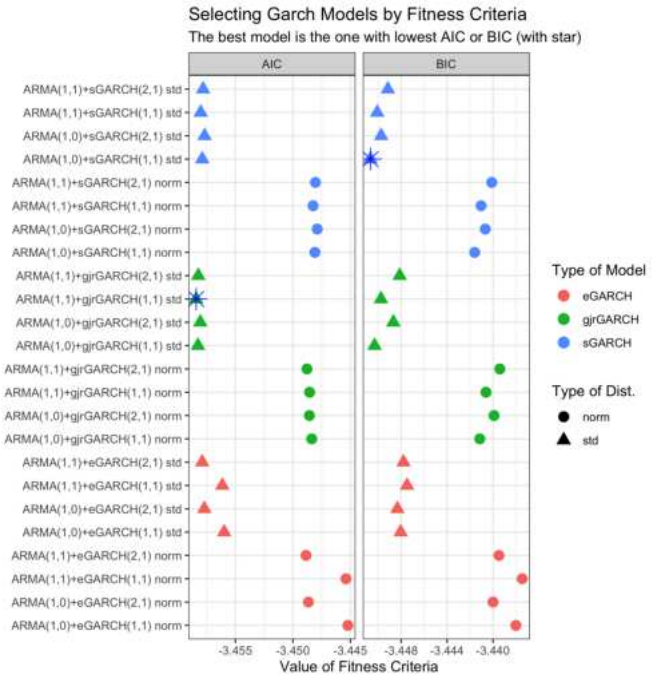


Rysunek A.29. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki JPM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

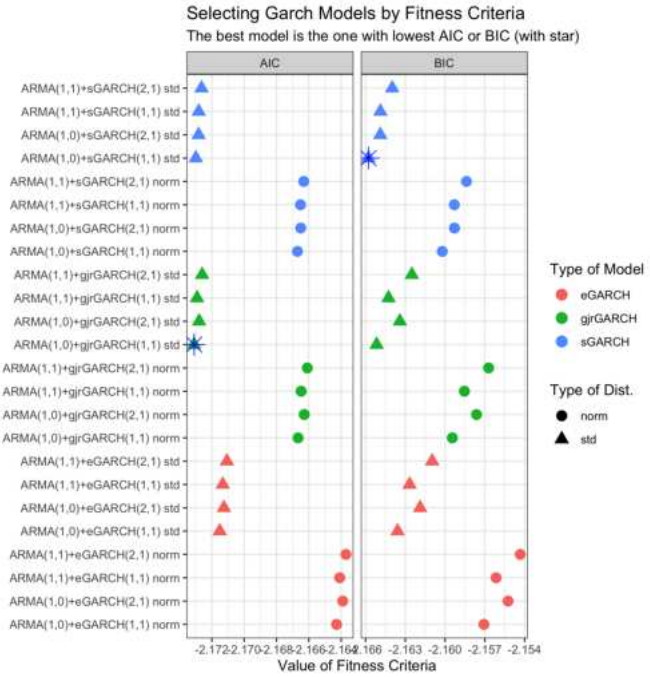


Rysunek A.30. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki MSFT. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

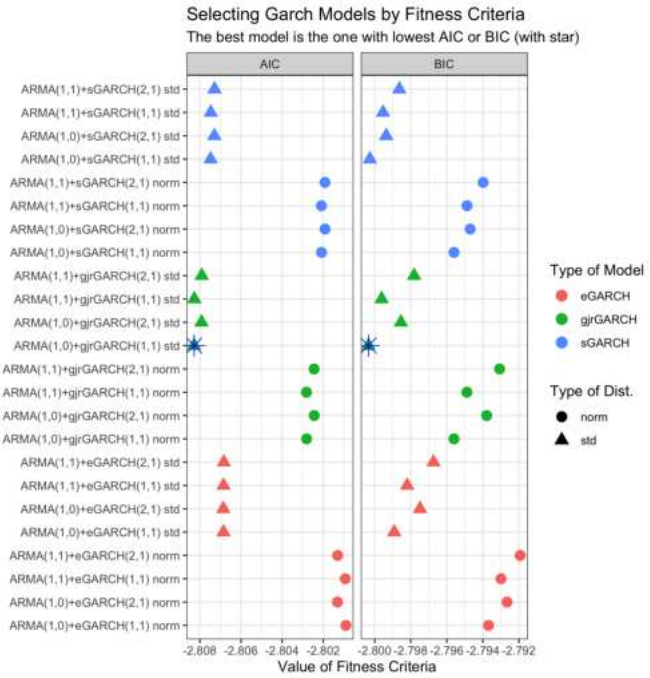
A.3.1. Modele z enkoderem o długości $d_{encoder} = 4$



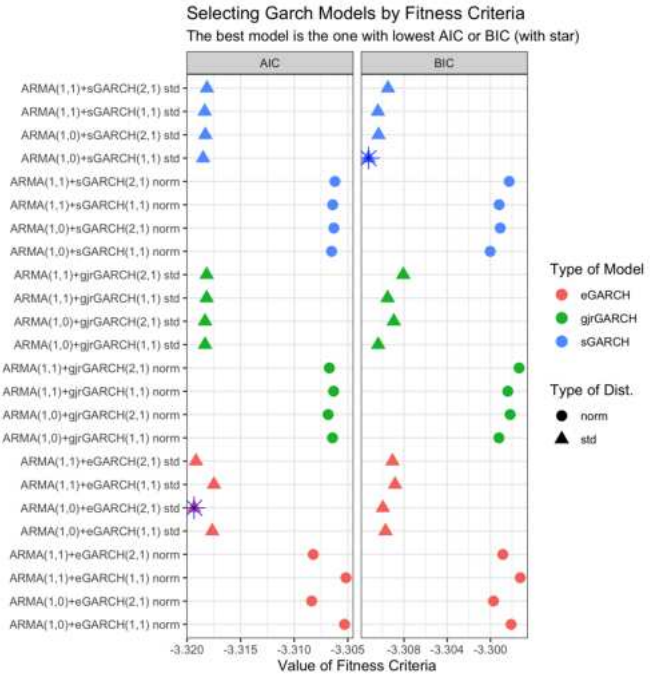
Rysunek A.31. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki GS. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



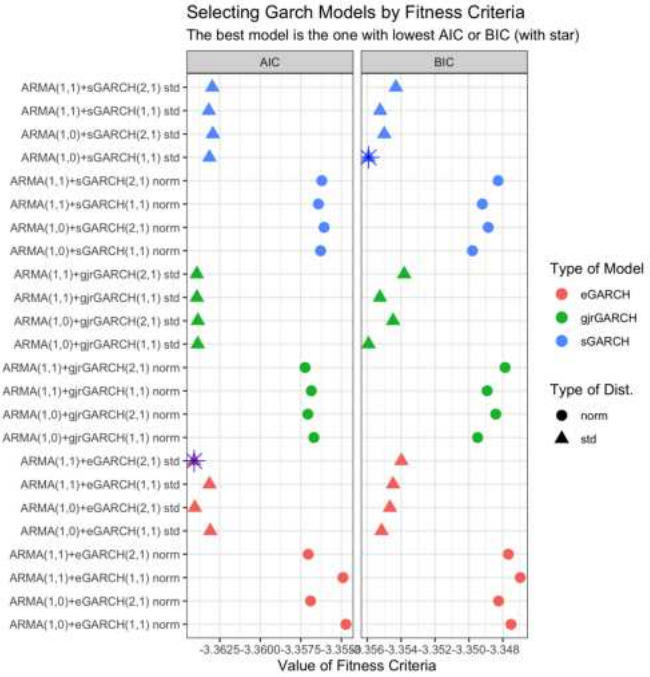
Rysunek A.32. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki NVDA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.33. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki F. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

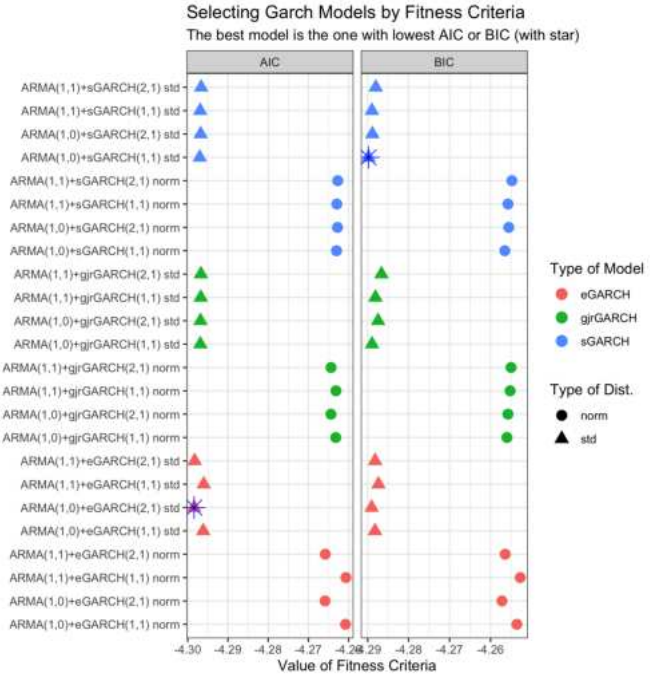


Rysunek A.34. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki C. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

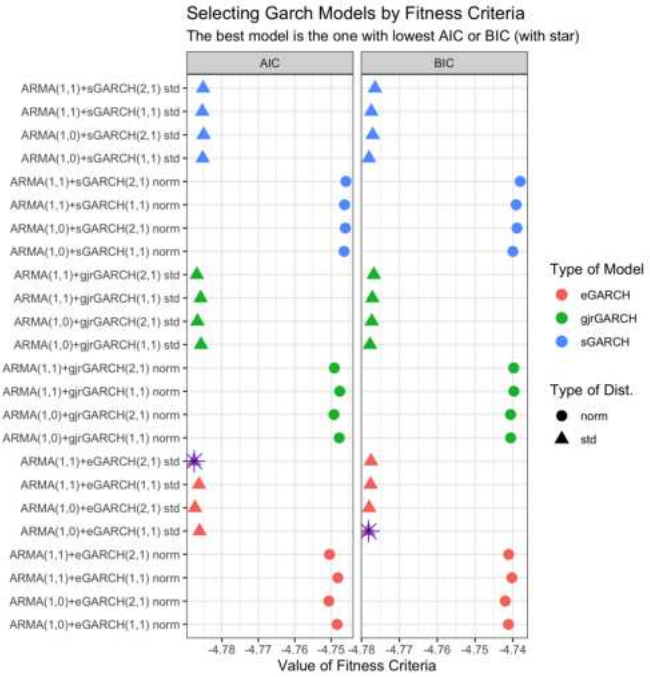


Rysunek A.35. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki MS. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

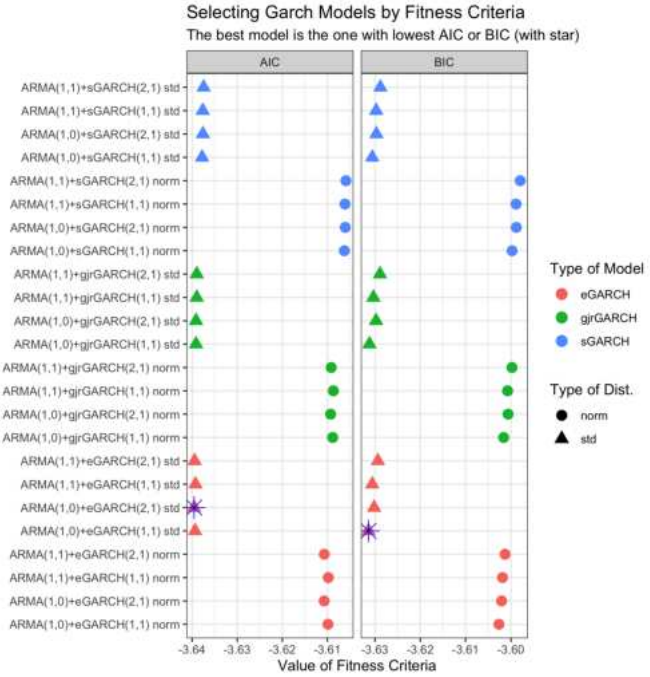
Rysunek A.36. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki INTC. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



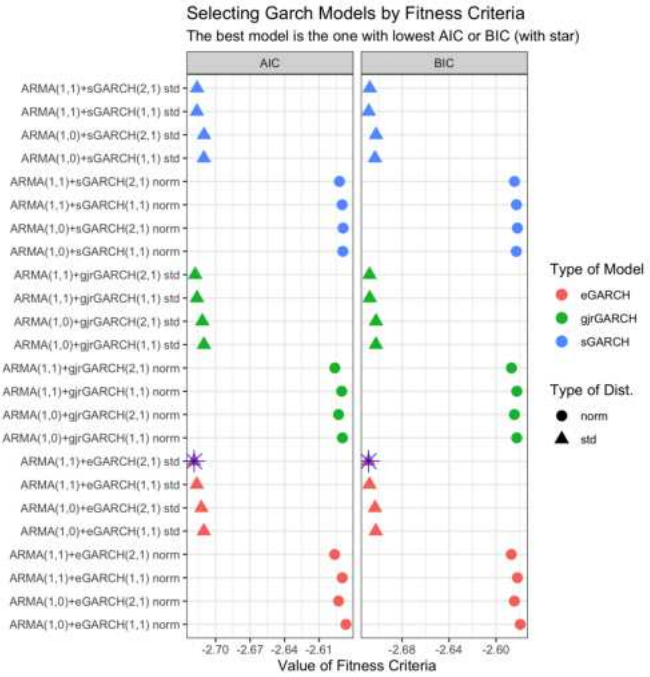
Rysunek A.37. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki VZ. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.38. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu złota. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

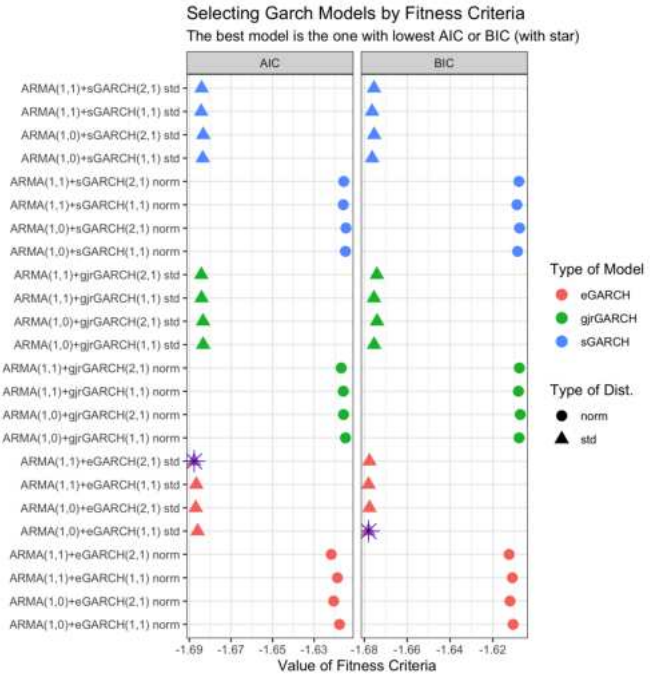


Rysunek A.39. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu srebra. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

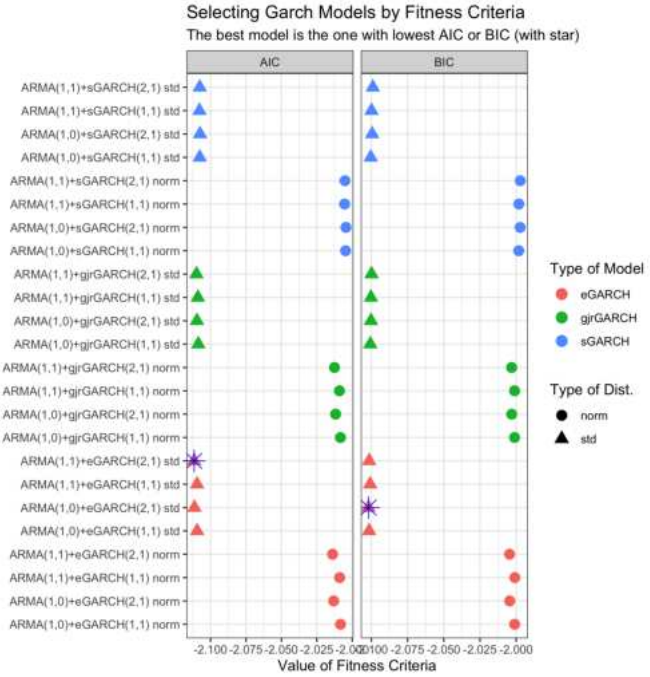


Rysunek A.40. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu Bitcoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

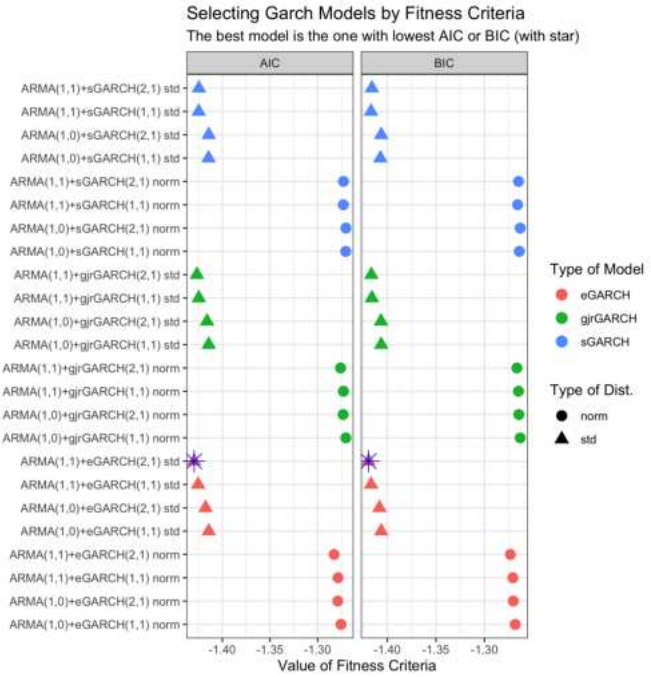
Rysunek A.41. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki Ethereum. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



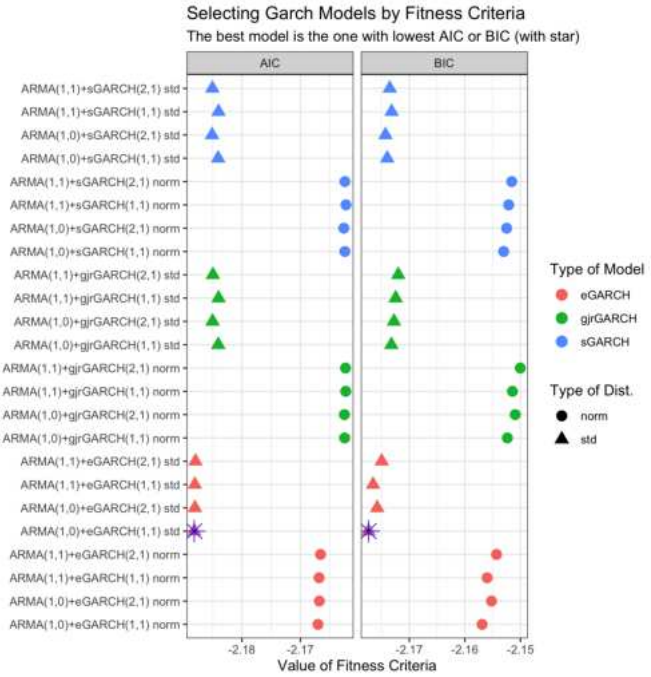
Rysunek A.42. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki Cardano. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



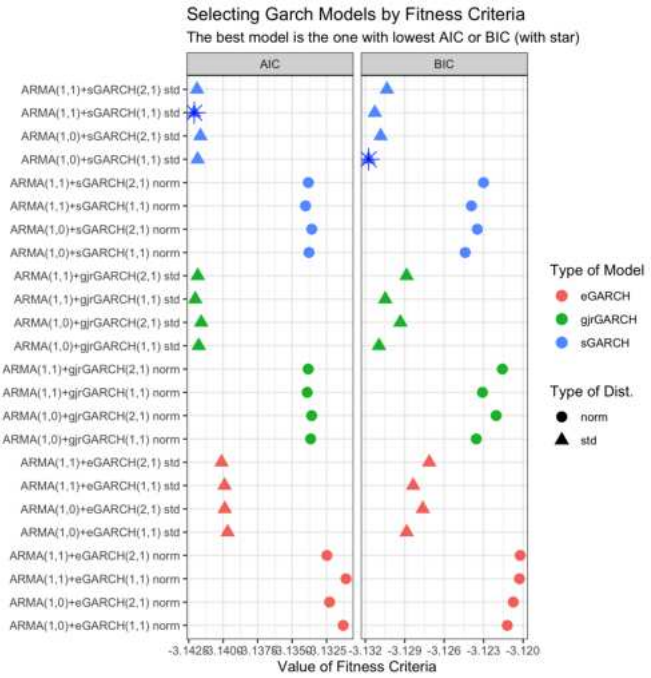
Rysunek A.43. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki Litecoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



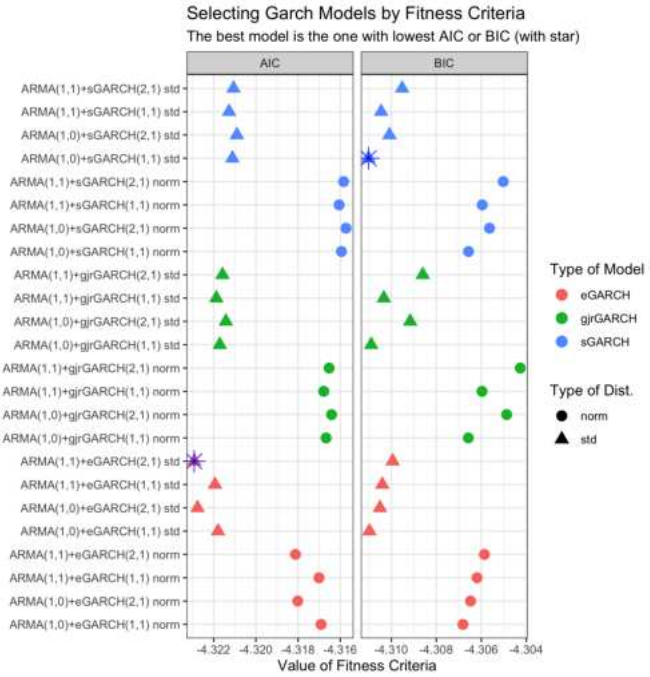
Rysunek A.44. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 4$) dla logarytmicznych stóp zwrotu spółki Dogecoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów. Źródło: Obliczenia własne.



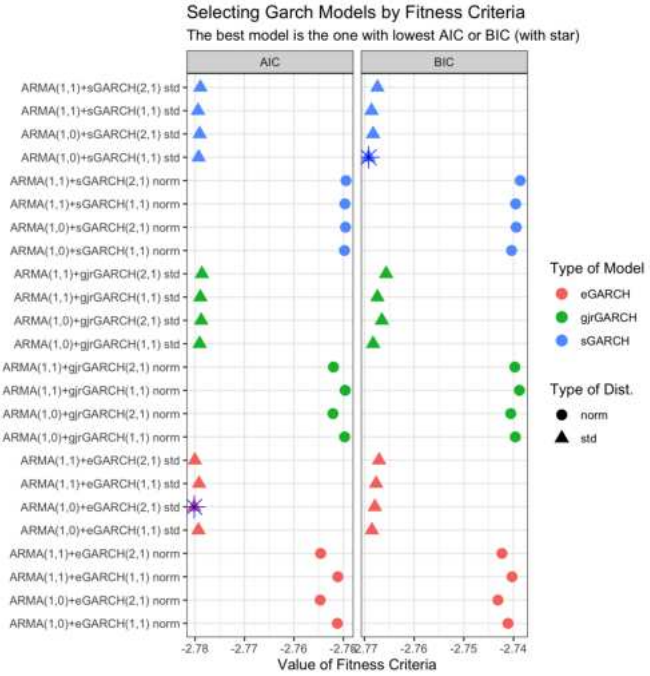
Rysunek A.45. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki TSLA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



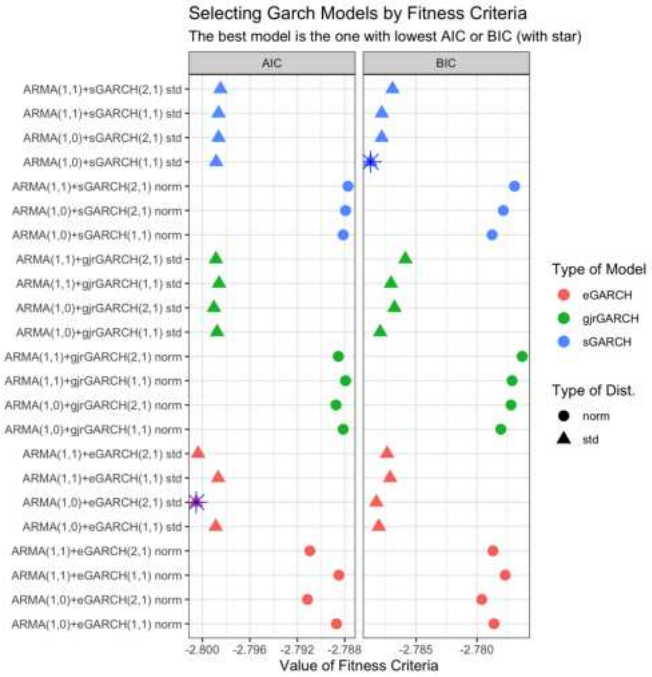
Rysunek A.46. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki NKE. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



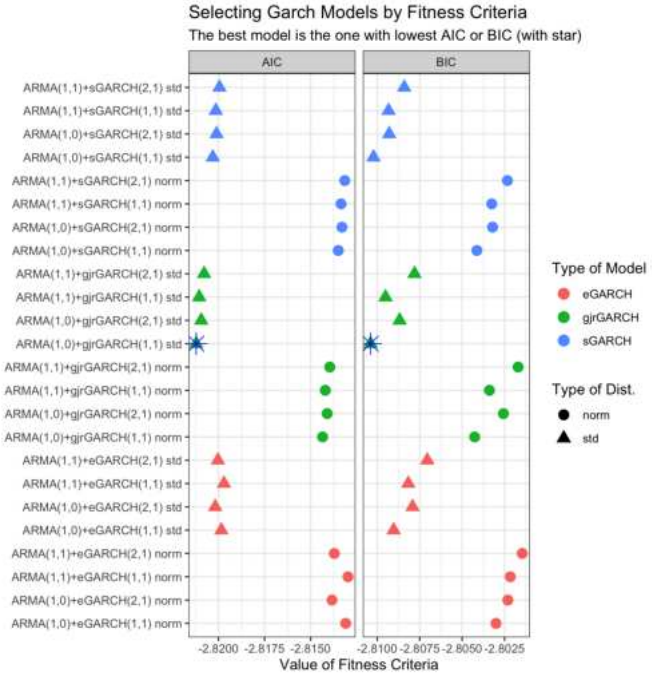
Rysunek A.47. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki KO. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



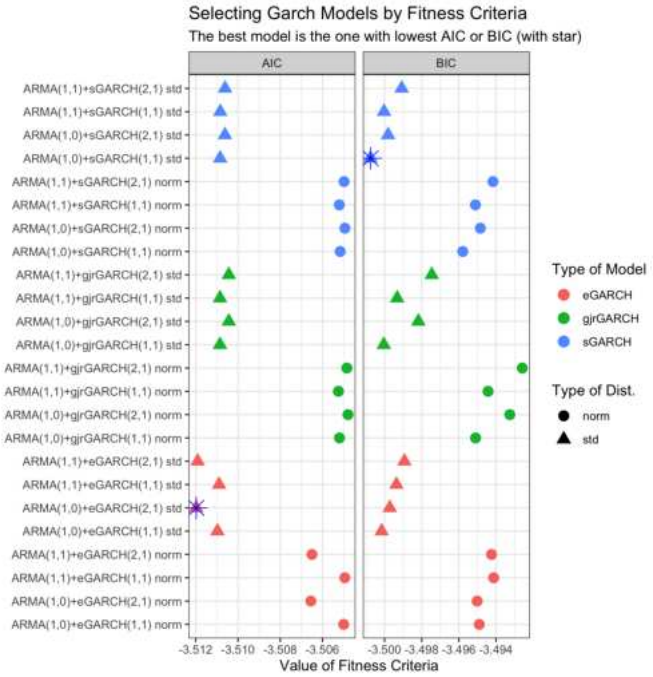
Rysunek A.48. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki BA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



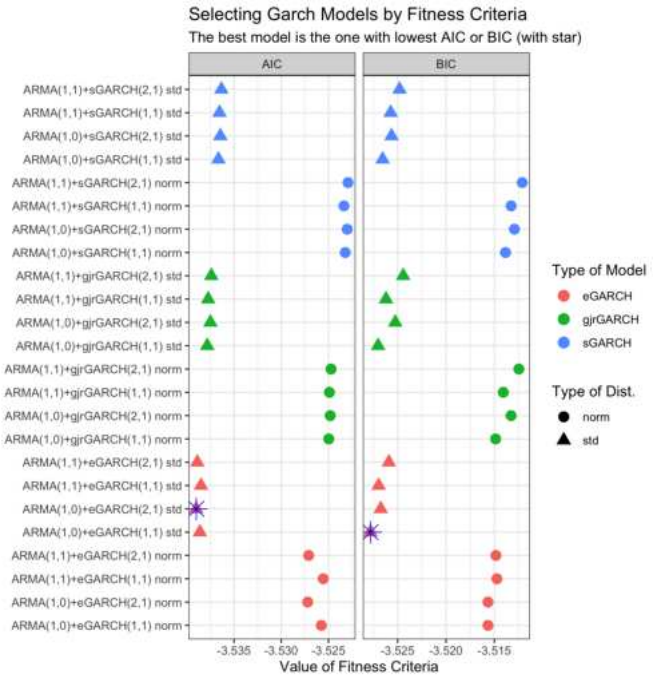
Rysunek A.49. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki CRM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



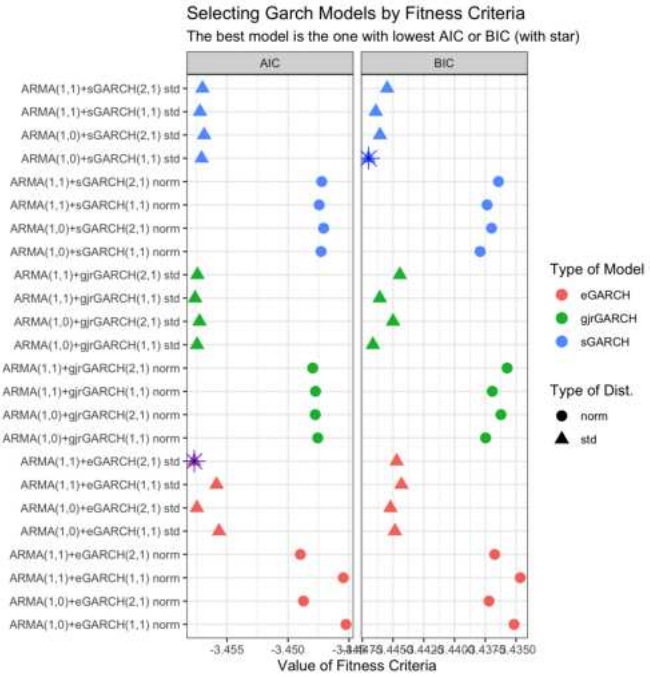
Rysunek A.50. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki GM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



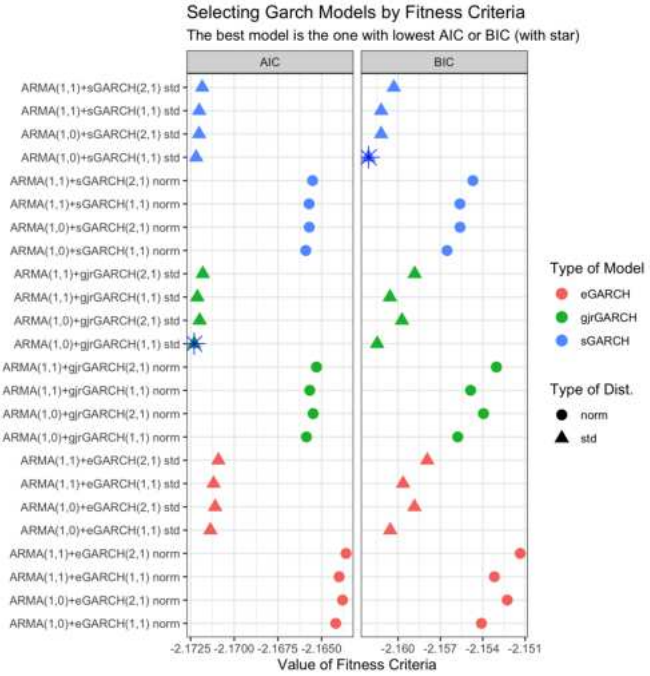
Rysunek A.51. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki JPM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



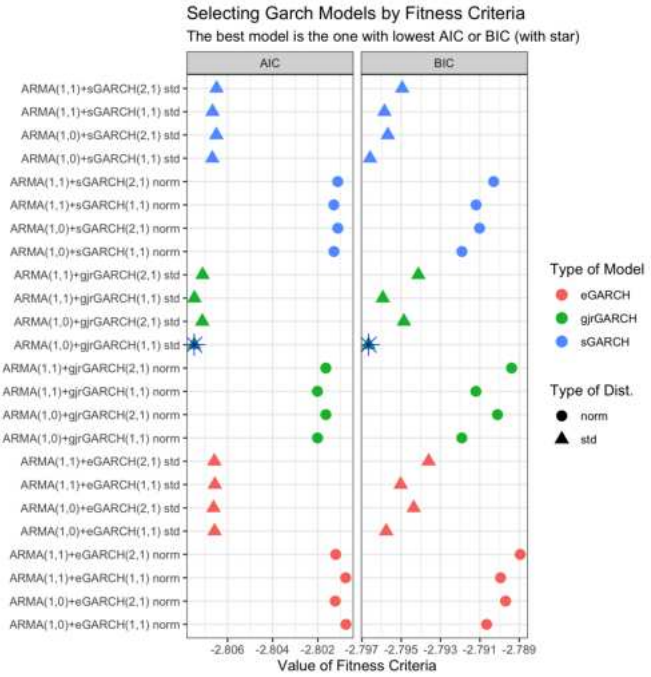
Rysunek A.52. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki MSFT. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



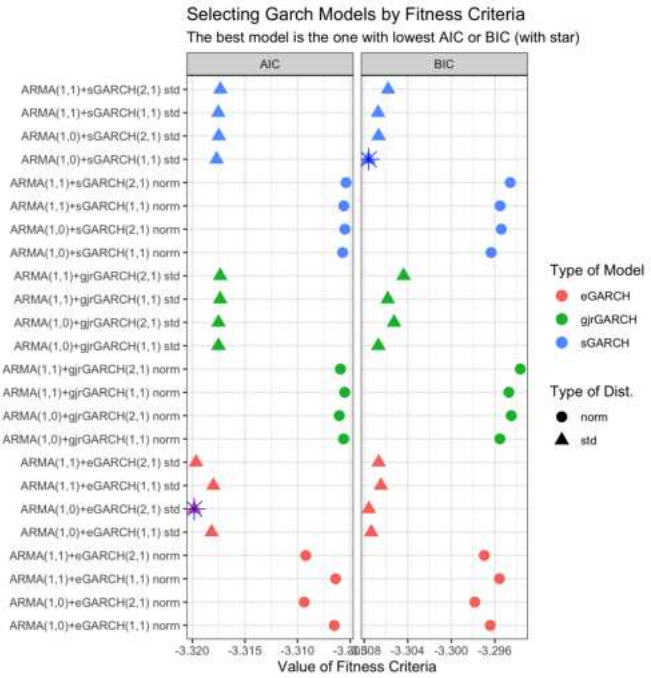
Rysunek A.53. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki GS. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.54. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki NVDA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

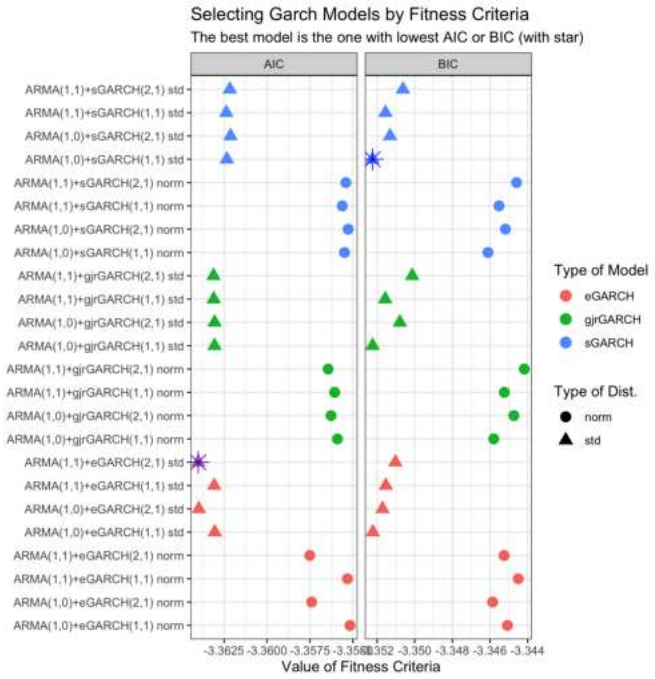


Rysunek A.55. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki F. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

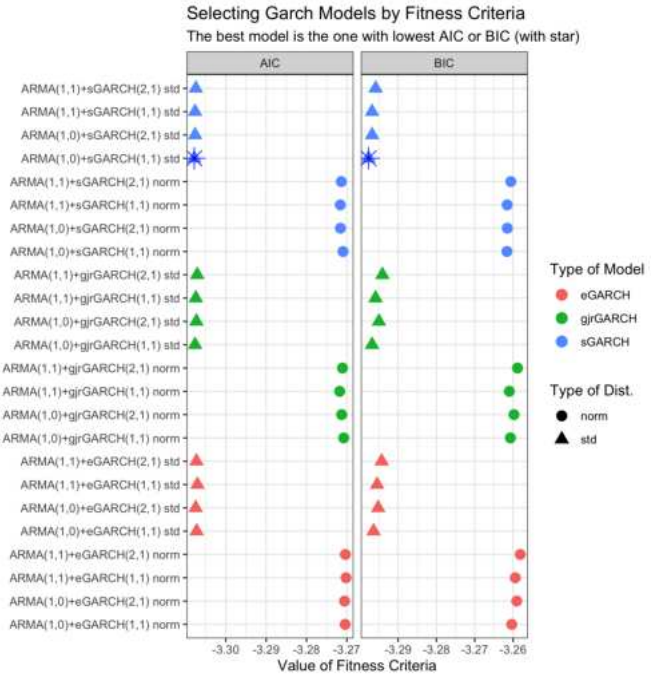


Rysunek A.56. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki C. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

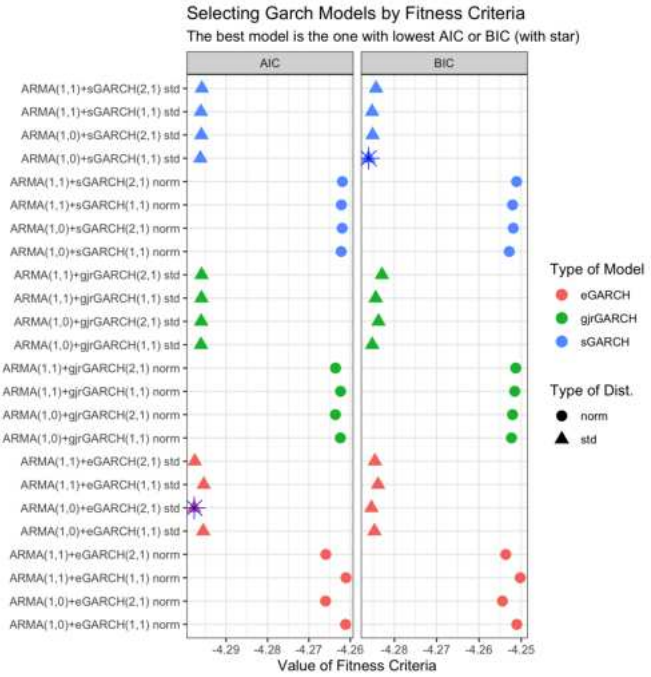
A.3.2. Modele z enkoderem o długości $d_{encoder} = 8$



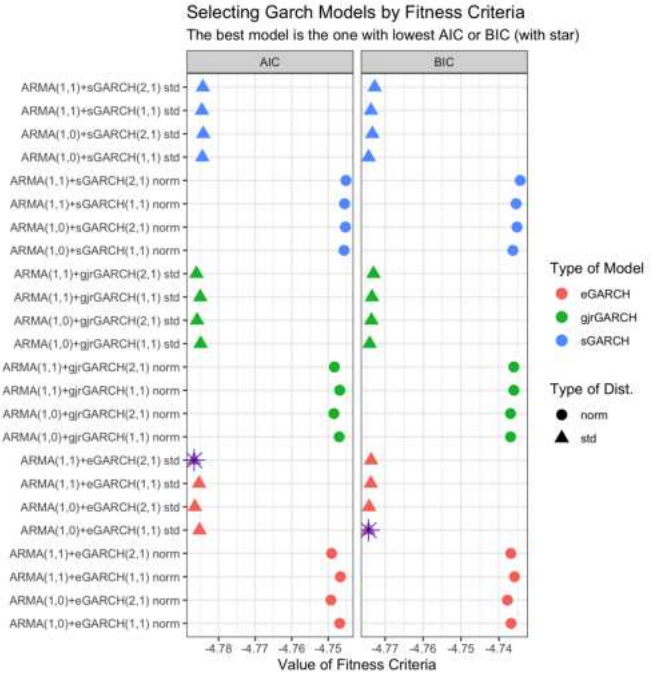
Rysunek A.57. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki MS. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



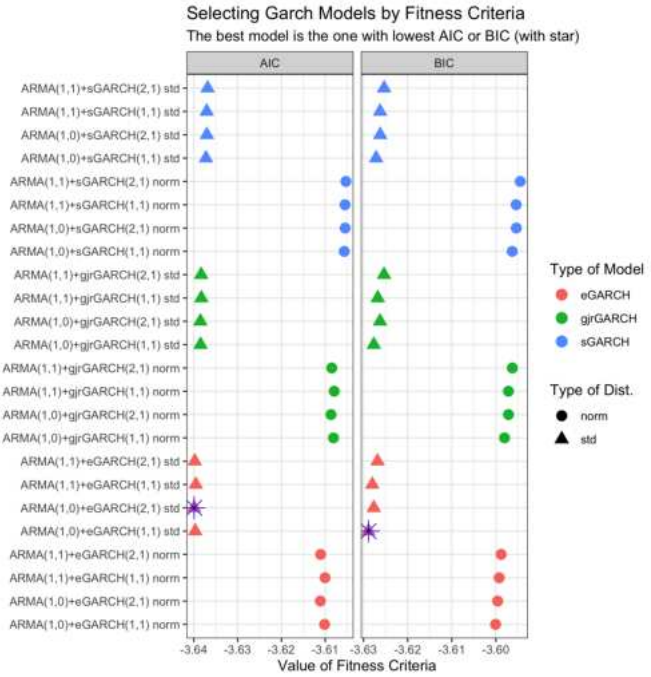
Rysunek A.58. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki INTC. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



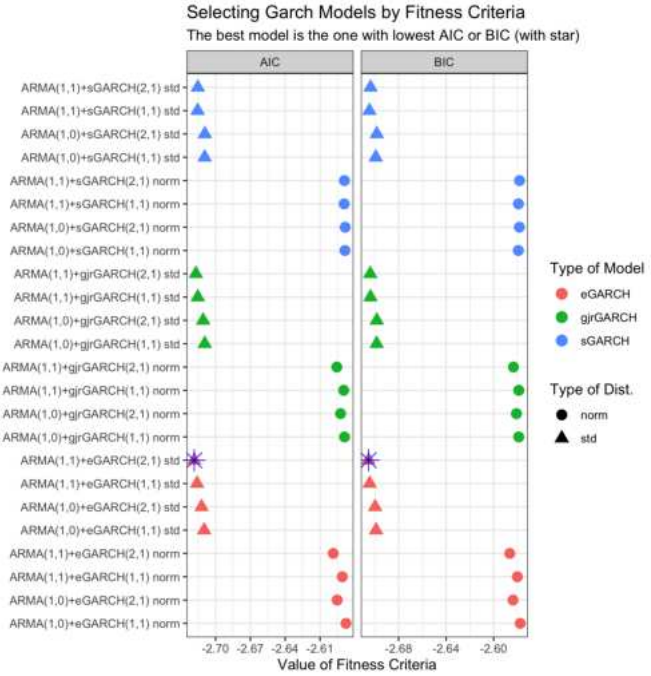
Rysunek A.59. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki VZ. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



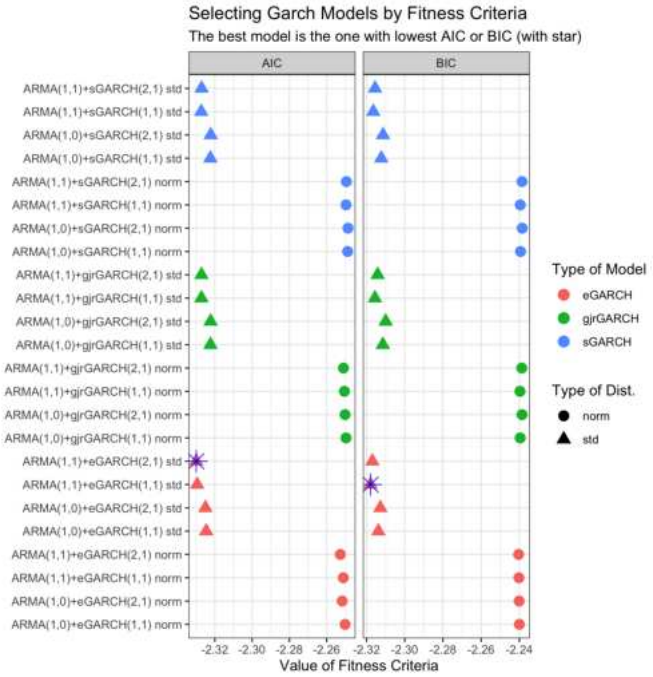
Rysunek A.60. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu złota. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



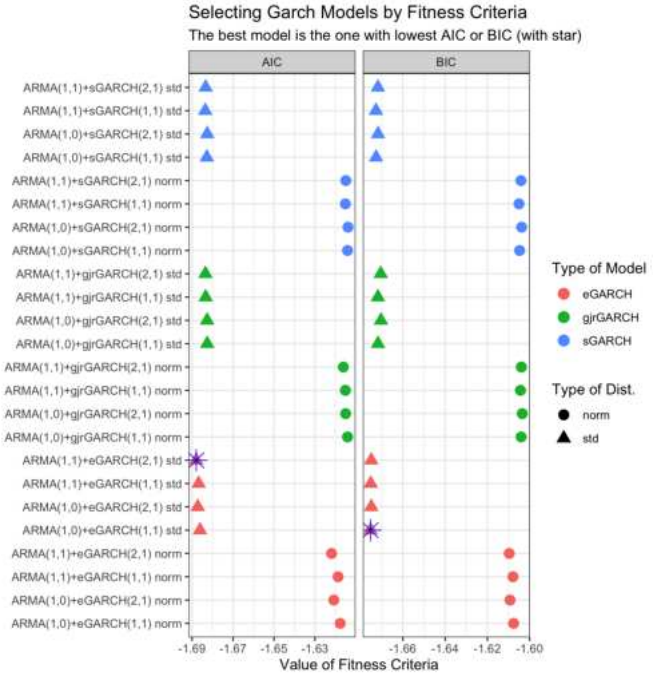
Rysunek A.61. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu srebra. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



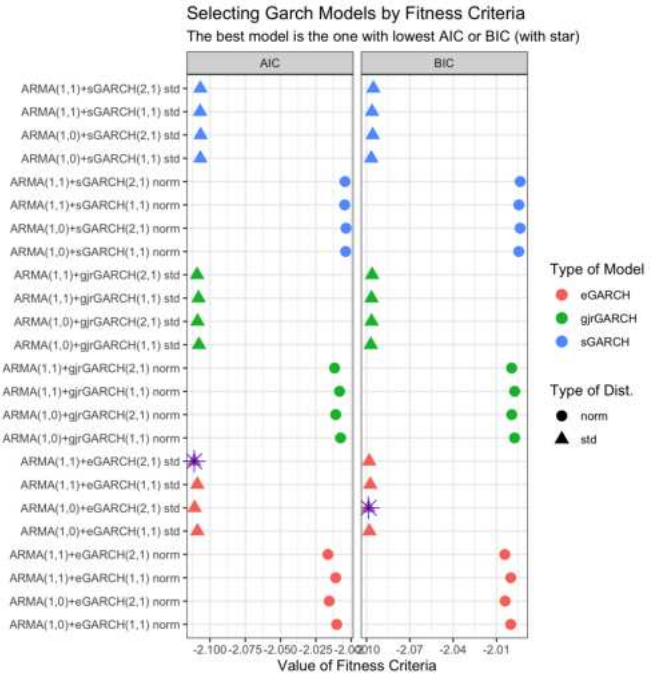
Rysunek A.62. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu Bitcoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



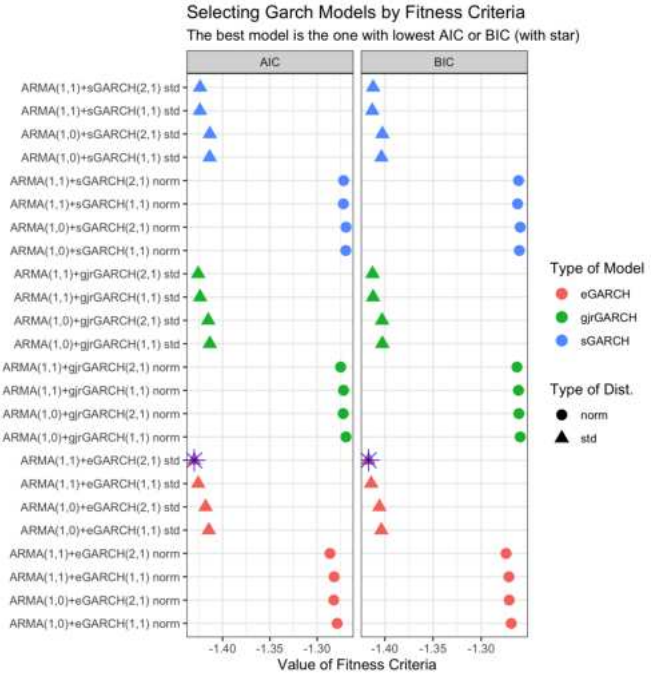
Rysunek A.63. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki Ethereum. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



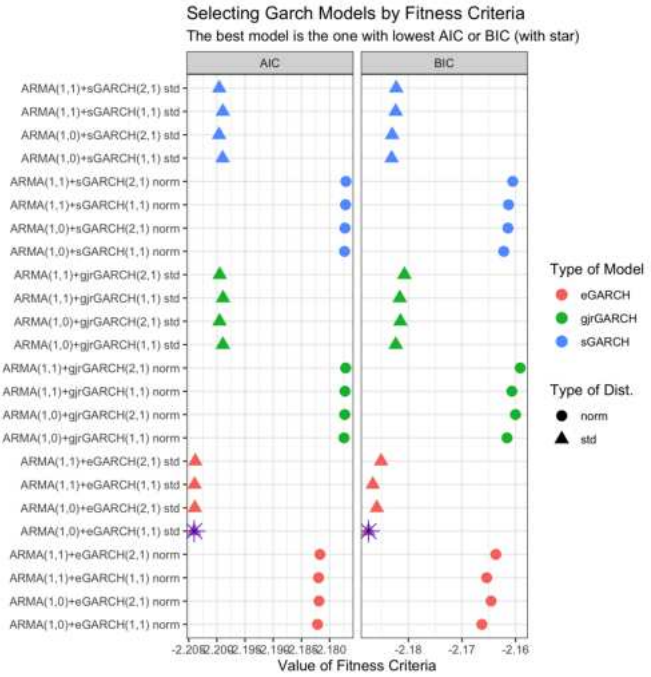
Rysunek A.64. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki Cardano. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



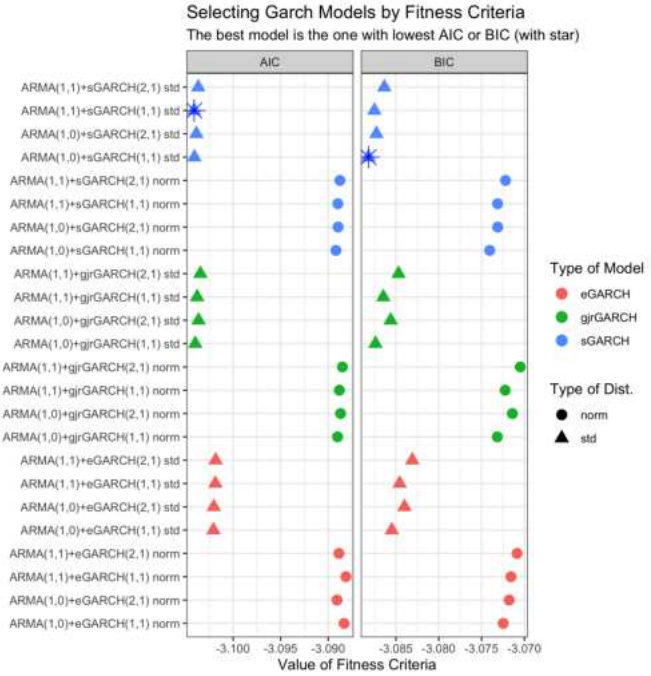
Rysunek A.65. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki Litecoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



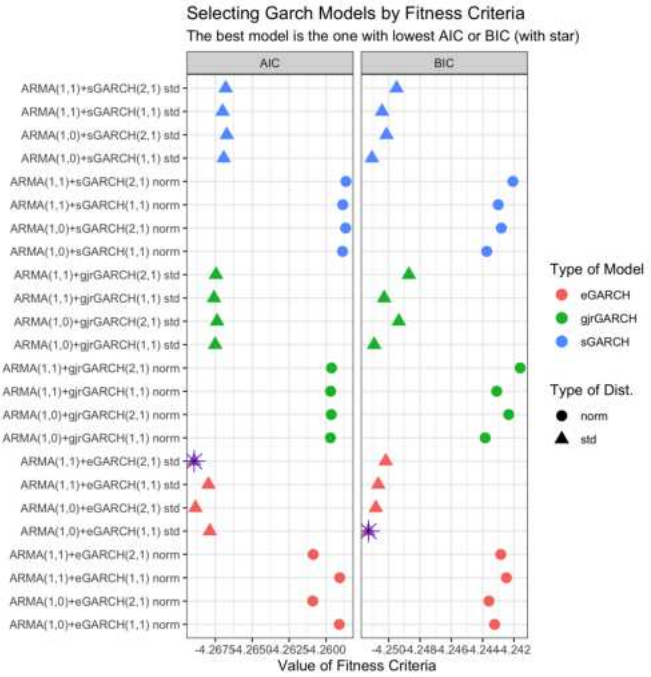
Rysunek A.66. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 8$) dla logarytmicznych stóp zwrotu spółki Dogecoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



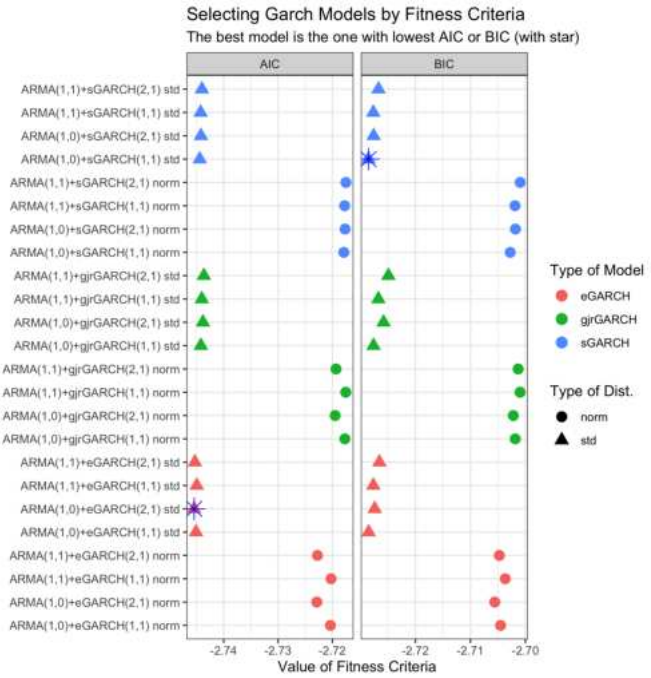
Rysunek A.67. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki TSLA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.68. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki NKE. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

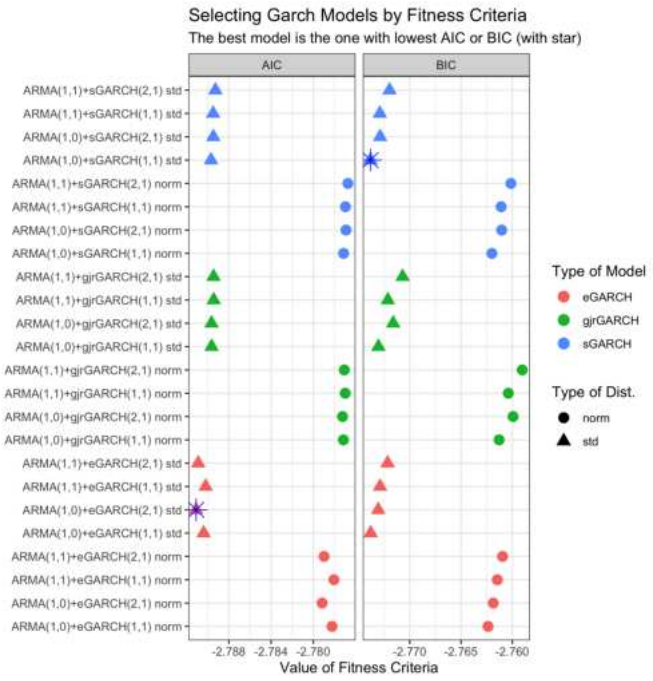


Rysunek A.69. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki KO. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

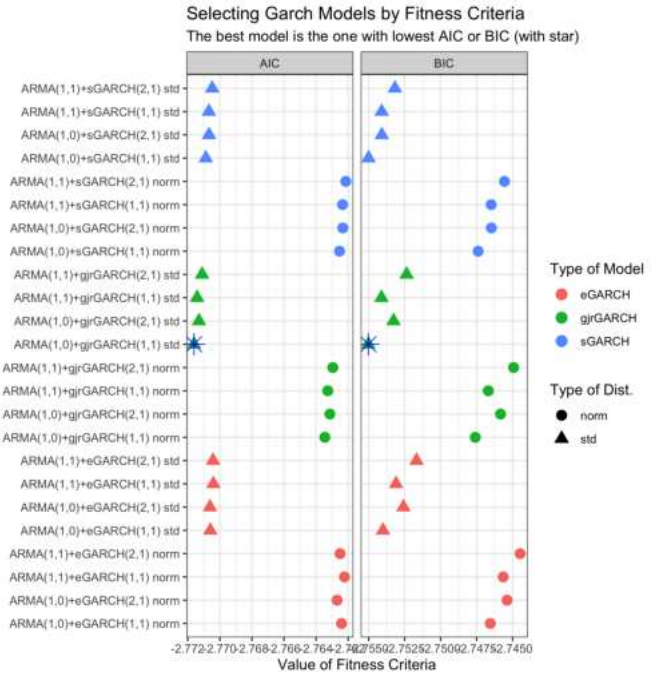


Rysunek A.70. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki BA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

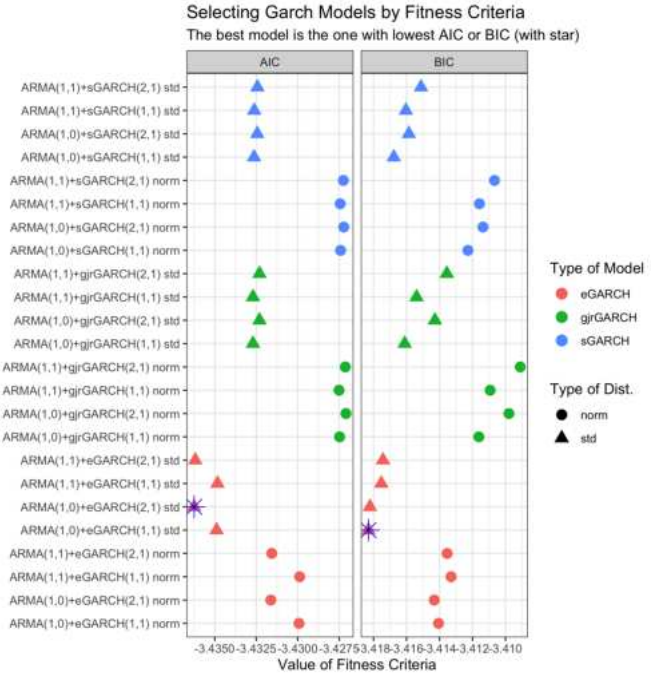
A.3.3. Modele z enkoderem o długości $d_{encoder} = 16$



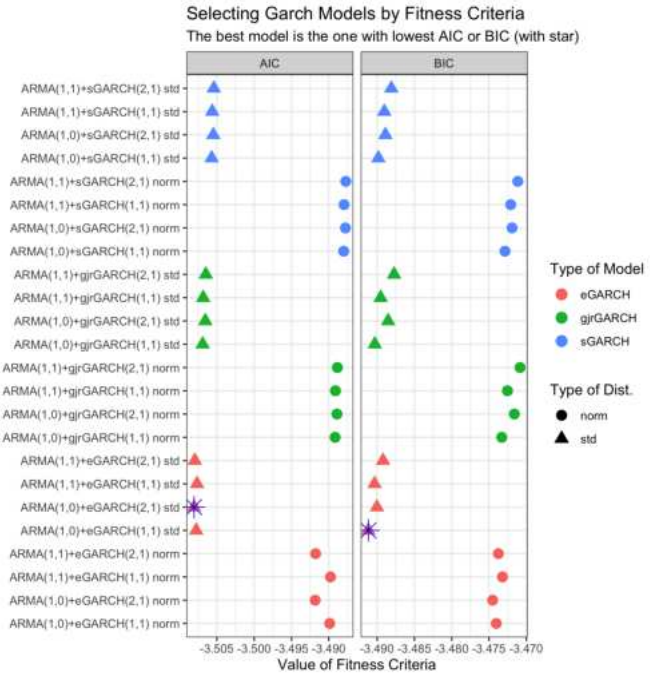
Rysunek A.71. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki CRM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



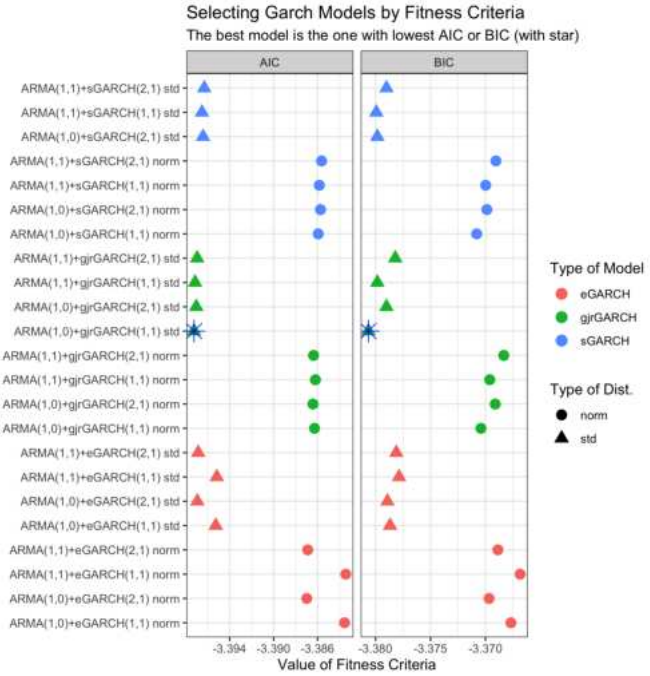
Rysunek A.72. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki GM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



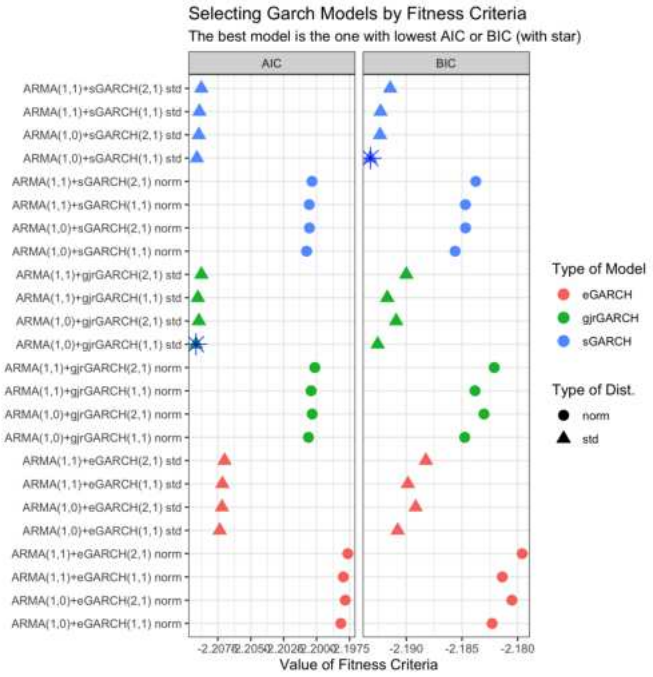
Rysunek A.73. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki JPM. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



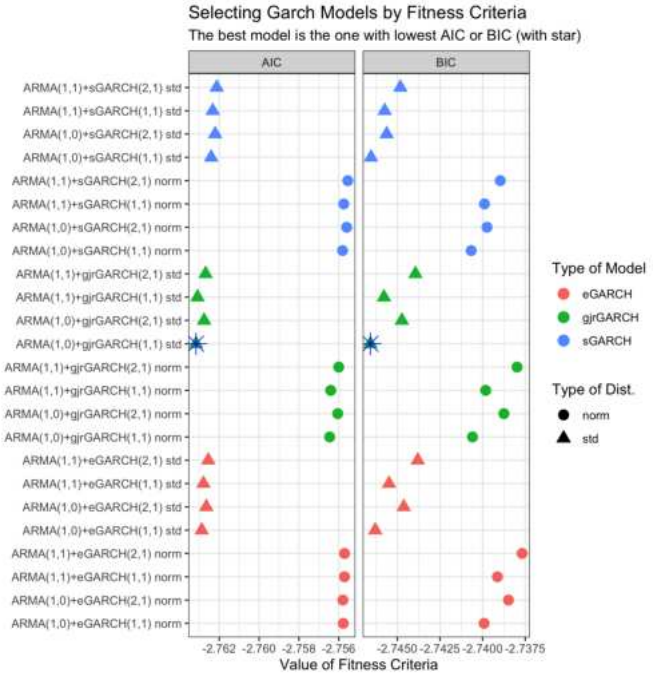
Rysunek A.74. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki MSFT. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



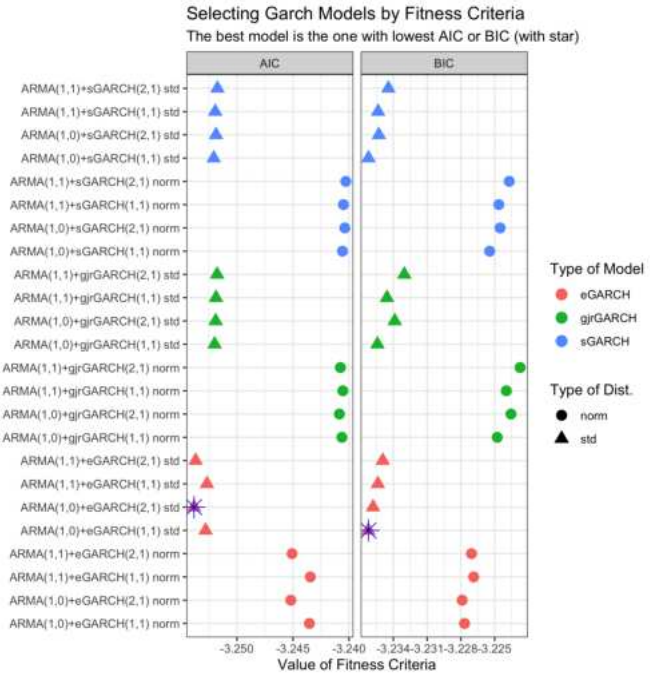
Rysunek A.75. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki GS. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



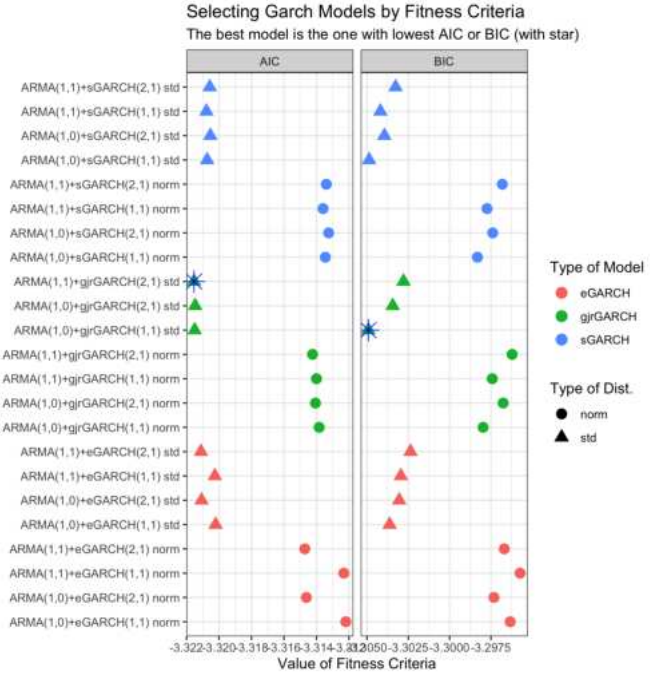
Rysunek A.76. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki NVDA. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



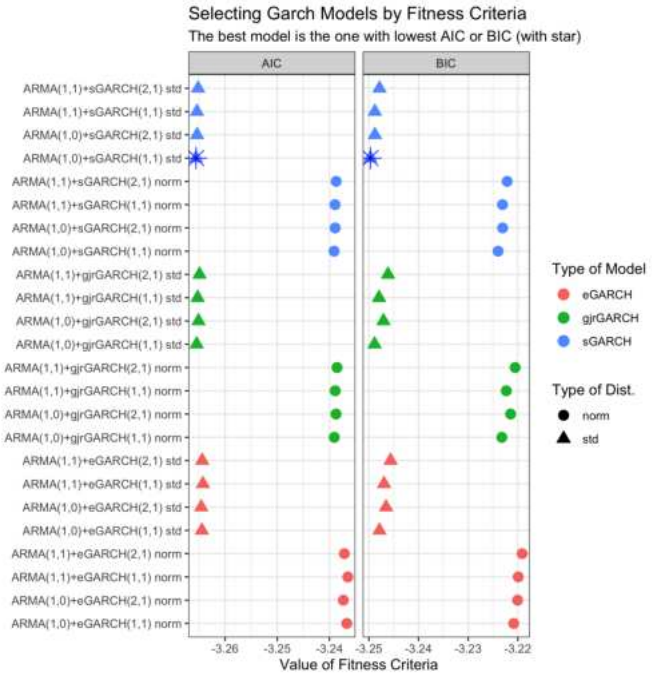
Rysunek A.77. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki F. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



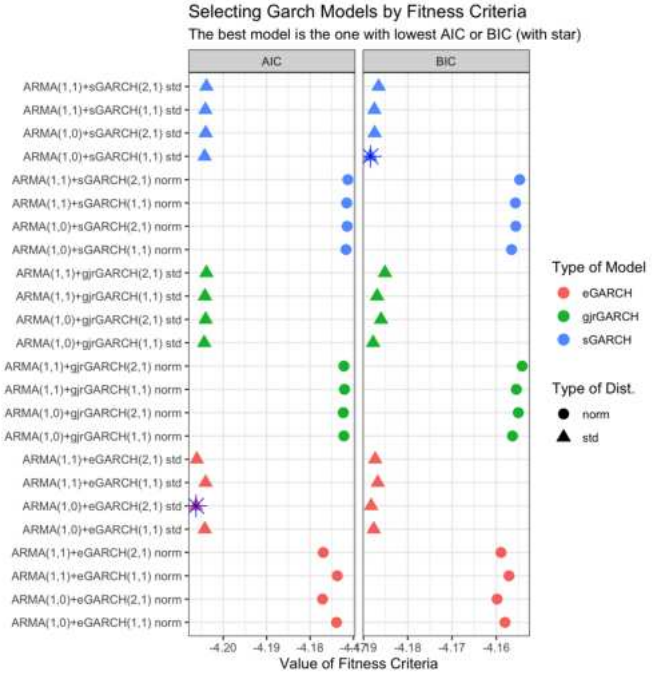
Rysunek A.78. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki C. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



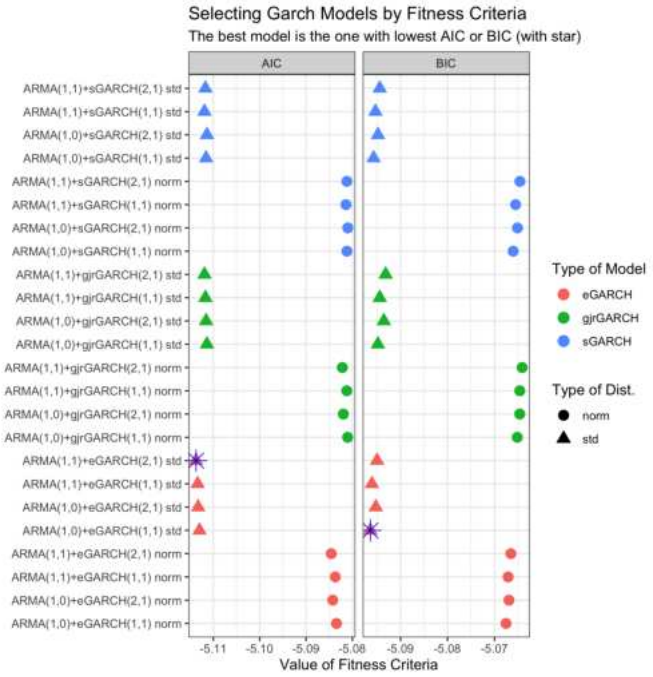
Rysunek A.79. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki MS. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



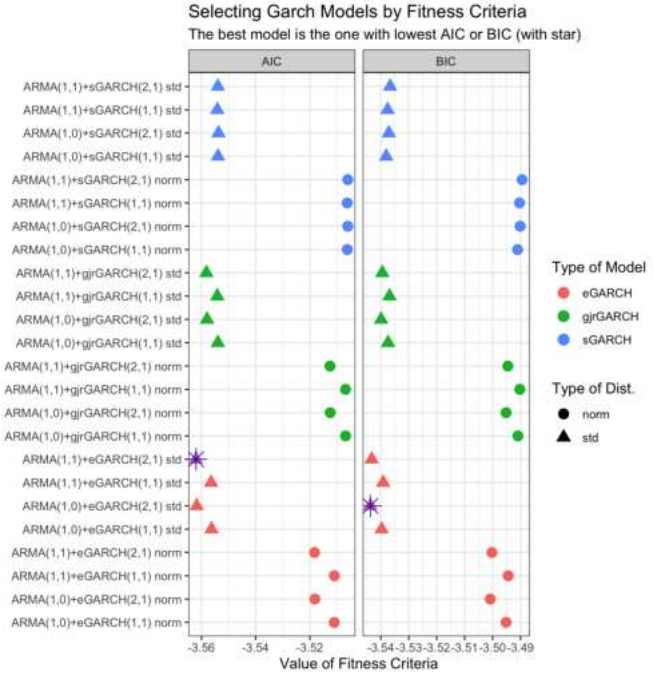
Rysunek A.80. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki INTC. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.81. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki VZ. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

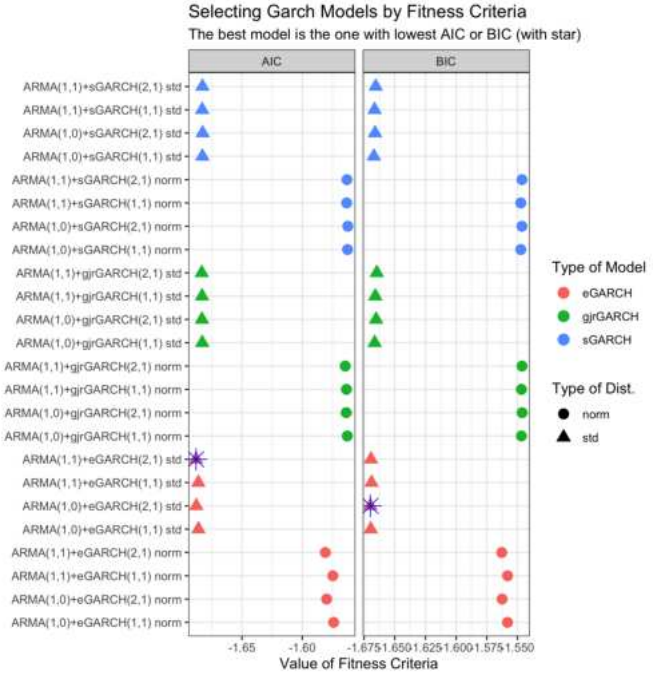


Rysunek A.82. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu złota. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



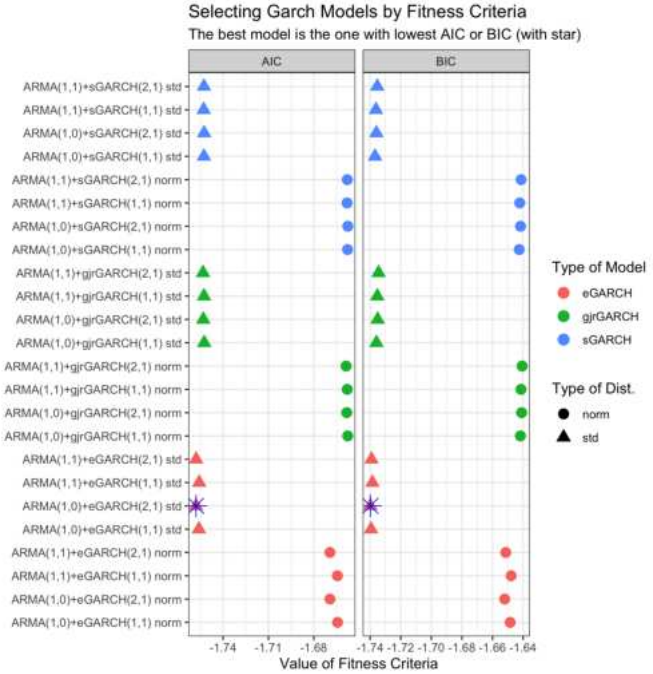
Rysunek A.83. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu srebra. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

Rysunek A.84. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu Bitcoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

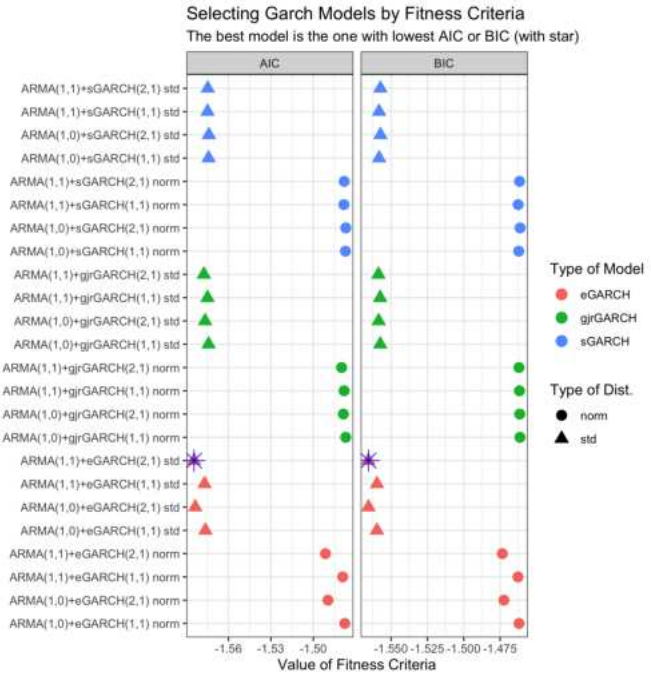


Rysunek A.85. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki Ethereum. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.

Rysunek A.86. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki Cardano. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.87. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki Litecoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów.
Źródło: Obliczenia własne.



Rysunek A.88. Wyniki kryteriów informacyjnych AIC oraz BIC dla różnych specyfikacji i rzędów opóźnień modelu ARMA-GARCH-FENE ($d_{encoder} = 16$) dla logarytmicznych stóp zwrotu spółki Dogecoina. Gwiazdka oznacza najlepszy model dla każdego z kryteriów. Źródło: Obliczenia własne.

A.3.4. Podsumowanie wartości kryterium BIC dla modeli hybrydowych

Tabela A.1. Optymalne rzędy opóźnień, specyfikacje i rozkłady prawdopodobieństwa modeli ARMA-GARCH-FENE ($d_{encoder} = 4$) dla badanych instrumentów finansowych na podstawie kryterium informacyjnego BIC.

	ARMA		GARCH			
Instrument	p	q	q	p	specyfikacja	rozkład
TSLA	1	0	1	1	EGARCH	t-Studenta
NKE	1	0	1	1	**GARCH**	t-Studenta
KO	1	0	1	1	EGARCH	t-Studenta
BA	1	0	1	1	**GARCH**	t-Studenta
CRM	1	0	1	1	**GARCH**	t-Studenta
GM	1	0	1	1	**GJR-GARCH**	t-Studenta
JPM	1	0	1	1	**GARCH**	t-Studenta
MSFT	1	0	1	1	**GJR-GARCH**	t-Studenta
GS	1	0	1	1	**GARCH**	t-Studenta
NVDA	1	0	1	1	**GARCH**	t-Studenta
F	1	0	1	1	**GJR-GARCH**	t-Studenta
C	1	0	1	1	**GARCH**	t-Studenta
MS	1	0	1	1	**GARCH**	t-Studenta
INTC	1	0	1	1	**GARCH**	t-Studenta
VZ	1	0	1	1	**GARCH**	t-Studenta
Złoto	1	0	1	1	EGARCH	t-Studenta
Srebro	1	0	1	1	EGARCH	t-Studenta
Bitcoin	1	1	2	1	EGARCH	t-Studenta
Cardano	1	0	1	1	EGARCH	t-Studenta
Ethereum	1	1	1	1	EGARCH	t-Studenta
Litecoin	1	0	2	1	EGARCH	t-Studenta
Dogecoin	1	1	2	1	EGARCH	t-Studenta

Źródło: Obliczenia własne.

Tabela A.2. Optymalne rzędy opóźnień, specyfikacje i rozkłady prawdopodobieństwa modeli ARMA-GARCH-FENE ($d_{encoder} = 8$) dla badanych instrumentów finansowych na podstawie kryterium informacyjnego BIC.

	ARMA		GARCH			
Instrument	p	q	q	p	specyfikacja	rozkład
TSLA	1	0	1	1	EGARCH	t-Studenta
NKE	1	0	1	1	**GARCH**	t-Studenta
KO	1	0	1	1	**GARCH**	t-Studenta
BA	1	0	1	1	**GARCH**	t-Studenta
CRM	1	0	1	1	**GARCH**	t-Studenta
GM	1	0	1	1	**GJR-GARCH**	t-Studenta
JPM	1	0	1	1	**GARCH**	t-Studenta
MSFT	1	0	1	1	EGARCH	t-Studenta
GS	1	0	1	1	**GARCH**	t-Studenta
NVDA	1	0	1	1	**GARCH**	t-Studenta
F	1	0	1	1	**GJR-GARCH**	t-Studenta
C	1	0	1	1	**GARCH**	t-Studenta
MS	1	0	1	1	**GARCH**	t-Studenta
INTC	1	0	1	1	**GARCH**	t-Studenta
VZ	1	0	1	1	**GARCH**	t-Studenta
Złoto	1	0	1	1	EGARCH	t-Studenta
Srebro	1	0	1	1	EGARCH	t-Studenta
Bitcoin	1	1	2	1	EGARCH	t-Studenta
Cardano	1	0	1	1	EGARCH	t-Studenta
Ethereum	1	1	1	1	EGARCH	t-Studenta
Litecoin	1	0	2	1	EGARCH	t-Studenta
Dogecoin	1	1	2	1	EGARCH	t-Studenta

Źródło: Obliczenia własne.

Tabela A.3. Optymalne rzędy opóźnień, specyfikacje i rozkłady prawdopodobieństwa modeli ARMA-GARCH-FENE ($d_{encoder} = 16$) dla badanych instrumentów finansowych na podstawie kryterium informacyjnego BIC.

	ARMA		GARCH			
Instrument	p	q	q	p	specyfikacja	rozkład
TSLA	1	0	1	1	EGARCH	t-Studenta
NKE	1	0	1	1	**GARCH**	t-Studenta
KO	1	0	1	1	EGARCH	t-Studenta
BA	1	0	1	1	**GARCH**	t-Studenta
CRM	1	0	1	1	**GARCH**	t-Studenta
GM	1	0	1	1	**GJR-GARCH**	t-Studenta
JPM	1	0	1	1	EGARCH	t-Studenta
MSFT	1	0	1	1	EGARCH	t-Studenta
GS	1	0	1	1	**GJR-GARCH**	t-Studenta
NVDA	1	0	1	1	**GARCH**	t-Studenta
F	1	0	1	1	**GJR-GARCH**	t-Studenta
C	1	0	1	1	EGARCH	t-Studenta
MS	1	0	1	1	**GJR-GARCH**	t-Studenta
INTC	1	0	1	1	**GARCH**	t-Studenta
VZ	1	0	1	1	**GARCH**	t-Studenta
Złoto	1	0	1	1	EGARCH	t-Studenta
Srebro	1	0	2	1	EGARCH	t-Studenta
Bitcoin	1	0	1	1	EGARCH	t-Studenta
Cardano	1	1	2	1	EGARCH	t-Studenta
Ethereum	1	0	2	1	EGARCH	t-Studenta
Litecoin	1	0	2	1	EGARCH	t-Studenta
Dogecoin	1	1	2	1	EGARCH	t-Studenta

Źródło: Obliczenia własne.