

Uniwersytet Warszawski  
Wydział Nauk Ekonomicznych

Mateusz Falkus  
Nr albumu: 416260

**Zastosowanie nowoczesnych metod inwestowania na  
przykładzie kontraktu terminowego *futures* na indeks  
NASDAQ-100 w latach 2003 – 2019**

Praca magisterska  
na kierunku: Ekonomia

Praca wykonana pod kierunkiem  
Dr Katarzyny Dąbrowskiej-Gruszczyńskiej,  
z Katedry Bankowości, Finansów i Rachunkowości  
WNE UW

Warszawa, marzec 2021

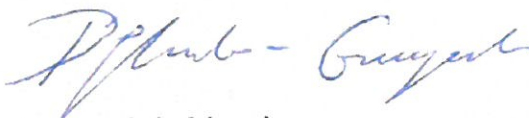
*Oświadczenie kierującego pracą*

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

28.03.2021

Podpis kierującego pracą



*Oświadczenie autora pracy*

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa (magisterska) została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

28.03.2021

Podpis autora pracy



## **Streszczenie**

Praca dotyczy kształtowania się cen na rynku kapitałowym i podejmuje próbę zidentyfikowania czynników wpływających na wartość papierów wartościowych oraz wykorzystania nowych rozwiązań z zakresu sieci neuronowych w celu stworzenia skutecznego systemu transakcyjnego, generującego decyzje inwestycyjne w oparciu o dane historyczne. Pierwszą część pracy stanowi opis tradycyjnych metod, którymi praktycy wspierają się w podejmowaniu decyzji inwestycyjnych. Druga to opis sieci neuronowych – narzędzia, które posłuży do zbudowania systemu transakcyjnego. Zwieńczeniem pracy jest rozdział poświęcony budowie systemów transakcyjnych na kontrakt futures dla indeksu NASDAQ-100 w oparciu o neurony LSTM.

## **Słowa kluczowe**

Sieci neuronowe, systemy transakcyjne, rynek kapitałowy, LSTM, giełda, prognozowanie, statystyka

## **Dziedzina pracy (kody wg programu Socrates-Erasmus)**

Ekonomia (14300)

## **Klasyfikacja tematyczna**

Sieci neuronowe

## **Tytuł pracy w języku angielskim**

The use of modern investment methods on the example of futures contract for NASDAQ-100 between 2003 and 2019.

## Spis treści

<b>WPROWADZENIE</b> .....	4
<b>1. WARTOŚĆ INWESTYCJI NA RYNKU KAPITAŁOWYM</b> .....	6
1.1. Metody analizowania inwestycji .....	6
1.1.1. Analiza portfelowa .....	7
1.1.2. Analiza fundamentalna .....	13
1.1.3. Analiza techniczna .....	17
1.1.4. Wskaźniki analizy technicznej .....	26
1.2. Determinanty cen akcji .....	31
1.2.1. Makroekonomiczne .....	31
1.2.2. Z rynków kapitałowych .....	37
<b>2. SIECI NEURONOWE JAKO NARZĘDZIE INWESTYCYJNE</b> .....	41
2.1. Budowa sieci neuronowych .....	41
2.1.1. Opis podstaw sieci neuronowych .....	42
2.1.2. O danych w sieciach neuronowych .....	46
2.1.3. Trenowanie modeli .....	47
2.1.4. Trudności i wyzwania .....	54
2.2. Rodzaje modeli .....	56
2.2.1. Perceptron wielowarstwowy .....	59
2.2.2. Rekurencyjne sieci neuronowe .....	66
2.2.3. Sieci LSTM .....	71
2.3. Przykłady zastosowania sieci neuronowych w finansach i ekonomii .....	74
<b>3. ZASTOSOWANIE MODELU SIECI LSTM DO BUDOWY SYSTEMU TRANSAKCYJNEGO DLA KONTRAKTU NA INDEKS NASDAQ-100 W LATACH 2003 – 2019 – BADANIE EMPIRYCZNE</b> .....	83
3.1. Opis modelowanego waloru .....	83
3.2. Dane modelu .....	85
3.3. Opis wykorzystanych narzędzi .....	90
3.4. Zmienne modelu .....	91
3.5. Proces budowania sieci neuronowej dla zadania regresji .....	92
3.6. Proces budowania sieci neuronowej dla zadania klasyfikacji .....	108
3.7. Weryfikacja sygnałów inwestycyjnych modeli .....	113
<b>PODSUMOWANIE</b> .....	119
<b>BIBLIOGRAFIA</b> .....	121
<b>ZESTAWIENIE SPISÓW</b> .....	125
<b>ZAŁĄCZNIKI</b> .....	128

## WPROWADZENIE

Czynniki kształtujące ceny akcji i wpływ rynku kapitałowego na gospodarkę od zawsze cieszyły się zainteresowaniem ekonomistów. Zależności pomiędzy wycenami na rynku kapitałowym, a nowymi informacjami dotyczącymi poszczególnych walorów są fundamentami rynku efektywnego, w którym lepiej poinformowana jednostka ma szansę podjąć lepsze decyzje inwestycyjne. Zrozumienie podstaw kształtowania cen na rynku kapitałowym jest więc istotnym zagadnieniem również dla praktyków, a wiedza na temat determinant cen walorów to przewaga informacyjna, która może prowadzić do osiągania ponadprzeciętnych zysków.

Uczestnicy rynku posługują się różnymi podejściami do wyznaczania potencjalnej rentowności inwestycji i ograniczania związanego z lokowaniem kapitału ryzyka. Do tradycyjnych metod zalicza się analizę fundamentalną, analizę portfelową i analizę techniczną (oraz ich kombinacje), ale wraz z postępującą cyfryzacją, podejście do inwestowania stopniowo się zmienia. Powszechnym zjawiskiem staje się handel algorytmiczny, wykorzystujący duże zasoby mocy obliczeniowej do dokonywania błyskawicznych transakcji.

Połączenie rozwoju zastosowań statystyki i prawdopodobieństwa oraz większa dostępność dużej mocy obliczeniowej przyczyniła się do upowszechnienia nauczania maszynowego i sieci neuronowych, które szacują wyniki dzięki iteracyjnemu uczeniu się zależności między danymi. Dzięki łatwemu dostępowi do danych giełdowych i makroekonomicznych, upowszechnia się zastosowanie tych metod również dla prognozowania przyszłych wartości kursów giełdowych. *Data science* to nieustannie rozwijająca się dziedzina nauki, a nowe odkrycia w jej obszarze dają szansę na większą dokładność prognozowania, również wartości walorów na giełdzie. Wśród relatywnie nowych osiągnięć tej dziedziny, dających obiecujące wyniki dla szeregów czasowych, można wymienić sieci rekurencyjne oraz sieci LSTM, które potrafią zapamiętywać zależności pomiędzy oddalonymi od siebie punktami w czasie. Modele bazujące na nich posłużą w tej pracy za nowoczesne narzędzia, na bazie których zbudowany zostanie system transakcyjny.

Każdy inwestor dąży do jak największej skuteczności swojego systemu transakcyjnego. Mimo iż osiągnięcie stuprocentowej (lub nawet zbliżonej do tego progu) skuteczności wydaje się niemożliwe przy wykorzystaniu tylko danych historycznych, to wśród praktyków przyjmuje się, że osiągnięcie skuteczności wyższej niż 50% jest dobrym wynikiem. Sieci neuronowe to

rozwiązanie coraz częściej stosowane do budowania systemów transakcyjnych, dlatego celem niniejszej pracy jest przeanalizowanie ich wykorzystania w generowaniu sygnałów inwestycyjnych.

Z przeglądu literatury wynika, że sieci neuronowe, a zwłaszcza sieci rekurencyjne, wydają się być obiecującym narzędziem do wspomagania decyzji inwestycyjnych. Dlatego hipotezą tej pracy jest stwierdzenie, że sieci zbudowane w oparciu o neurony LSTM, wykorzystujące historyczne dane takie jak: ceny walorów, ceny surowców, kursy walutowe, dane makroekonomiczne oraz wskaźniki analizy technicznej, mogą stanowić skuteczne narzędzie do generowania sygnałów inwestycyjnych.

Praca składa się z trzech rozdziałów – w pierwszym omówione zostaną podejścia inwestorów do analizy inwestycji oraz czynniki kształtujące ceny kursów walorów. W drugim zostaną opisane podstawy sieci neuronowych wraz z zastosowaną w części badawczej architekturą LSTM oraz rezultaty innych prac naukowych badających jej skuteczność w zastosowaniach ekonomicznych. Ostatni rozdział będzie stanowił opis budowy systemów transakcyjnych, opartych na modelach sieci neuronowych i ich wyników.

# 1. WARTOŚĆ INWESTYCJI NA RYNKU KAPITAŁOWYM

## 1.1. Metody analizowania inwestycji

Zrozumienie, jakie czynniki kształtują wartość akcji i potencjał jej wzrostu stanowią podstawę skutecznego lokowania kapitału. Inwestorzy korzystają z szeregu podejść i metod w celu przeanalizowania dostępnych danych, a następnie podjęcia decyzji inwestycyjnej. Zrozumienie zasad działania tych podejść jest istotne dla skutecznego prognozowania zmian kursów giełdowych przy użyciu metod statystycznych. Ten rozdział poświęcony będzie skrótowemu opisowi najczęściej wykorzystywanych technik.

Borowski (2014) wśród metod na budowę systemu transakcyjnego wymienia: **a) analizę techniczną, b) analizę fundamentalną, c) analizę międzyrynkową, d) teorie portfelową, e) modele matematyczne, f) teorię fraktali oraz g) teorię chaosu** jako często wykorzystywane metody analizy inwestycji. Z kolei za najbardziej popularne uważa pierwsze trzy z wyżej wymienionych.<sup>1</sup>

Analiza techniczna, fundamentalna i portfelowa będą omówione w osobnych podrozdziałach, ten rozdział zostanie poświęcony na ogólne omówienie pozostałych podejść. Inwestorzy najczęściej łączą metody dla osiągnięcia bardziej odpornych na błąd jednej metody wyników.

**Analiza międzyrynkowa** zajmuje się badaniem powiązań między różnymi rodzajami rynków finansowych (walutowym, towarowym, akcji, obligacji, opcji itd.). Żaden z tych rynków nie działa w odizolowaniu i wszystkie są ze sobą w większym lub mniejszym stopniu powiązane – często spadek wartości walorów na jednym z nich jest wiąże ze wzrostem wartości walorów na innym. Jednym z przykładów zastosowania analizy międzyrynkowej może być obserwacja wpływu obniżenia stóp procentowych na wartości różnych walorów.<sup>2</sup>

**Modele matematyczne** dzielą się na dwie główne grupy: a) **modele ekonometryczne** wykorzystujące statystykę do prognozowania stóp zwrotu (model opisany w części badawczej tej pracy jest przykładem takiej metody), b) **modele cen akcji** – czyli próba opracowania funkcji,

---

<sup>1</sup> Borowski, K., 2014, Analiza fundamentalna, metody wyceny przedsiębiorstwa, Wyd.: Difin SA, Warszawa s. 13 – 16.

<sup>2</sup> Ibidem.

której zmienną zależną jest cena waloru, a zmiennymi niezależnymi wszystkie wpływające na nią czynniki.<sup>3</sup>

W **teorii chaosu** zakłada się, że istnieje jakiś wzorzec (zwany atraktorem, czyli punktem w przestrzeni stanów, wokół którego oscylują wszystkie kursy walorów), który przez długi czas pozostaje niezmienny i dopiero jego zmiana oznacza koniec dotychczasowego trendu. Z kolei teoria fraktali bazuje na odnajdywaniu samopodobnych struktur na rynkach. Znalezienie i rozpoznanie takiej struktury umożliwiłoby przewidzenie zmian cen walorów.<sup>4</sup>

#### 1.1.1. Analiza portfelowa

Opisywanie analizy portfelowej wymaga przedstawienia pojęcia ryzyka inwestycyjnego, ponieważ według podstaw tego zagadnienia, inwestorzy maksymalizują zysk przy określonym poziomie ryzyka, lub minimalizują ryzyko przy określonym poziomie zysku.

Ryzyko inwestycyjne jest możliwością straty części lub całości zainwestowanego kapitału (a przy wykorzystaniu dźwigni kwot nawet wyższych niż zainwestowane), mimo planowanego osiągnięcia zysków. W sytuacji rynkowej zysk jednego inwestora najczęściej wynika ze straty drugiego, który podjął gorsze decyzje inwestycyjne (zapewne oparte na mniejszej dostępności do informacji lub mniej trafnej interpretacji informacji). Ryzyko na rynku kapitałowym dzieli się na kilka składników. Podstawowym z nich są: **a) ryzyko rynkowe, b) ryzyko finansowe, c) ryzyko inflacyjne, d) ryzyko walutowe.** **Ryzyko rynkowe** to niekorzystne (z punktu widzenia strumienia przyszłych przychodów firmy (której akcje posiada dany inwestor) zmiany gospodarcze lub społeczno-polityczne. Taka zmiana może spowodować obniżenie wartości waloru poniżej poziomu, po którym kupił go inwestor i doprowadzić do realnej straty części lub całości zainwestowanego kapitału (a w przypadkach opcji nawet do utraty wielokrotności zainwestowanego kapitału). **Ryzyko finansowe** można opisać jako kondycję finansową emitenta. Zła kondycja finansowa oznacza często niską wartość sprzedaży produktów/usług emitenta i w dłuższym horyzoncie prawie zawsze prowadzi do obniżenia strumienia przyszłych przychodów, a co za tym idzie, do obniżenia wartości waloru. **Ryzyko inflacyjne** to zagrożenie spadku siły nabywczej pieniądza. Stopień inflacji jest istotnym czynnikiem wyboru odpowiednich instrumentów inwestycyjnych. Wysoka inflacja powiązana jest często z interwencjami banku

---

<sup>3</sup> Ibidem.

<sup>4</sup> Ibidem.



centralnego – na przykład z podniesieniem stopy procentowej. Podniesienie stopy procentowej sprawi z kolei, że inwestorzy będą częściej wybierać inwestycje o niższym ryzyku (jak na przykład obligacje skarbowe lub lokaty bankowe), które po podniesieniu stopy procentowej oferują wyższy zysk. Taki ruch kapitału z kolei spowoduje przecenę akcji na giełdzie. **Ryzyko walutowe** to nagłe zmiany w cenie jednej waluty wyrażanej drugą walutą. Zagraża ono nie tylko inwestorom, którzy lokują swój kapitał w walucie obcej lub inwestują w akcje spółek zagranicznych. Wahania kursów walut mogą doprowadzić także do bankructwa podmioty krajowe, które mają mocne powiązania handlowe z podmiotami zagranicznymi. Każdy typ ryzyka można próbować ograniczyć, na przykład poprzez dywersyfikację i trzymanie walorów z różnych sektorów w swoim portfelu, ale żadna metoda nie jest całkowicie odporna na niespodziewane zdarzenia.<sup>5</sup>

Zbiór aktywów, w które inwestor ulokował pieniądze, nazywany jest często portfelem inwestycyjnym, wybór walorów do portfela to złożony proces w trakcie którego inwestor kieruje się rozmaitymi informacjami i metodami – jedną z nich jest mająca oparcie w statystyce analiza portfelowa. Markowitz (1952) wymienia podstawowe założenia analizy portfelowej: proces inwestycyjny składa się z dwóch części – pierwsza z nich to obserwacja i doświadczenie, które prowadzą do wytworzenia u inwestora oczekiwań co do przyszłej wartości danego aktywa, a druga to zestawienie oczekiwań co do większej ilości aktywów i wyłonienie wśród nich tych, które wejdą w skład portfela inwestycyjnego. Markowitz odrzuca wcześniejszy konsensus co do podstawy decyzji inwestorów, a mianowicie że powinni oni kierować się jedynie maksymalizacją zdyskontowanych zwrotów z inwestycji. Jeżeli inwestorzy kierowaliby się tylko maksymalizacją zwrotu, nie można byłoby udowodnić tego, że zdywersyfikowane portfele są lepsze od niezdywersyfikowanych (a dywersyfikacja jest sensowna oraz obserwowana w zachowaniach inwestorów). W takim wypadku inwestorzy zawsze lokowaliby cały swój kapitał w papier wartościowy z największym oczekiwanym zwrotem, a jeśli dwa aktywa miałyby taki sam oczekiwany zwrot, jakakolwiek kombinacja tych dwóch byłaby tak samo pożądana.<sup>6</sup>

Zamiast tego Markowitz proponuje, że rozważając wybór aktywów do portfela, inwestor kieruje się (oraz powinien się kierować) oczekiwanym zwrotem oraz oczekiwanym ryzykiem, w postaci wariancji danego aktywa. Takie matematyczne podejście pozwala na ujęcie dywersyfikacji

---

<sup>5</sup> Dębski, W., 2014, Rynek finansowy i jego mechanizmy, Wyd. Naukowe PWN, Warszawa. s. 101 – 104.

<sup>6</sup> Markowitz, H., 1952, Portfolio Selection, The Journal of Finance, Vol. 7, No. 1. (Mar., 1952), s. 77 – 78.

oraz prowadzi do wniosku, że istnieje jakiś przelicznik, który przy pewnych portfelach pozwala inwestorowi na zmniejszenie ryzyka przy jednoczesnym zmniejszeniu oczekiwanych zwrotów z inwestycji. Inwestor, przy danych egzogenicznie oczekiwanych stopach zwrotu, wybiera procentowy udział poszczególnych aktywów w swoim portfelu. Co można opisać za pomocą wzoru:

$$R = \sum R_i X_i \quad (1.1)$$

gdzie  $R$  to całkowity zwrot z inwestycji, który równa się sumie iloczynów zwrotów z poszczególnych aktywów z procentowym ich udziałem w portfelu (który sumuje się do 1). Wartość oczekiwana takiego portfela dla  $N$  aktywów będzie wynosiła:

$$E = \sum_{i=1}^N X_i \mu_i \quad (1.2)$$

gdzie  $\mu_i$  to wartość oczekiwania zwrotu z inwestycji w jeden papier wartościowy ( $R_i$ ), przy wariancji równej:

$$V = \sum_{i=1}^N \sum_{j=1}^N \sigma_{ij} X_i X_j \quad (1.3)$$

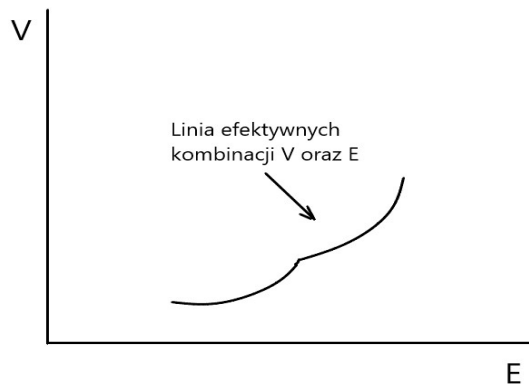
gdzie  $\sigma_{ij}$  jest wartością oczekiwaną iloczynu różnic pomiędzy zwrotem z aktywa  $R_i$  i wartością oczekiwaną zwrotu z  $R_i$  oraz zwrotu z  $R_i$  i wartością oczekiwaną zwrotu z  $R_j$ .<sup>7</sup>

Zależność pomiędzy zwrotem z inwestycji w aktywa, a ich wariancją nie jest liniowa. Możliwe jest więc znalezienie portfela  $X$  o wyższym zwrocie i niższej wariancji niż portfel  $Y$ , ale generalną tendencją rynkową jest, że aktywa o wyższej wariancji mają wyższą możliwą stopę zwrotu (uwzględniają premię za ryzyko) i inwestor powinien szukać możliwych portfeli na linii efektywnej, to jest takiej, która zakłada jak najwyższy zwrot przy jak najniższej zmienności. Dokładne umiejscowienie portfela inwestora na krzywej widocznej na rysunku 1. zależy od preferencji inwestora (minimalizacja ryzyka przy niższym możliwym zysku lub maksymalizacja zysku przy wyższym możliwym ryzyku jako dwa najważniejsze kryteria wyboru).

---

<sup>7</sup> Op. cit., s. 79 – 81.

Rys. 1. Portfele efektywne (optymalne połączenia zmienności i wartości oczekiwanej zwrotów).



Źródło: Opracowanie własne na podstawie Markowitz, H., 1952, Portfolio Selection, The Journal of Finance, Vol. 7, No. 1. (Mar., 1952), s. 88

Markowitz założył, że suma udziałów aktywów wybranych do portfela musi równać się 1 (100%) oraz wykluczył możliwość krótkiej sprzedaży (udział i-tego aktywa X musi być wyższy od 0%) oraz. Taki model i wyjaśnienia lepiej posłużą inwestorowi skupionemu na inwestycji (patrzącemu na dalszy horyzont czasowy) niż takiemu, który zajmuje się spekulacją. Matematyczne wnioski jego pracy nie oznaczają, że zawsze zdywersyfikowany portfel będzie lepszy od niezdywersyfikowanego, ponieważ można sobie wyobrazić sytuację, w której jeden papier wartościowy ma znacznie wyższą stopę zwrotu przy mniejszej zmienności niż wszystkie pozostałe aktywa.<sup>8</sup>

Natomiast w ujęciu ogólnym zasada E-V (oczekiwany zwrot do zmienności), którą zaproponował najczęściej prowadzi do wyłonienia zdywersyfikowanych portfeli jako optymalnych. Markowitz nie tylko propaguje dywersyfikację, ale też sugeruje, jaka dywersyfikacja jest „właściwa”. Ze względu na swoją postać, jego teoria skłania do inwestowania w aktywa o niższym współczynniku korelacji. Właściwa dywersyfikacja nie jest więc tylko kupowaniem wielu różnych aktywów, a portfel składający się z 10 spółek z sektora bankowego nie byłby tak zdywersyfikowany jak portfel tego samego rozmiaru ze spółkami z różnych sektorów gospodarki (górnictwa, bankowości, gier komputerowych, usług cyfrowych itd.). W niektórych sytuacjach korzystnym zabiegiem może być zbudowanie dwóch portfeli z taką samą (niską) wariancją,

<sup>8</sup> Op. cit., s. 82 – 91.

ponieważ wariancja takiego podwójnego portfela powinna być niższa niż wariancja każdego z osobna (przy założeniu niskiej korelacji między portfelami).<sup>9</sup>

Steinbach (2001) wspomina między innymi o wciąż nierozwiązanym zagadnieniu mierzenia ryzyka wymieniając semiwariancję (średnia podniesionych do kwadratu odchyłeń wartości, które są poniżej średniej) jako jedną z powszechniej przyjmowanych miar. Steinbach skupił się na matematycznej analizie wielookresowego modelu, który uwzględniał różne stopy ryzyka danych inwestycji oraz analizował międzyokresowe potencjalne zachowania inwestorów, którzy mogli – wycofać się z inwestycji, zmienić aktywa na takie o mniejszym/większym ryzyku, wypłacić całość lub część zarobionych po pierwszym okresie pieniędzy (a resztę reinwestować). Doszedł on do wniosku, że przy tak przyjętych założeniach modele wielookresowe zachowują się podobnie do jednookresowych, ale pozwalają uniknąć nadmiernej skuteczności (która wiąże się również ze zwiększającym się ryzykiem - semiwariancją) poprzez wypłacanie całości lub części zarobionych po danym okresie pieniędzy. Jeszcze większą efektywność portfela (i spełnienie jego pierwotnych założeń) inwestor może osiągnąć poprzez zainwestowanie części/całości zarobionych pieniędzy w aktywa bez ryzyka lub o bardzo niskim ryzyku. Steinbach podkreśla, że inwestor budując swój portfel powinien rozważyć wszystkie strategie nieobarczone ryzykiem (wraz z wszystkimi portfelami efektywnymi). Aktywa wolne od ryzyka są sensowną częścią portfela, które obniżają jego ogólną semiwariancję oraz zwiększają szansę osiągnięcia zakładanego celu (na koniec wszystkich okresów).<sup>10</sup>

Jorion (2006), nawiązując do pracy Markowitza, skupia się na VAR, czyli *Value at Risk*, co w wolnym tłumaczeniu oznacza wartość przy danym ryzyku. VAR to strategia, polegająca na oszacowaniu maksymalnego ryzyka dla danego portfela (lub jednego aktywa). VAR podsumowuje najgorszą możliwą stratę w danym przedziale czasowym, która nie będzie większa z danym poziomem ufności. VAR opisuje kwantyl dystrybucji zysków i strat dla danego horyzontu czasowego. Jeśli  $c$  jest poziomem ufności, to VAR odpowiada  $1 - c$  dolnego przedziału. W praktyce VAR liczy się poprzez wykorzystanie danych historycznych (zazwyczaj im dłuższy horyzont, tym dokładniejsze obliczenia). Przykładem może być hipotetyczny inwestor, który w swoim portfelu ma 100 milionów dolarów w średnioterminowych obligacjach. Żeby policzyć VAR należy

---

<sup>9</sup> Ibidem.

<sup>10</sup> Steinbach, M. C., 2001, Markowitz Revisited: Mean-Variance Models in Financial Portfolio Analysis, *SIAM Rev.*, 43(1), s. 32 – 73.

sprawdzić, jak duże były miesięczne wahania w jakimś okresie. Założono, że maksymalnie było to -5% w dół oraz +5% w górę. Następnie należy zbudować histogram tych wartości. Kolejnym krokiem jest wybranie poziomu ufności – w świecie finansów często wybiera się 99%. W związku z tym możemy odrzucić 1% największych ujemnych wahań. Założmy, że po odrzuceniu tych wartości najniższa pozostała wartość to -3,6%. W tym przykładzie wniosek jest taki, że z 99% pewnością wartość portfela w ciągu miesiąca nie spadnie o więcej niż 3,6% (dla tego przykładu – 3,6 miliona dolarów).<sup>11</sup>

VAR może okazać się przydatnym narzędziem dla inwestorów, którzy w prosty sposób chcą sprawować kontrolę nad podejmowanym ryzykiem wiążącym się z daną inwestycją. Istnieje kilka wariantów obliczania tego współczynnika, ale bazową wersję można policzyć korzystając z następujących kroków: a) oszacowanie obecnej wartości inwestycji, b) zmierzenie zmienności czynnika ryzyka (istnieją rozmaite możliwe metody obliczania ryzyka), c) ustalenie horyzontu czasowego (ściśle związane z tym, ile czasu papier wartościowy będzie znajdować się w portfelu), d) wybór poziomu ufności, e) Po odliczeniu skrajnych wartości nie mieszczących się w wybranym procencie ufności, podanie najwyższej możliwej straty.<sup>12</sup>

Jorion uważa, że VAR jest statystyczną miarą ryzyka, która opisuje dystrybucję zwrotów jedną liczbą. VAR ma większość cech, które powinna mieć miara ryzyka – monotoniczność; *translation invariance* – dodanie gotówki w wysokości  $k$  do portfela powinno zmniejszyć jego ryzyko o  $k$ ; homogeniczność – zwiększenie rozmiaru portfela o  $b$  razy powinno skutkować wyskalowaniem ryzyka o tę samą wartość; subaddytywność – ryzyko dwóch portfeli połączonych w jeden nie powinno być wyższe niż w przypadku sumy ryzyka tych samych portfeli działających osobno.

VAR może służyć także jako punkt odniesienia w ocenie ryzyka. Większe instytucje często wykorzystują różne metody oceny ryzyka i porównują je z VAR.<sup>13</sup> Ponadto, VAR może być użyty przed inwestora do oceny rentowności przedsięwzięcia przy danym ryzyku. To znaczy, że jeśli inwestor oczekuje, że dany walor w najbliższym roku zwiększy swoją wartość o 5%, a z analizy VAR wynika, że największa strata, jaką może ponieść inwestor na tym papierze wartościowym w

---

<sup>11</sup> Jorion, P., 2006, Value at Risk: The New Benchmark for Managing Financial Risk, 3rd Edition, Wyd.: The McGraw-Hill Companies Inc., Nowy Jork, s. 17 - 20.

<sup>12</sup> Op. cit., s. 107 – 108.

<sup>13</sup> Op. cit., s. 113 – 115.

ciągu roku wynosi 3,5% z 99% poziomem ufności, to inwestycja jest rentowna (zakładając, że stopa wolna od ryzyka jest niższa niż 5%).

### 1.1.2. Analiza fundamentalna

Analiza fundamentalna to zbiór narzędzi, który inwestorzy wykorzystują, chcąc oszacować rentowność z potencjalnej inwestycji (zwłaszcza przy długoterminowej inwestycji). Analiza fundamentalna pozwala przybliżyć odpowiedź na pytanie – o ile więcej będzie warta spółka w wyznaczonym horyzoncie czasowym.

Poznanie rzeczywistej wartości akcji jest podstawą dobrej inwestycji i pozwala na ocenę, czy cena rynkowa jest ceną zgodną z wartością przedsiębiorstwa. Takie podejście ma wielu inwestorów, w tym Warren Buffet. Analiza fundamentalna składa się z wielu składników – niektóre czynniki, które się ocenia mają charakter ilościowy (na przykład bilanse, sprawozdania finansowe, zyski, dywidendy) i można poddać je statystycznej analizie, inne z kolei mają charakter jakościowy i trudniej nadać im wartość liczbową (na przykład kwalifikacje kierownictwa firmy oraz istota działalności przedsiębiorstwa). Analizę fundamentalną dzieli się na trzy główne obszary:

- Analiza czynników związanych z otoczeniem gospodarczym kraju i branżą z którą związana jest spółka
- Analiza finansowa
- Obliczenie wartości wewnętrznej akcji

**Analiza czynników związanych z otoczeniem gospodarczym i branżowym** polega na zbadaniu elementów przedsiębiorstwa wpływających na postrzeganie go przez otoczenie (inwestorów, akcjonariuszy, dostawców itd.). Te czynniki są istotnymi składowymi analizy, ale są jednocześnie trudno mierzalne za pomocą matematyki czy ekonometrii. **Analiza finansowa** polega na badaniu sytuacji finansowej przedsiębiorstwa oraz kalkulacji wskaźników ekonomicznych, takich jak płynności, zadłużenia, zyskowności itd. **Obliczanie wartości wewnętrznej (realnej) akcji** to próba oszacowania właściwej ceny, po jakiej powinny sprzedawane być akcje przy użyciu rozmaitych sposobów – na przykład metody zdyskontowanych przepływów pieniężnych.<sup>14</sup>

Analiza fundamentalna to pojęcie szerokie, ponieważ powyższe trzy jej elementy można osiągnąć na wiele różnych sposobów i przy wykorzystaniu wielu różnych metod, zaliczają się do

---

<sup>14</sup> Borowski, Op. cit, s. 9 – 11.

nich na przykład metody księgowe, mnożnikowe, dochodowe, opcyjne i mieszane. Analiza fundamentalna zmienia się wraz z rynkiem, o czym świadczy chociażby fakt, że w literaturze naukowej pojawia się coraz więcej metod wycen akcji.<sup>15</sup> Ze względu na ograniczony format tej pracy, nie będę zajmował się tutaj omawianiem wszystkich metod, których inwestor może użyć jako swoich narzędzi, a jedynie nakreślę ogólny charakter zastosowań analizy fundamentalnej i zwrócę uwagę na jego praktyczność.

Badanie czynników finansowych i niefinansowych spółek kończące się wyceną realnej wartości ich akcji i wydania rekomendacji: kup, sprzedaj, trzymaj można podzielić na 5 etapów:

- Analiza makroekonomiczna
- Analiza sektorowa (branżowa)
- Analiza sytuacji spółki
- Analiza finansowa
- Wycena akcji

**Analiza makroekonomiczna** polega na zdiagnozowaniu obecnej sytuacji rynkowej, gospodarczej, politycznej i społecznej z punktu widzenia inwestowania w danym kraju oraz z uwzględnieniem zmian, które mogą nastąpić w zakładanym horyzoncie inwestycyjnym. W analizie makroekonomicznej zastosowanie odnajdują podstawowe wskaźniki makroekonomiczne, takie jak: PKB, tempo wzrostu PKB, zadłużenie kraju względem PKB, inflacja czy stopa bezrobocia. Na wszystkie te czynniki, poza sytuacją gospodarczą na całym świecie, wpływają między innymi polityka podatkowa oraz polityka pieniężna banku centralnego. Im lepsza sytuacja gospodarcza w danym kraju, tym większego wzrostu cen walorów można się spodziewać.<sup>16</sup>

**Analiza sektorowa** skupia się na ocenie atrakcyjności inwestowania w walory z danej branży. Uwzględnia się przede wszystkim historyczne zwroty z inwestycji w branży oraz oczekiwane stopy zwrotu w przyszłości, ryzyko wiążące się z sektorem, wrażliwość sektora na sytuację gospodarczą oraz położenie sektora na mapie cyklu danej branży – start, przyspieszenie wzrostu, dojrzałość, zanik sektora (dla przykładu - można się spodziewać, że branża naftowa będzie powoli ustępowała branży odnawialnych źródeł energii). Przeprowadzając analizę

---

<sup>15</sup> Ibidem.

<sup>16</sup> Op. cit., s. 30 – 32.

fundamentalną najczęściej rozpatruje się trzy scenariusze dla danej branży – optymistyczny, neutralny i pesymistyczny.<sup>17</sup>

Analizę sektora oraz analizę cyklu życia sektora można przeprowadzić przy użyciu modelu pięciu sił konkurencyjnych Portera, który wskazuje pięć głównych czynników, dzięki którym można określić charakter i rentowność branży:

- możliwość wejścia na rynek nowych producentów
- siła przetargowa nabywców
- siła przetargowa dostawców
- pojawienie się nowych substytutów
- poziom konkurencyjności danej branży

**Możliwości wejścia na rynek** zależą od wysokości bariery wejścia (początkowych nakładów finansowych oraz barier prawnych) i od reakcji dotychczasowych konsumentów produktów na produkty innych producentów. **Siła przetargowa nabywców** – niewielka liczba nabywców lub ich silna przewaga konkurencyjna mogą stanowić utrudnienie dla firmy w sektorze. Silni nabywcy mogą wywierać naciski na upusty cenowe lub negocjację warunków aż do poziomu poniżej opłacalności. **Siła przetargowa dostawców** – możliwość wymuszenia na firmie wysokich opłat za dostarczane surowce czy usługi, prowadząc do niekonkurencyjności analizowanej firmy. **Pojawienie się nowych substytutów** – może stanowić zagrożenie dla firm działających w sektorze i wymusić na nich większą konkurencyjność (obniżenie cen i utratę przychodów). **Poziom konkurencyjności danej branży** – im więcej konkurencyjnych bytów, tym większa szansa na spadek rentowności sektora, poprzez ciągłe zaniżanie cen.<sup>18</sup>

Z kolei **analiza sytuacyjna** spółki oscyluje wokół konkretnego, analizowanego przedsiębiorstwa. Ocenia się głównie kryteria niefinansowe, czyli: strategię przedsiębiorstwa, sytuację na tle sektora, bariery wejścia, dalsze możliwości rozwoju, liczbę dostawców i odbiorców, jakość kadry zarządczej, kapitał intelektualny i wiele innych, często jakościowych zmiennych. Podchodząc do tego etapu należy inwestor rozważyć dość szeroki horyzont danych historycznych (żeby uzyskać lepszy obraz zmian) oraz przewidzieć kierunek zmian tych czynników w przyszłości.

---

<sup>17</sup> Ibidem.

<sup>18</sup> Op. cit., s. 34 – 38.



Borowski (2014) jako dobrą praktykę podaje określenie tempa wzrostu firm z danego faktora i wyłonienie kilku faworytów o najbardziej przyspieszającym tempie wzrostu. Skutecznym narzędziem do przeprowadzania tego etapu analizy fundamentalnej jest analiza SWOT (zestawienie słabych i mocnych stron oraz szans i zagrożeń danego przedsiębiorstwa).<sup>19</sup>

**Analiza finansowa** spółki koncentruje się na zmiennych ilościowych i daje się przeprowadzić za pomocą wielu modeli opartych o założenia statystyczne i wyliczenia matematyczne. Na tym etapie inwestor bierze pod uwagę dane pochodzące ze sprawozdań finansowych – bilansu, rachunku zysków i strat, rachunku *cash flow*. Analiza przepływów pieniężnych i sprawdzania ich salda odgrywa szczególnie ważną rolę i pozwala dobrze oszacować sytuację finansową przedsiębiorstwa. Analiza finansowa posługuje się wieloma wskaźnikami opracowanymi na przestrzeni lat przez ekonomistów. Należą do nich wskaźniki określające poziom: płynności (zdolności do wywiązywania się z bieżących zobowiązań), zadłużenia (analiza długoterminowych zobowiązań przedsiębiorstwa i oceny stopnia zastosowanej dźwigni finansowej), aktywności (ocena zarządzania kapitałem), zyskowności (ocena osiągniętych dochodów spółki względem ich kosztów), wartości rynkowej.<sup>20</sup> Uwzględnienie poszczególnych wskaźników może okazać się istotne dla modelu mającego prognozować kurs akcji konkretnej spółki – w przypadku tej pracy nie będą miały bezpośredniego zastosowania.

Ostatni etap to właściwa **wycena akcji**. Jej podstawowym zadaniem jest oszacowanie wartości wewnętrznej akcji (określanej także jako *fair value*). Wartość wewnętrzną akcji porównuje się z ceną papieru wartościowego na giełdzie i wydaje się rekomendację kupuj, jeśli wartość wewnętrzna akcji jest wyższa niż jej obecna cena oraz rekomendację sprzedaj w przeciwnym wypadku. Istnieje wiele sposobów obliczania wartości wewnętrznej akcji i wiele z nich korzysta przede wszystkim z danych ilościowych.<sup>21</sup> Dębski (2014) sugeruje, że dominujące metody praktycznej wyceny wartości akcji to metody dochodowe, do których można zaliczyć: **a) modele dyskontowe** (odpowiadające na pytanie jaka jest teraźniejsza wartość akcji po uwzględnieniu przyszłych przychodów danego przedsiębiorstwa) oraz **b) modele empiryczno-indukcyjne** określające wartość akcji opierając się o modele regresyjne w którym czynniki

---

<sup>19</sup> Ibidem.

<sup>20</sup> Ibidem.

<sup>21</sup> Op. cit., s. 38 – 43.

ekonomiczne opisują zmienną zależną, czyli cenę akcji<sup>22</sup> (do takich modeli można zaliczyć również model zaproponowany w badawczej części tej pracy).

Dębski uważa, że modele dyskontowe opierają się na założeniu oczekiwanych korzyści, które może przynieść inwestycja w daną akcję. Korzyści w tym przypadku odnoszą się do wysokości dywidendy i zysku osiąganego przez wzrost ceny rynkowej akcji. W najprostszym ujęciu matematycznie można zapisać realną wartość akcji za pomocą wzoru:

$$\widehat{P}_0 = \sum_{t=1}^n K_t / (1 + r)^t \quad (1.4)$$

gdzie  $\widehat{P}_0$  oznacza realną wartość akcji w danym okresie,  $K_t$  wartość przewidywanych dochodów w poszczególnych okresach K, a  $r$  stopę dyskontową.<sup>23</sup>

### 1.1.3. Analiza techniczna

Analiza techniczna to jedno z podstawowych narzędzi analitycznych służących do sprawdzania potencjalnej opłacalności inwestycji. Założeniem analizy technicznej jest to, że skoro ceny papierów wartościowych są wypadkową oczekiwań inwestorów, to informacje otrzymywane z innych sposobów analiz są mniej ważne niż wiedza o oczekiwaniach innych inwestorów co do danego papieru wartościowego. Analiza techniczna jest procesem analizy historycznego kształtowania się cen papierów wartościowych w celu próby przewidzenia przyszłych cen tego waloru. Posługując się tą metodą, zakłada się, że przyszłości można nauczyć się z przeszłości. Analiza techniczna najczęściej przeprowadzana jest dla krótkiego horyzontu czasowego – w długim horyzoncie lepsze podstawy ma analiza fundamentalna.<sup>24</sup>

W związku z tym można stwierdzić, że dla modelu przewidującego cenę waloru następnego dnia istotniejsze będą wskaźniki przeprowadzonej analizy technicznej niż dokładne dane finansowe spółek, których ceny akcji są przewidywane.

Inwestorzy posługujący się analizą techniczną zakładają, że zjawiska zachodzące na giełdzie wyprzedzają w czasie (i kształtują) zjawiska ekonomiczne, a dzięki rynkowi kapitałowemu możemy zdyskontować przyszłość. Cena rynkowa waloru jest wypadkową działania wszystkich

---

<sup>22</sup> Dębski, Op. cit., s. 222 – 223.

<sup>23</sup> Ibidem.

<sup>24</sup> Tarczyński, W., 2001, Rynki Kapitałowe: Metody Ilościowe – Vol. I, wyd. 2, Wyd.: Agencja Wydawnicza „PLACET”, Warszawa, s. 45 – 49.

uczestników giełdy, bazującą na podstawowych rynkowych założeniach – cena wzrasta, jeśli na walor jest popyt i ograniczona podaż i spada, gdy popyt maleje, a podaż się zwiększa. Jeśli spełnione jest założenie o równym dostępie uczestników giełdy do informacji, wahania kursów zawierają już wpływ ze wszystkich dostępnych informacji. W związku z tym bieżące ceny, na podstawie których bazuje się analizę techniczną, odzwierciedlają czynniki mające wpływ na zmianę popytu i podaży na akcje spółek. Analiza techniczna często daje możliwość zareagowania na takie zmiany, zanim jeszcze jakakolwiek inna analiza pozwoliłaby wychwycić dokładny powód zmian i dostosować do niego działanie inwestycyjne.<sup>25</sup>

Wspomaganie się analizą techniczną często daje niejednoznaczne wyniki, w związku z mnogością metod, które czasem zalecają sprzeczne decyzje inwestycyjne co do jednego waloru. W związku z tym posługując się analizą techniczną inwestor powinien uwzględnić następujące zasady:

- Akcja jest dobrem rynkowym i jej cena kształtuje się tak, jak każdego innego dobra na rynku – przez relację pomiędzy popytem a podażą.
- Oprócz czynników racjonalnych, w wycenie akcji należy także uwzględnić czynniki irracjonalne.
- Ceny waloru podążają za generalnym trendem, który trwa przez jakiś czas.
- Trendy i ich zmiany zdarzają się w powtarzalnych cyklach w podobnej formie.
- Wykres zmiany cen akcji jest źródłem informacji o zmianie popytu i podaży.
- Podstawą analizy technicznej jest trend bazowy, który utrzymuje się przez co najmniej kilkanaście lat (na który nakładają się trzy rodzaje trendów: pierwotny, wtórny i trzeciorzędowy).
- Cena akcji jest ściśle skorelowana z jej obrotem.
- Metody analizy technicznej znacznie lepiej sprawdzają się dla walorów o dużej płynności.
- Nie powinno się podejmować decyzji na podstawie tylko jednego sygnału zakupu/sprzedaży – dopiero potwierdzenie w postaci zgodnych sygnałów pochodzących z

---

<sup>25</sup> Murphy, J. J., 1999, *Technical Analysis of the Financial Markets. A Comprehensive Guide to Trading Methods and Applications*, New York Institute of Finance, Nowy Jork, s. 1 – 3.

innych wskaźników analizy technicznej lub fundamentalnej stanowią racjonalną podstawę decyzji.

- Rynek kapitałowy zachowuje się zgodnie z ekonomicznym prawem popytu i podaży, a najważniejsze zmiany na nim zachodzące są powtarzalne w pewnych wzorcach, które nazywa się formacjami.
- Na rynek oddziałują zjawiska losowe, które mogą skutecznie zagłuszyć analizę techniczną (na przykład nadspodziewanie wysokie/niskie zyski spółki, nagłe ogłoszenie bardzo intratnego kontraktu itd.).<sup>26</sup>

Podstawę analizy technicznej stanowią wykresy, gdzie na osi OY nanosi się cenę waloru, a na osi OX daty notowań w równych odstępach. Inwestorzy skupiają się na analizie wzrostu (spadku) wartości waloru, dlatego wykres w liniowej skali może być czasem mylący. Jeżeli na osi OY naniesione są ceny walorów w wartościach nominalnych, to bardziej widoczne są mniejsze (w ujęciu procentowym) zmiany dla wyższych wartości, niż większe (w ujęciu procentowym) zmiany dla niższych wartości. W celu lepszej obserwacji stopy wzrostu wartości waloru można zastąpić wartości liniowe wartościami skali logarytmicznej. Inwestor korzystający z analizy technicznej powinien wykorzystywać na tyle długi horyzont czasowy, żeby dostrzec powtarzające się formacje, ale nie zbyt długi, który mógłby być niemiarodajny. Zazwyczaj rozważa się dwuletni lub roczny okres czasu.<sup>27</sup>

Rozpatrując samą cenę waloru dla danej daty uwzględnia się, że składa się ona z kilku czynników – ceny otwarcia w dniu sesyjnym, ceny zamknięcia, ceny maksymalnej (podczas danej sesji), minimalnej, oferowanej i pożądanej. Do tego obserwuje się towarzyszący im wolumen obrotu. Praktycznym narzędziem jest wykres świecowy, który pozwala zobrazować między innymi cenę maksymalną i minimalną dla danej sesji.<sup>28</sup> Jednakże z punktu widzenia tej pracy nie jest to najistotniejsze zagadnienie, ponieważ sieć neuronowa analizuje te zależności na podstawie dostarczonych danych liczbowych, a nie wizualnej analizy wykresów.

Z punktu widzenia analizy technicznej, jednymi z najistotniejszych pojęć są: trend (wyróżniamy trend rosnący, malejący i boczny) oraz kanał trendu. Trend to linia, która wyznacza

---

<sup>26</sup> Tarczyński, Op. cit., s. 45 – 49.

<sup>27</sup> Op. cit., s. 50 – 57.

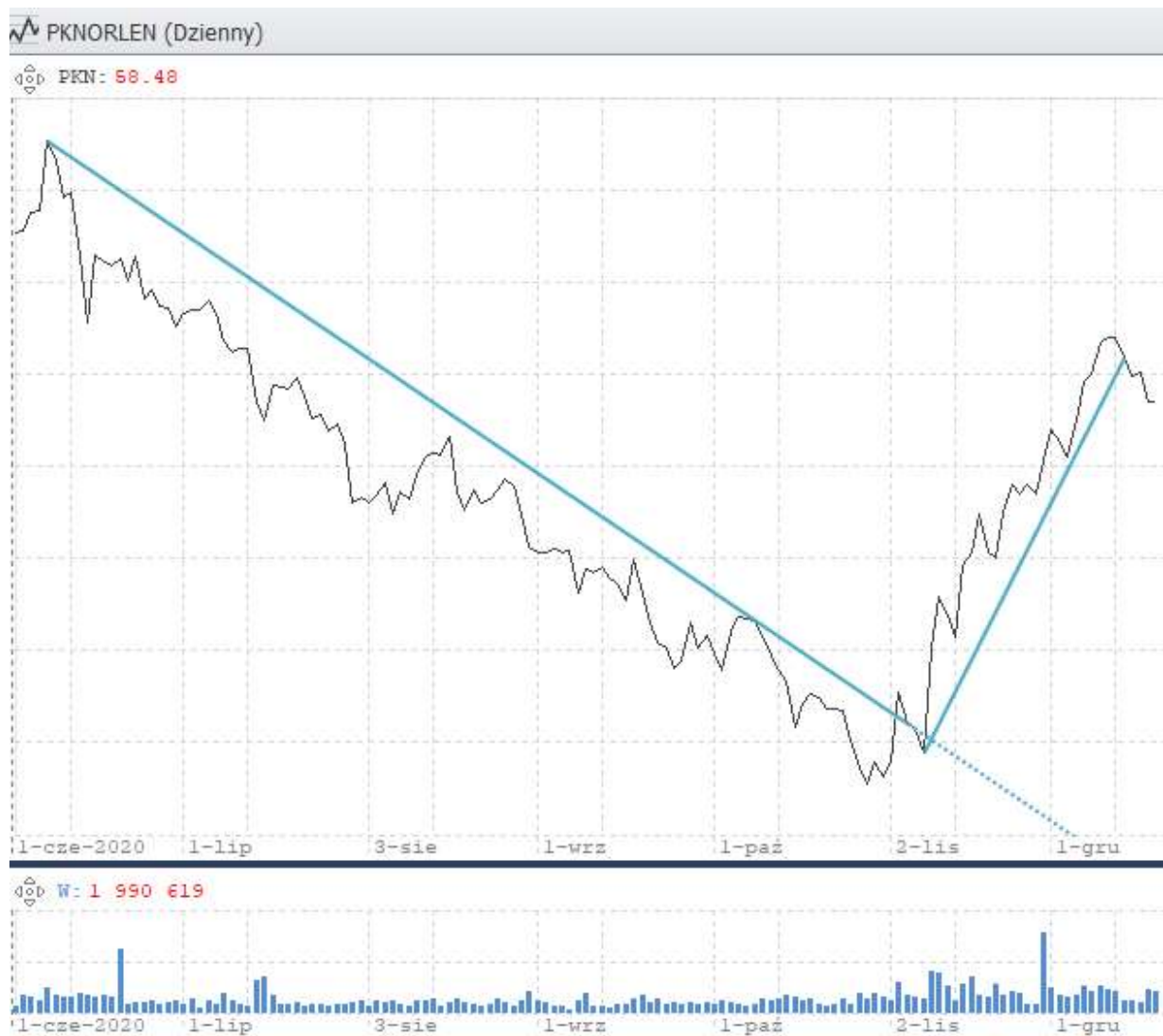
<sup>28</sup> Ibidem.

ogólny kierunek zmiany wartości waloru w danym okresie. Wyróżniamy trend wzrostowy (wierzchołki linii ceny z czasem przyjmują coraz wyższe wartości), spadkowy (dołki linii ceny z czasem przyjmują coraz niższe wartości) oraz trend boczny, w którym występują wahania cen w obie strony, bez długoterminowego zwiększania/zmniejszania ceny waloru. W trendzie bocznym wyznacza się jego kanał, czyli dwie linie, wskazujące pomiędzy jakimi wartościami porusza się cena akcji. Potwierdzeniem trendu jest najczęściej wolumen obrotu – obrót powinien wzrastać wraz ze wzrostem wartości kursu, a spadać wraz z ich spadkiem. Można wysunąć wniosek, że im większy obrót, tym większa pewność co do prawdziwości ruchu cen. Linię trendu wzrostowego rysuje się poniżej linii wyznaczającej wartość danego waloru, w ten sposób służy ona za linię sygnałną (jeśli kurs spadnie poniżej tej linii przy trendzie wzrostowym, można się spodziewać odwrócenia trendu i należy sprzedać walor). Zazwyczaj wyznacza się granicę bezpieczeństwa, która będzie odporna na chwilowe wahania niewpływające na trend i zakłada, że walor należy sprzedać tylko jeśli jego wartość spadnie o więcej niż kilka procent względem linii trendu. Trend bada się w dla różnych okresów czasu, w zależności od planowanego czasu trwania inwestycji. Dany walor może wzrastać w długim okresie czasu (jeśli prześledzimy 10 lat jego historii), więc można spodziewać się, że przyniesie on inwestorowi zysk, jeśli ulokuje w niego kapitał na wiele lat, a w krótkim okresie czasu (na przykład na przestrzeni pół roku) może być w trendzie spadkowym, co sugerowałoby, że inwestycja w ten walor nie jest dobrym wyborem dla inwestora, chcącego trzymać go w swoim portfelu maksymalnie kilka miesięcy.<sup>29</sup>

---

<sup>29</sup> Murphy, Op. cit., s. 45 – 72.

Rys. 2. Linie trendu dla ceny akcji PKN ORLEN SA w okresie 01.06.2020 – 20.12.2020.



Źródło: Opracowanie własne w programie Statica.

Na rysunku 2. zaznaczone są dwie linie trendu – pierwsza (patrząc od lewej) jest spadkowa, a druga jest rosnąca.

Kolejnymi liniami branymi pod uwagę w analizie technicznej są linia wsparcia i oporu. Należy na nie zwracać uwagę w okresach trendów bocznych, czyli dłuższych okresach czasu, w których wartość waloru waha się jedynie w określonym kanale (raz wzrastając, raz spadając) bez znaczącej długookresowej zmiany procentowej. Linia oporu to górna linia, do której wartość waloru okresowo dociera, ale jej nie przebija, a linia wsparcia to dolna linia wartości waloru, poniżej której jego wartość nie spada przez dłuższy okres czasu. Identyfikowanie trendów

bocznych i ograniczających je linii wsparcia i oporu jest ważne z punktu widzenia inwestora długoterminowego, dla którego liczy się realny wzrost w dłuższym okresie czasu.<sup>30</sup>

Rys. 3. Linie oporu i wsparcia dla ceny akcji banku PEKAO SA w okresie 01.03.2020 – 01.07.2020.



Źródło: Opracowanie własne w programie Statistica 4.

Na rysunku 3. przedstawiony został kanał trendu bocznego, w obrębie którego nie zachodzą znaczne zmiany wzrostowe czy spadkowe. Kurs banku PEKAO nie obierał w nich żadnego znaczącego kierunku wzrostu czy spadku, co może świadczyć o stagnacji w działaniach banku. Okres gwałtownego spadku wartości waloru (poprzedzający wejście w trend boczny) przypada na okres paniki na giełdach wywołanej pandemią Covid-19. W przeciwieństwie do wielu innych

<sup>30</sup> Tarczyński, Op. cit., s. 66 – 72.

walorów, banki odrabiały swoją przedpandemiczną wartość znacznie wolniej (lub w ogóle), między innymi ze względu na obniżenie stóp procentowych niemalże do zera, od których bardzo zależna jest rentowność banków.

W praktyce wykres waloru rysuje często specyficzne formacje (kształty) – ich rozpoznanie jest kluczowym zadaniem analizy technicznej i pozwala na ocenę odpowiedniego momentu zakupu (gdy cena waloru jest niska, ale wkrótce jego wartość zacznie rosnąć) i jego sprzedaży (gdy cena rynkowa waloru osiągnęła najwyższy pułap i wkrótce zacznie spadać). Formacje występują cyklicznie i mimo tego, że bywają mylące, to dobrze rozpoznane są pomocnym narzędziem inwestora. Rozpoznaje się dwa główne rodzaje formacji – zapowiadające zmianę trendu oraz zapowiadające kontynuację trendu. Z punktu widzenia inwestora pierwszy typ jest istotniejszy, ponieważ oznacza konieczność rzeczywistego działania i przeprowadzania transakcji. Kontynuacja trendu daje inwestorowi dodatkowe potwierdzenie, że optymalnie wykorzystuje swoje środki.<sup>31</sup> Z punktu widzenia zaawansowanej sieci neuronowej formacje analizy technicznej nie niosą aż tak istotnej informacji – odpowiednio głęboka sieć powinna sama rozpoznawać (mając daną wartość waloru w czasie) autoregresyjne zależności, które dla człowieka widoczne są jako specyficzne formacje.

Do najbardziej wiarygodnych formacji analizy technicznej zalicza się między innymi układ głowy z ramionami. Lewe ramie to dynamiczny wzrost wartości akcji z bardzo dużym wolumenem obrotu, który prowadzi do realizacji zysków i korekty (ze spadkiem obrotów). Głowa to maksymalny punkt formacji – po osiągnięciu minimum w lewym ramieniu, następuje wzrost cen przy niedużych obrotach, maksimum jest związane ze wzrostem obrotów, łączny poziom wartości waloru zawsze przekracza górny poziom lewego ramienia. W kolejnym etapie cena spada przy malejących obrotach do poziomu startowego tej formacji. Prawe ramię polega na wzroście cen przy niskich obrotach poziomu niższego niż poziom głowy. W tym momencie następuje spadek wartości waloru, często wiążący się ze znacznym wzrostem obrotów. Najwyższą wartość walor przyjmuje w momencie „głowy”, ale formacja potwierdza się dopiero, gdy po lokalnym maksimum z prawego ramienia wartość waloru zaczyna spadać – generując sygnał sprzedaży waloru. Formacja ma również swój odwrócony wariant, gdzie ramiona i głowa skierowane są w dół. Taka

---

<sup>31</sup> Op. cit., s. 137 - 161



odwrócona formacja oznacza zmianę trendu ze spadkowego na wzrostowy.<sup>32</sup> Na rysunku 4. przedstawiona została formacja głowy z ramionami dla kursu indeksu WIG20, kończąca krótki okres wzrostu wartości waloru. Po wyjściu z tej formacji nastąpił spadek wartości waloru.

Rys. 4. Wykres kursu indeksu WIG20 w okresie 01.02.2018 – 01.08.2019 z zaznaczoną formacją głowy z ramionami.



Źródło: Opracowanie własne w programie Statistica 4.

Innymi, często rozpoznawanymi formacjami są litera V i odwrócona litera V, występujący w nich spadek i wzrost następują pod tym samym kątem. Są ryzykowne w interpretacji, ponieważ często są składowymi innych, bardziej skomplikowanych formacji. Do takich należą między

<sup>32</sup> Ibidem.

innymi formacje podwójnego szczytu i podwójnego dna (w praktyce tworzące litery M i W). Podobnie jak w głowie z ramionami, potwierdzenia formacji szuka się w informacji o wolumenie obrotów. W formacji M, gdy kurs po osiągnięciu wierzchołka drugiego ramienia litery M przebija linię wsparcia, akcja otrzymuje sygnał spadku. Korzystając z tej formacji inwestorzy również zwracają uwagę na wolumen obrotu. Przy drugim wierzchołku wolumen powinien być niższy niż przy formowaniu się pierwszego. Formacja M jest więc sygnałem, że trend kursu akcji z rosnącego zmieni się na malejący, a formacja W kończy trend spadkowy i rozpoczyna wzrostowy. Formacje podwójnego szczytu i podwójnego dna kształtują się w niedługim czasie i nie prowadzą do dużych ruchów cenowych w dłuższym okresie.<sup>33</sup> Na rysunku 5. można zaobserwować podwójny szczyt z malejącymi obrotami podczas formowania się drugiego ramienia.

Rys. 5. Formacja podwójnego szczytu dla waloru ORANGEPL w okresie 01.07.2020 – 05.08.2020.



Źródło: Opracowanie własne w programie Statistica 4.

Do innych wartych uwagi (często występujących) formacji można zaliczyć także: formacje spodka i odwróconego spodka (tymczasowe trendy boczne zawierające się w pół elipsoidalnym kształcie), formacje trójkątów (wzwyżkujące, zniżkujące, symetryczne i odwrócone), a także

<sup>33</sup> Ibidem.

formacje zapowiadające kontynuację trendu, czyli: formacja klina, formacja flagi, kwadratu, prostokąta i diamentu.<sup>34</sup>

#### 1.1.4. Wskaźniki analizy technicznej

Poza śledzeniem formacji na wykresie ceny, istotnym aspektem analizy technicznej są wskaźniki, czyli operacje matematyczne przeprowadzane na cenie waloru z jakiegoś okresu, dzięki którym można wysnuć wnioski co do przyszłego kierunku zmiany jego wartości. Do najpopularniejszych rodzajów wskaźników zalicza się średnie kroczące z dwóch (lub więcej) różnych okresów, oscylatory pomagające oszacować stopień wykupienia danych akcji oraz wskaźniki pozwalające określić tempo zmian cen. Poniżej opisane zostaną najpopularniejsze ze wskaźników wśród wskazanych rodzajów, które później zostaną obliczone dla prognozowanego waloru i wykorzystane w uczeniu sieci neuronowej.

Jednym z narzędzi analizy technicznej jest generowanie sygnałów kupna i sprzedaży w oparciu o śledzenie krótko i długoterminowych średnich ruchomych i momentów ich przecinania się (które są sygnałami zakupu lub sprzedaży). Zaletą tej metody jest jej obniżona wrażliwość na okresowe i przypadkowe wahania, które wyłuszczone są dzięki średniej wartości kursu z kilkunastu-kilkudziesięciu sesji. Taka średnia obrazuje związek między siłą popytu i podaży na dany walor – dzięki niej można przewidywać trend na najbliższe sesje.<sup>35</sup> Z powodu możliwości aplikacji w krótkim okresie czasu (najbliższe sesje), sygnały z średnich kroczących mogą okazać się dobrym wsparciem sieci neuronowej przewidującej wartość waloru na krótkim horyzoncie czasowym.

W zależności od giełdy, waloru i obecnej sytuacji rynkowej stosuje się ruchome średnie o różnym okresie obliczania. Pring (2014), powołując się na przeprowadzone badania, proponuje wykorzystanie średnich kroczących z 8 i 17 okresów z linią sygnałną z 9 okresów lub wykorzystanie średnich kroczących z 12 i 25 okresów z linią sygnałną z 9 okresów. Miejsca, w których przecinają się te średnie są sygnałami zmiany trendu. Sygnał sprzedaży otrzymuje się, gdy średnia o krótszym kroku przebije się od góry przez średnią o dłuższym kroku, a sygnał kupna, gdy linia średniej o krótszym kroku przetnie od dołu linię średniej o dłuższym kroku. Bazowy wzór na średnią ruchomą wygląda następująco:

---

<sup>34</sup> Ibidem.

<sup>35</sup> Pring, M. J., 2014, *Technical Analysis Explained* (5th Edition), McGraw Hill, Singapur, s. 209 – 232 i 296 - 300

$$\overline{x_n} = \frac{\sum_{i=n-k+1}^n x_i}{k}, n \geq k \quad (1.5)$$

gdzie  $\overline{x_n}$  to kolejna wartość średniej arytmetycznej, zmieniająca się od k do liczby notowań, k to okienko czasowe średniej ruchomej, a  $x_i$  to kurs akcji na i-tej sesji.<sup>36</sup> W praktyce często stosuje się średnie kroczące ważone lub EMA (*exponential moving average*). Ponadto, informacje które można wyczytać z wykresów średnich ruchomych dobrze zawarte są również we wskaźniku MACD (który, jako wartość liczbowa, lepiej nadaje się jako dana wejściowa do modelu sieci neuronowej), opisanym w dalszej części tego rozdziału.

Kolejnymi praktycznymi narzędziami (w formie ilościowej), które wchodzą w skład analizy technicznej są wskaźniki. Chcąc potwierdzić słuszność sygnału kupna wygenerowanego przez formacje kształtujące się na wykresie, inwestor sięga po różnego rodzaju miary odzwierciedlające giełdową sytuację. Istnieje wiele takich miar i wskaźników, a różni uczestnicy posługują się różnymi ich pulami. Racjonalnie działający inwestor będzie podejmował decyzje poparte większością wskaźników (w obrębie tych, które stosuje), przy czym zaleca się stosowanie co najmniej kilku, żeby uodpornić się na błędy mniejszej puli.<sup>37</sup> Poniżej opisano kilka popularnie stosowanych wskaźników, które wykorzystane zostaną również w części badawczej pracy.

Wskaźnik ROC (*rate of change*) to narzędzie do oceny tempa ewolucji akcji kursu. Wśród jego praktycznych zalet należy wymienić łatwość obliczania i interpretacji. Formuła na ROC wygląda następująco:

$$ROC_n = \left( \frac{x_n}{x_{n-k}} - 1 \right) * 100\% \quad (1.6)$$

gdzie  $ROC_n$  to wskaźnik szybkości zmiany w okresie n,  $x_n$  to cena akcji na n-tej sesji, a  $x_{n-k}$  to cena akcji na sesji o k wcześniejszej niż n. Otrzymujemy w ten sposób krzywą oscylującą wokół zera. W praktyce najczęściej stosuje się 5 lub 10 okresowe opóźnienie. Przy k = 5 szybkość sygnałów jest większa, ale są one bardziej podatne na krótkookresowe wahania. Głównymi zastosowaniami wskaźnika ROC są: wyznaczenie poziomu wykupienia i wysprzedaży oraz badanie dynamiki wzrostu/spadku kursów. Jeżeli w jakimś okresie cena akcji i wartość wskaźnika ROC rosną, to dochodzi do sytuacji, gdy rynek ulega „wyczerpaniu” – kupcy generujący popyt

---

<sup>36</sup> Ibidem.

<sup>37</sup> Op. cit., s. 9.

zaspokoili swoje potrzeby. W konsekwencji dany walor zostaje osłabiony. W tej sytuacji zazwyczaj trend wskaźnika ulega zmianie. Taką sytuację nazywa się wykupieniem rynku. W sytuacji odwrotnej (cena waloru i ROC spadają) w końcu rynek się wysprzeda i następuje zmiana tego trendu. Wykorzystanie ROC do obserwacji dynamiki wskaźnika to kolejny sposób wykorzystania go w praktyce. Jeśli dynamika wzrostu kursów staje się mniejsza, ROC to sygnalizuje, bo procentowa zmiana kursu w danym dniu jest mniejsza niż sprzed k notowań. W przypadku gdy przed zmianą trendu następuje słabnąca dynamika wzrostu, można wykorzystać ROC jako zapowiedź nadchodzącej zmiany. Im wyższa jest wartość wskaźnika (ponad zerem), tym bardziej cena bieżąca różni się od ceny sprzed k notowań i świadczy o rosnącym rynku. O silnym, rosnącym rynku świadczy stale rosnąca wartość wskaźnika ROC. Sygnałem zakupu jest moment, gdy ROC jest ujemne, ale rośnie w kierunku zera, a jeśli dodatkowo minimum wartości waloru przypadło na okres, gdy ROC już wzrastało, jest to bardzo silny sygnał kupna.<sup>38</sup>

Wskaźnik MACD (*Moving Average Convergence Divergence*) jest liczbowym wyznacznikiem tego, co można uzyskać przy pomocy obserwacji średnich ruchomych dla wartości walorów. Jest to grupa wskaźników, które za podstawę biorą sobie średnie krótko i długo terminowe. Wzór na MACD prezentuje się następująco:

$$MACD = \bar{x}_{k1} - \bar{x}_{k2} \text{ lub } MACD = \frac{\bar{x}_{k1}}{\bar{x}_{k2}} \quad (1.7)$$

gdzie  $k_1$  i  $k_2$  to krok uśrednienia (drugi jest zawsze większy od pierwszego), a  $\bar{x}_{k1}$  i  $\bar{x}_{k2}$  to średnie wykładnicze okresowe, wyznaczone za pomocą wzoru:

$$\bar{x}_{ne} = \bar{x}_n * \left(1 - \frac{2}{k}\right) + \left(\frac{2}{k}\right) * x_n \quad (1.8)$$

gdzie  $\bar{x}_{ne}$  to wielodniowa średnia wykładnicza,  $\bar{x}_n$  wielodniowa średnia arytmetyczna (lub ważona),  $x_n$  to cena akcji na n-tej sesji, k to krok uśredniania, a n to kolejna wartość średniej. Obliczenia aktualizuje się po każdej sesji w sposób kroczący. Najczęściej stosuje się okres krótszy (z 8 lub 12 sesji) i dłuższy (z 12 lub 25 sesji), do tego wyznacza się linię sygnałną (średnią wykładniczą z, na przykład, 9 okresów dla wartości MACD). Jest to wskaźnik szczególnie istotny dla inwestorów krótkoterminowych, ponieważ małe wartości k odzwierciedlają także przypadkowe zmiany kursów, które nie wpływają na długoterminowe zmiany trendów. Stosowanie MACD w

---

<sup>38</sup> Op. cit. s. 246 – 253.

praktyce polega na obserwowaniu wzajemnego położenia MACD i jej linii sygnałnej. Równowagą jest poziom zerowy, wskaźnik rosnący powyżej zera oznacza rynek wzrostowy, a spadający poniżej zera jest oznaką rynku malejącego. Sygnał kupna jest generowany w momencie przecięcia od dołu linii sygnałnej przez linię MACD. Sygnał sprzedaży jest generowany, gdy linia dodatnia MACD przebiega z góry linię sygnału.<sup>39</sup>

Innym wskaźnikiem informującym o wykupieniu rynku jest wskaźnik relatywnej siły RSI (*Relative Strength Index*). RSI czasem nazywany jest wskaźnikiem siły wewnętrznej akcji. Pokazuje on stan rynku z perspektywy sił popytu i podaży w skali unormowanej (z przedziału 0 do 100). Linie sygnałne zaznacza się dla poziomów 70 (wykupienia) i 30 (wyprzedania). Wskaźnik najczęściej wyznacza się w okresach 9 lub 14 dniowych. Główne założenie jest takie, że silny popyt jest korzystny dla wzrostu cen, a silna podaż sprzyja ich spadkowi, dlatego też przeprowadza się oddzielną analizę dla wzrostów i oddzielną dla spadków cen na kolejnych sesjach. Wskaźnik wyznacza się za pomocą wzoru:

$$RSI = 100 - \left( \frac{100}{1 + RS} \right),$$

$$RS = \frac{\text{średni wzrost kursu z } k \text{ dni}}{\text{średni spadek kursu z } k \text{ dni}} \quad (1.9)$$

Dla danego okresu  $k$  dzieli się zmiany ceny na dodatnie i ujemne i dla obu tych grup wyznacza średnią arytmetyczną, dzięki czemu można policzyć  $RS$  i unormować wynik za pomocą wzoru na  $RSI$ . Przy bardzo dużej dynamice rynku miara traci na wartości i rzadko jest stosowana. Najczęściej przekroczenie przez  $RSI$  linii sygnałnych (na poziomach 70 oraz 30) jest jedynie wskazówką dla inwestorów. Mocny, długotrwały trend w jedną ze stron zawsze może wytworzyć ekstremalny odczyt wskaźnika, ale bazowanie na takich sygnałach mogłoby być przedwczesne.<sup>40</sup>

Do innej grupy wskaźnik należy  $CCI$  (*Commodity Channel Index*), który służy do obliczania impetu cenowego (wskaźnik momentum do odchylenia arytmetycznego dla ceny akcji). Wyznacza się go za pomocą wzoru:

$$CCI_n = \frac{x_n - \bar{x}_k}{0,015 - D_n}$$

<sup>39</sup> Tarczyński, Op. cit., s. 99 – 101.

<sup>40</sup> Murphy, Op. cit., s. 239 – 246.

$$D_n = \frac{\sum_{i=1}^n |x_i - \bar{x}_k|}{n} \quad (1.10)$$

gdzie  $x_n$  to typowa cena akcji dla sesji  $n$  wyznaczona jako średnia arytmetyczna z ceny maksymalnej, minimalnej i zamknięcia dla danego okresu (przy założeniu okresu równego 5 ceną zamknięcia jest cena z piątego okresu), a  $\bar{x}$  to średnia arytmetyczna ruchoma wyznaczona dla wybranego okienka  $k$  z ceny typowej.<sup>41</sup>

Jeśli wartość wskaźnika CCI rośnie, to cena akcji jest wysoka w porównaniu z średnią ceną akcji. Jeśli CCI maleje, to cena akcji jest zdecydowanie niższa od średniej. Stosując wskaźnik CCI należy szukać rozbieżności wartości wskaźnika z ceną akcji. CCI informuje o poziomie wykupienia (poziom 100) oraz wysprzedań (poziom -100).<sup>42</sup>

Większość oscylatorów porównuje cenę akcji z ceną wcześniejszą o  $k$  okresów, w związku z tym wartości takich oscylatorów są silnie powiązane z wybranym okresem. Z myślą o uodpornieniu oscylatorów na wahania związane z okresem powstał oscylator *ultimate*, który łączy trzy inne oscylatory wykorzystując ich ważone sumy. Wartości oscylatora aktualizuje się po każdym notowaniu, proces obliczania go jest następujący:

- wyznaczenie *mocy kupowania*  $B_n$  odejmując od bieżącej ceny akcji cenę minimalną z ostatniego tygodnia,
- wyznaczenie dzisiejszej *prawdziwej wartości*  $R_n$  przyjmując za nią maksymalną cenę z ostatniego tygodnia,
- wyznaczenie trzech sum  $B_n(SB(7), SB(14), SB(28))$  dla trzech różnych okresów (7, 14 i 28 dni),
- wyznaczenie analogicznych sum dla  $R_n$  (SR(7), SR(14), (SR(28)),
- podzielenie sum  $B_n$  i  $R_n$  mnożąc je przez odpowiednie wagi:  

$$\frac{SB(7)}{SR(7)} * 4, \frac{SB(14)}{SR(14)} * 2, \frac{SB(28)}{SR(28)}$$
- Obliczenie OU przez dodanie powyższych ilorazów:

$$OU = \frac{SB(7)}{SR(7)} * 4 + \frac{SB(14)}{SR(14)} * 2 + \frac{SB(28)}{SR(28)} \quad (1.11)$$

<sup>41</sup> Tarczyński, Op. cit., s. 121 – 123.

<sup>42</sup> Ibidem.

Sygnały kupna generowane są w momencie dywergencji malejącej ceny z oscylatorem oraz gdy linia oscylatora przebiega od dołu linii wysprzedania. Sygnał sprzedaży jest generowany, gdy oscylator przekracza granicę 50, a później spada poniżej 45 lub gdy oscylator przekracza poziom 70.<sup>43</sup>

## 1.2. Determinanty cen akcji

### 1.2.1. Makroekonomiczne

Giełdy papierów wartościowych, jako pewna część światowej gospodarki, funkcjonują w otoczeniu rynkowym składającym się z wielu elementów. Dlatego też kursy akcji na giełdach wzajemnie oddziałują ze sobą z innymi rodzajami przepływów finansowych oraz szeroko pojętą sytuacją makroekonomiczną danego obszaru. Budując portfel inwestycyjny, inwestorzy zwracają uwagę na ostatnie zmiany czynników makroekonomicznych oraz ich prognozowane zmiany w przyszłości. Wiele dzieł naukowych podejmuje problem wpływu tych czynników na kursy giełdowe. Zrozumienie, jakie czynniki wpływają na kurs akcji będzie kluczowe przy zbieraniu danych, które zostaną wykorzystane w modelu sieci neuronowej w części badawczej tej pracy.

Flannery i in. (2002) uważają, że papiery wartościowe, na których kursy wpływają niedywersyfikowalne czynniki ryzyka, powinny dostawać z tytułu tego możliwego ryzyka premię (w naszej stroniącej od ryzyka ekonomii). Zmienne makroekonomiczne są przykładem pozarynkowych czynników ryzyka, ponieważ bezpośrednio wpływają na przepływy pieniężne wielu firm oraz mogą wpływać na uzależnioną od ryzyka stopę dyskontową.<sup>44</sup>

Flannery i in. sprawdzili wpływ trzech zmiennych nominalnych – CPI (Wskaźnik cen towarów i usług konsumpcyjnych), PPI (Wskaźnik cen dóbr produkcyjnych) i Agregat pieniężny (M1 lub M2) oraz trzech zmiennych realnych – stopę bezrobocia, bilans handlowy i liczbę rozpoczętych budów domów (*housing starts*). Co istotne, tylko podaż pieniądza wpływa zarówno na poziom jak i zmienność zwrotów z aktywów. Pozostałe dwie zmienne nominalne oddziałują na poziom zwrotu, a trzy wymienione wyżej makroekonomiczne zmienne realne na zmienność. Flannery i in. twierdzą, że po serii badań na danych giełdowych z lat 1980-1996 przy użyciu

---

<sup>43</sup> Op. cit., s. 124 – 125.

<sup>44</sup> Flannery, M. J., Protopapadakis, A. A., Macroeconomic Factors Do Influence Aggregate Stock Return. *The Review of Financial Studies*, Volume 15, Issue 3, April 2002, s. 751-753.



modelu GARCH udało im się udowodnić, że wszystkie wymienione powyżej zmienne istotnie oddziałują na rentowność inwestowania w rynek kapitałowy.<sup>45</sup>

Wszystkie trzy współczynniki CPI, PPI jak i M1 są ujemnie skorelowane ze zwrotem z akcji – jeśli inflacja lub podaż pieniądza okaże się wyższa niż oczekiwano, wysokość zwrotów z inwestycji na rynku kapitałowym spada o statystycznie istotną wielkość. Takiej korelacji ze stopą zwrotów nie wykazują zmienne realne.<sup>46</sup>

Również Geske i in. (1983) zwracają uwagę na to, że szoki inflacyjne (które mogą być mierzone niespodziewanymi zmianami CPI) wpływają na zagregowane zwroty z inwestycji na giełdach papierów wartościowych. Dzieje się tak dlatego, że nagła zmiana inflacji spowodowana jest zmienionym tempem podaży pieniądza. Racjonalni inwestorzy zdają sobie sprawę z tego, że taka zmiana pociągnie za sobą serię odpowiedzi w postaci zmian w polityce fiskalnej i monetarnej – losowy negatywny szok wpływający na zwroty inwestycji giełdowych doprowadzi do wyższego bezrobocia i niższych przychodów w przedsiębiorstwach. To z kolei prowadzi do obniżenia wpływów do budżetu państwa z podatków CIT. Wydatki rządowe nie uwzględniają tej zmiany, więc powiększy się deficyt budżetowy, który Bank Centralny będzie musiał uzupełnić pożyczając pieniądze z sektora publicznego.<sup>47</sup>

Zmienne realne: BOT (bilans handlowy), EMP (stopa bezrobocia) oraz RGNP (realny produkt krajowy netto) wpływają na zmienność zwrotów na poziomie istotności 1%. Spośród wymienionych EMP najsilniej oddziałuje na zmienność, ogłoszenie tego czynnika podnosi warunkową zmienność o 68,5%. Natomiast ogłoszenie BOT podnosi zmienność o 15%. Liczba rozpoczęć budów domów wpływa na zmienność na poziomie istotności 10% i również podnosi ją o więcej niż 15%. Na uwagę zasługują również zmienne oznaczające podaż pieniądza M1 i M2. Wpływają one na zmienność zwrotów na poziomach istotności kolejno: 5% i 10%. Tylko M1 wpływa więc zarówno na stopę zwrotów, jak i na zmienność zwrotów.<sup>48</sup>

Flannery i in. (2002) zwracają uwagę na to, że jeśli czynnik istotnie wpływa na rynek, jego zmiana powinna wiązać się ze zwiększonym wolumenem obrotu. Ekonometrycznie zbadali

---

<sup>45</sup> Ibidem.

<sup>46</sup> Op. cit., s. 765.

<sup>47</sup> Geske, R., Roll, R., 1983, "The Fiscal and Monetary Linkage Between Stock Returns and Inflation", The Journal of Finance, Vol. 38, No. 1 (Mar., 1983), s. 28-29.

<sup>48</sup> Flannery i in., Op. cit, s. 765.

zmienność wolumenu obrotu w zależności od tego, czy danego dnia były ogłoszone jakieś czynniki makroekonomiczne, czy nie. Z ich analizy wynika, że ogłoszenia prognoz dla CPI, PPI, M1, M2, rozpoczęcie budów domów oraz stopy bezrobocia istotnie wpływają na wolumen obrotu, co potwierdza wyniki ich badań, wskazujące na istotność tych zmiennych co do poziomu zwrotu i zmienności zwrotów z rynku kapitałowego. RGNP, mimo wpływu na zmienność zwrotów, nie zmienia istotnie wolumenu obrotu.<sup>49</sup>

Inwestorzy oczekują dodatnich zwrotów z inwestycji za narażanie się na dwa typy ryzyka – niepewność zwrotu z inwestycji w okresie bieżącym oraz możliwe zmiany które nastąpią w przyszłych możliwościach inwestycyjnych. Z tego wynika, że jakiegokolwiek zmienne ekonomiczne, które są skorelowane z marginalną użytecznością konsumpcji, mogą być istotnymi elementami równania dążącego do równowagi. Sensowne jest więc założenie, że czynniki makroekonomiczne istotnie przyczyniają się do oceny szans inwestycyjnych. Na przykład, zmiana w stopie bezrobocia to informacja dotycząca tego, jaki będzie zwrot do kapitału ludzkiego, a z kolei zmiana w bilansie handlowym wskazuje na to, że zmieniać będzie się również kurs walutowy.<sup>50</sup>

Dane makroekonomiczne ogłaszane są regularnie przez bank centralny i/lub inne instytucje państwowe. Natomiast należy się również zastanowić, czy lepiej uwzględniać realne dane, czy też prognozy tego co zostanie ogłoszone. Prognozy najczęściej zbierane i ogłoszone są na około tydzień przed ogłoszeniem przez różne związane z ekonomią instytucje i być może to one lepiej pokazują obecny sentyment rynkowy.<sup>51</sup>

Badania empiryczne wykazują, że realny wpływ na zagregowane wartości akcji na giełdzie mają również zmiany krótkoterminowych stóp procentowych. Wartość tego czynnika jest ujemnie skorelowana ze zwrotami z akcji na giełdzie.<sup>52</sup> Wyższa stopa procentowa oznacza wyższą stopę wolną od ryzyka (dla inwestycji w obligacje państwowe), a więc inwestorzy tracą część zachęty do ponoszenia ryzyka na giełdzie.

Lamont (2001) podjął się podobnych badań, przyjął on jednak inną metodologię, niż opisano we wcześniejszych dziełach – wykorzystał technikę „*economic tracking portfolio*”, czyli

---

<sup>49</sup> Op. cit., s. 771-773.

<sup>50</sup> Op. cit., s. 755.

<sup>51</sup> Op. cit., s. 760.

<sup>52</sup> Geske i in., Op. cit., s. 28-29.

portfolio aktywów, którego zwroty mierzą jakąś ekonomiczną zmienną. Miesięczne zwroty z inwestycji giełdowych są użyteczne do prognozowania konsumpcji, przychodu pracy, inflacji, zwrotów z inwestycji czy obligacji. Zależności między tymi prognozami definiują portfolia, które mierzą oczekiwania rynku na temat przyszłych wartości zmiennych ekonomicznych. Używanie ich jako narzędzi do śledzenia przyszłych wartości zmiennych ekonomicznych znacznie lepiej obrazuje wrażliwość cen aktywów na informacje o tych zmiennych. „*Tracking portfolios*” z jednej strony obrazują zwroty z aktywów, a z drugiej strony są to zwroty z interpretowalną wartością ekonomiczną. Ich konstruowanie polega na używaniu danych o obecnych zwrotach z aktywów jako instrumentów do przewidywania zmian danych zmiennych w przyszłości. Dzięki takiemu podejściu „*tracking portfolios*” pozwalają na oszacowanie premii za ryzyko.<sup>53</sup>

Po przeprowadzeniu serii testów regresji przy użyciu metody najmniejszych kwadratów, Lamont konkluduje, że czynnikami makroekonomicznymi najbardziej wpływającymi na zmiany cen aktywów na giełdzie papierów wartościowych są: produkcja przemysłowa, konsumpcja oraz wynagrodzenie pracy. Spośród wymienionych produkcja przemysłowa najistotniej wpływa na zwroty z inwestycji – według analizy regresji, współczynnik beta dla tej zmiennej wynosi 0,1, a więc prognozowane zwiększenie produkcji przemysłowej o 1%, powinno doprowadzić do zwiększenia się zagregowanej wartości cen akcji o 0,1%.<sup>54</sup>

McQueen i in. (1993) rozważają, dlaczego tak wiele prac naukowych badających wpływ czynników makroekonomicznych na zwroty z inwestycji giełdowych, nie może jednoznacznie udowodnić, że istnieje pomiędzy nimi zależność. Wysznuwają tezę, że badacze zbyt często zakładają, że inwestorzy reagują na dany czynnik identycznie, niezależnie od sytuacji rynkowej (od tego w jakim momencie cyklu znajduje się gospodarka).<sup>55</sup>

Pozytywna, niespodziewana zmiana w produkcji przemysłowej w czasach kryzysu, w okresie szczytowej stopy bezrobocia i nadmiaru czynników produkcji w przemyśle, może oznaczać zarówno koniec kryzysu, jak i prognozę znacznie wyższych przepływów finansowych w firmach. Ogłoszenie tego czynnika to dobra wiadomość dla giełdy papierów wartościowych. Natomiast takie samo ogłoszenie, gdy stopa bezrobocia jest na niskim poziomie, a fabryki produkują z niemal

---

<sup>53</sup> Lamont, O., 2001, „Economic Tracking Portfolios”, *Journal of Econometrics*, 105, s. 161-163.

<sup>54</sup> Op. cit., s. 179-180.

<sup>55</sup> McQueen, G., Roley V., 1993, „Stock Prices, News, and Business Conditions,” *Review of Financial Studies*, 6, s. 683-684.

pełnym potencjałem, może prowadzić do obaw przed przegrzaniem się gospodarki, zwiększenia inflacji oraz możliwych działań ustawodawców prowadzących do zwiększenia stopy procentowej. To natomiast jest niekorzystne dla gospodarki.<sup>56</sup>

Kolejnym podobnym przykładem był wpływ ogłoszenia zmian w stopie bezrobocia zależny od koniunktury. W 1983 roku, po 16 miesiącach recesji, stopa bezrobocia spadła do 10,4% (wynik o 0,2-0,3 punktów procentowych niższy niż oczekiwano), co spowodowało wzrost na indeksie Dow Jones Industrial Average o 13,25%. Natomiast gdy w 1988 roku, po kilku latach wzrostów gospodarczych, ogłoszono, że stopa bezrobocia spadła do 5,3% (o 0,1-0,2 punktów procentowych niżej niż oczekiwano), wartości indeksów na giełdzie spadły, bo inwestorzy zaczęli się obawiać zacieśnienia polityki finansowej.<sup>57</sup>

McQueen i in. identyfikują kilka czynników makroekonomicznych, których zmiana może wywierać realny wpływ na zwroty z inwestycji giełdowych:

- realna aktywność gospodarcza (której miarą może być RGNP), ponieważ jeśli gospodarka działa poniżej swojego potencjału, niespodziewany wzrost aktywności ekonomicznej może oznaczać zwiększenie przepływów pieniężnych bardziej niż zwiększenie stóp dyskontowych, co z kolei doprowadzi do wzrostu wartości akcji. Jeśli jednak gospodarka operuje powyżej swojego długoterminowego potencjału, taka sama niespodzianka może oznaczać duże podwyżki stopy dyskontowej i doprowadzić do spadku wartości akcji
- bilans handlowy – niespodziewane zwiększenie deficytu handlowego związane jest ze spadkiem zagranicznego popytu na dobra produkowane w danym kraju, co powoduje zmniejszenie cash flow w firmach w danym kraju. Ponadto, deficyty w bilansie handlowym mogą prowadzić do wprowadzania protekcyjnych barier handlowych oraz mogą zachęcać ustawodawców do obniżania wartości dolara (do czego mogą doprowadzić obniżając stopy procentowe).
- Szoki inflacyjne – ogłoszenie inflacji większej, niż oczekiwano, może prowadzić inwestorów do próby przewidzenia reakcji banku centralnego – wyższa inflacja może wiązać się z zacieśnieniem polityki finansowej, a to z kolei może spowodować mniejszy cash flow w spółkach akcyjnych.

---

<sup>56</sup> Op. cit., s. 685-686.

<sup>57</sup> Ibidem.

- Szoki w podaży pieniądza – Chęć Banku Centralnego Stanów Zjednoczonych do realizacji swoich celów oraz równoważenia niespodzianek w podaży pieniądza również zależeć może od stanu gospodarki. Pod koniec długiego okresu ekspansji gospodarczej, gdy obawy o przyszłą inflację wzrastają, niespodzianka w podaży pieniądza może mieć zupełnie inny wpływ na wartość spółek na giełdzie, niż gdy taka sama niespodzianka występuje w najgorszym momencie kryzysu. Dodatkowo, poza ogłoszeniami związanymi z podażą pieniądza, wiadomości o zmianach stopy dyskontowej przez Bank Centralny również mogą wpływać na ceny akcji. Takie zmiany mogą wiązać się z nowymi wytycznymi polityki pieniężnej, co z kolei może prowadzić do zmiany stopy dyskontowej kapitału własnego oraz przepływów pieniężnych.<sup>58</sup>

Dlatego też rozważając wpływ czynników makroekonomicznych na zwroty z inwestycji na giełdzie papierów wartościowych, powinno się brać pod uwagę na jakim etapie cyklu gospodarczego, zostały one ogłoszone, ponieważ ich interpretacji różni się w zależności od otoczenia rynkowego.<sup>59</sup> Niekiedy trudnościami pozostaje określenie, w którym momencie cyklu znajduje się gospodarka, ale praca McQueena i in. pokazują, że warto przyjmować w tej sprawie odpowiednie założenia.

Różne reakcje inwestorów giełdowych na podobne ogłoszenia spowodowane są zapewne przez oczekiwane przepływy finansowe (*cash flow*) firm. Nieoczekiwane wzrosty aktywności ekonomicznej w kryzysie zwiększały oczekiwania inwestorów co do przyszłej aktywności ekonomicznej i przepływów finansowych. Takie same informacje w momencie hossy nie prowadziły do oczekiwań wyższych przepływów finansowych.<sup>60</sup> I zapewne zwiększały obawy inwestorów o niemalże maksymalnym wykorzystaniu czynników produkcji oraz potencjalnie rosnącej inflacji (niosącej za sobą obostrzenia w polityce finansowej).

Czynnikiem równie ważnym co wysokość zwrotów z inwestycji w papiery wartościowe, jest ich zmienność w czasie. Zmienność jest niemal bezpośrednią miarą ryzyka inwestycyjnego – duża zmienność wiąże się z możliwościami dużych zwrotów oraz tak samo dużych strat. Jedną z miar zmienności jest odchylenie standardowe, które można mierzyć na przykład na miesięcznych

---

<sup>58</sup> Op. cit., s. 688-689.

<sup>59</sup> Op. cit., s. 692.

<sup>60</sup> Ibidem.

zmianach w zagregowanych zwrotach z inwestycji w papiery wartościowe. Zmienność jest więc czynnikiem bardzo często brany pod uwagę przez inwestorów i zarówno zmiana zmienności, jak i zmiana w czynnikach na nią wpływających, może wiązać się ze wzrostem lub spadkiem opłacalności inwestycji.

Schwert (1989) zaobserwował, że na giełdzie Amerykańskiej zmienność między 1857 a 1987 rokiem wahała się pomiędzy 2% a 20%. Taka rozpiętość sprawia, że szanse na to, że jest to tylko błąd oszacowania, są bardzo niewielkie. Wyniki potwierdzono sprawdzając zależność pomiędzy zmiennością kilku czynników makroekonomicznych, a zmiennością zwrotów z akcji giełdowych.<sup>61</sup>

Badając giełdę amerykańską w latach 1900-1952 Schwert wnioskuje, że zmienność inflacji, podaży pieniądza oraz produkcji przemysłowej znacząco wpływa na zmienność zwrotów z inwestycji giełdowych. Z modelu regresji który wynika, że suma współczynników przyporządkowanych do powyższych zmiennych wynosi 0,45 z błędem standardowym na poziomie 0,12 – jeśli więc zmienność inflacji, podaży pieniądza oraz produkcji przemysłowej wzrosłaby o 1%, zmienność zwrotów z inwestycji w papiery wartościowe wzrosłaby o 0,45%. Natomiast najbardziej wpływającym na zmienność wartości akcji we wszystkich okresach, okazała się podaż pieniądza. Schwert wnioskuje, że zmienność wartości papierów wartościowych największa jest w okresach recesji.<sup>62</sup>

#### 1.2.2. Z rynków kapitałowych

Poza czynnikami makroekonomicznymi, w wycenie kursu giełdowego oraz ocenie jego zmienności wartości analizuje się także dane z samej giełdy (lub raczej – giełd). Kursy powiązane są ze sobą na wiele sposobów – inwestor sprzedając akcje jednej spółki często decyduje się na zakup akcji innej. Indeksy giełdowe z kolei najczęściej składają się z konkretnej ilości największych spółek danej giełdy (na przykład NASDAQ100, WIG20) lub ze spółek działających w podobnym obszarze (na przykład WIG.GAMES). W związku z tym wartość indeksu, czy nawet akcji konkretnej spółki nie może być ustalona (a jej zmienność nie może być sprawdzana) w izolacji od pozostałych zmian dochodzących na danej giełdzie. Inwestor bacznie obserwuje więc szeroką gamę operacji, mających miejsce na giełdzie, na której znajduje się interesujący go walor.

---

<sup>61</sup> Schwert, G. W., 1989, "Why Does Stock Market Volatility Change Over Time?", *Journal of Finance*, 44, 1115-1116.

<sup>62</sup> Op. cit., s. 1143-1146.

Do tego, przy stosunkowo swobodnych przepływach kapitału pomiędzy krajami, należy również doszukiwać się zależności pomiędzy aktywami z różnych stron świata. Inwestor niechętny ryzykować pieniędzy na niepewnym rynku może przenieść swoje aktywa na inną giełdę, na której trwa w danym momencie hossa. Niedawno taka sytuacja mogła mieć miejsce na przykład na giełdzie londyńskiej – w związku z dość poważną pandemią Covid-19 w Zjednoczonym Królestwie indeks FTSE 100 stracił na wartości (20.02.2020: 7436,64 do 23.02.2020: 4993,89) i do końca lipca nie odzyskał nawet połowy z tego, co stracił (wartość na 31.07: 5897). Jednym z powodów takiego zjawiska może być dalsza niepewność brexitowa (przeciągające się negocjacje). Wiele innych giełd zachowała się inaczej i wyrównały, prawie wyrównały lub nawet przebiły wartości sprzed początku pandemii Covid-19 (na przykład NASDAQ czy DAX).

Jednym z czynników ściśle powiązanych z wartością danego waloru jest jego wolumen obrotu. Karpoff (1987) zwraca uwagę na wpływ wolumenu obrotu danego papieru wartościowego na jego cenę i wymienia cztery powody, dla których ta relacja jest istotna z punktu widzenia naukowego i z punktu widzenia inwestorów:

- Po pierwsze zależność między ceną papieru wartościowego a wolumenem jego obrotu dostarcza istotnych informacji o strukturze rynków finansowych. Modele omówione w tej pracy przewidują rozmaite zależności pomiędzy ceną a wolumenem obrotu, które zależne są od stopnia przepływu informacji, sposobu rozprowadzania informacji, stopnia do jakiego ceny rynkowe zawierają informacje, rozmiaru rynku oraz ograniczeń w krótkiej sprzedaży. Empiryczne zależności między ceną a wolumenem mogą pomóc zweryfikować hipotezy o strukturze rynku.
- Po drugie relacja między ceną a wolumenem jest ważna dla badania wydarzeń. Badania te korzystają z kombinacji zmian ceny i obrotu do wyciągania wniosków. Jeżeli zmianom w cenach towarzyszy duży obrót, silna relacja między ceną a obrotem uwiarygodni takie testy. Zmiany cen interpretowane są jako rynkowa ocena nowej informacji. Natomiast towarzyszący temu obrót może oznaczać stopień, w jakim inwestorzy nie zgadzają między sobą się co do znaczenia danej informacji.
- Po trzecie relacja między ceną a obrotem jest kluczowa w dyskusji o empirycznej dystrybucji spekulacyjnych cen. Jedną z teorii, którą potwierdzają testy wykorzystujące zależność między ceną a wolumenem jest, że dystrybucja stopni zwrotu wydaje się być

kurtotyczna, ponieważ dane zebrane są z mieszaniny dystrybucji o różnych warunkowych wariancjach. To z kolei ma kilka implikacji. Na przykład, wydaje się, że dane generowane są przez warunkowy, stochastyczny proces ze zmiennym parametrem wariancji, który można oszacować na podstawie wolumenu obrotu. W związku z tym, wiedza na temat relacji między ceną a wolumenem może być wykorzystana w badaniach nad wydarzeniami do mierzenia zmian w wariancji cen w okresach bez wydarzeń i w okresach z wydarzeniami.

- Po czwarte zależności między ceną a wolumenem obrotu ma istotne znaczenie dla badań nad rynkiem futures. Zmienność ceny wpływa na wolumen handlu kontraktami futures. To z kolei jest istotnym czynnikiem w dyskusji czy spekulacja jest stabilizującym czy destabilizującym czynnikiem dla cen kontraktów futures. Okres zapadalności kontraktu futures wpływa na wolumen obrotu i możliwe, że w związku z tym wpływa również na zmienność ceny.<sup>63</sup>

Karpoff dokonuje przeglądu badań o wpływie obrotu na cenę aktywów i wymienia, że do tej pory różni badacze dochodzili do różnych wniosków: a) nie ma zależności między zmianą ceny a wolumenem obrotu, b) korelacja pomiędzy wolumenem obrotu a bezwzględną wartością zmiany ceny jest dodatnia zarówno na rynku kapitałowym jak i na rynku kontraktów futures, c) korelacja pomiędzy wolumenem a zmianą ceny per se jest dodatnia na rynkach kapitałowych, d) wolumen jest wyższy w przypadku, gdy cena papieru wartościowego wzrasta niż gdy spada. Karpoff przeprowadził własne badanie tego zagadnienia i konkluduje, że: istnieje korelacja pomiędzy wartością bezwzględną zmiany ceny aktywów i ich obrotem, istnieje dodatnia korelacja pomiędzy obrotem a zmianą wartości papieru wartościowego i że wolumen jest wyższy dla aktywów, których cena wzrasta niż dla tych, dla których cena spada. Karpoff uważa, że są one tylko pozornie ze sobą sprzeczne i że relacja między ceną a wolumenem najprawdopodobniej nie jest monotoniczna i być może jest różnowartościowa, dlatego badacze dochodzili do odmiennych wniosków.<sup>64</sup>

Kim i in. (1991) przeprowadzili badanie w celu sprawdzenia jak ogłoszenia publiczne wpływają na zmiany cen i wolumenu, oraz czy są zależne od specyfiki ogłoszenia a także od

---

<sup>63</sup>Karpoff, J., 1987, The Relation between Price Changes and Trading Volume: A Survey, *Journal of Financial and Quantitative Analysis*, 22, issue 1, s. 109-110.

<sup>64</sup> Op. cit., s. 110 – 111, 120 – 121.



przekonań inwestorów w momencie udostępnienia ogłoszenia do publicznej wiadomości. Doszli do trzech głównych wniosków:

- Zmiana ceny w momencie udostępnienia ogłoszenia jest proporcjonalna do niespodziewanej części ogłoszenia oraz do względnej istotności wcześniejszych oczekiwań inwestorów.
- Wolumen obrotu jest proporcjonalny do bezwzględnej wartości zmiany ceny. Zmiana ceny odzwierciedla średnią zmianę w przekonaniach inwestorów dotyczących danego ogłoszenia, a wolumen odzwierciedla niezgodność inwestorów co do interpretacji możliwych skutków danego ogłoszenia. Również w związku z tym, że poszczególni inwestorzy mają mniejszy lub większy dostęp do informacji. W związku z tym wolumen odzwierciedla sumę różnic w reakcjach inwestorów, a zmiana w cenie oddaje jedynie uśrednioną reakcję.
- Oczekiwany wolumen i wariancja zmiany ceny są rosnącymi funkcjami precyzji ogłaszanej informacji i malejącymi funkcjami ilości przedogłoszeniowych informacji publicznych i prywatnych.<sup>65</sup>

Dębski (2014) zwraca uwagę na to, że podstawą obliczeń ceny akcji spółki są informacje o jej wynikach i możliwościach powiększania swojego zysku w przyszłości. W wycenie akcji uwzględnia się również fakt, że inwestycja w nie obarczona jest sporym ryzykiem związanym z długoterminowym charakterem inwestycji (co nie pozwala przewidzieć dokładnie przyszłych uwarunkowań rynkowych). Na kształtowanie się cen akcji mają wpływ rozmaite czynniki, czasem przypadkowe, jednakże w dłuższym okresie decydują przede wszystkim czynniki ekonomiczne. Dlatego w badaniach tendencji kursowych bardzo często się je uwzględnia.<sup>66</sup>

---

<sup>65</sup> Kim, O., Verrecchia, R. E., 1991, "Trading Volume and Price Reactions to Public Announcements." *Journal of Accounting Research* 29, no. 2, s. 302-303.

<sup>66</sup> Dębski, Op. cit., s. 216 – 217.

## 2. SIECI NEURONOWE JAKO NARZĘDZIE INWESTYCYJNE

### 2.1. Budowa sieci neuronowych

Uczenie maszynowe to dziedzina z pogranicza statystyki i informatyki, na którą składa się zbiór metod pozwalających (przy zastosowaniu odpowiedniej mocy obliczeniowej) komputerowi na znalezienie zależności pomiędzy danymi zmiennymi i przedstawienie tej/tych zależności w oczekiwany przez autora architektury sposób. Uczenie maszynowe może służyć do rozpoznawania głosu, kształtów na zdjęciach, zaprognozowania przyszłych wartości sprzedaży w przedsiębiorstwie czy tłumaczeniu tekstów z jednego języka na drugi. Przykładów zastosowania go jest znacznie więcej. Sieci neuronowe są szczególnie interesujące, ponieważ mimo iż naukowiec rozumie dokładnie jakie obliczenia i funkcje matematyczne zachodzą w poszczególnych jednostkach obliczeniowych jednej struktury, to nie jest do końca jasne w jaki sposób te obliczenia składają się oczekiwaną całość.

Uczenie maszynowe, a zwłaszcza jego najbardziej skomplikowana obliczeniowo metoda, głębokie sieci neuronowe, nieustannie zyskują na popularności i mnożą się ich zastosowania. Wpływ ma na to wiele czynników, najistotniejszymi są:

- większa powszechna dostępność mocy obliczeniowej, dzięki coraz bardziej zaawansowanym technologicznie komputerom osobistym
- cyfryzacja wiążąca się z większą dostępnością danych w postaci plików komputerowych i baz danych.

Szybki rozwój rozwiązań komercyjnych z tej dziedziny oraz jej rosnąca popularność przyczyniają się do dużego zainteresowania środowiska naukowego. To z kolei wiąże się ze znacznymi postępami i dużym tempem zmian panujących w uczeniu maszynowym trendów. Skalę tych zmian przedstawia między innymi pracownik naukowy MIT Lex Friedman, który na początku każdego roku omawia obecnie panujące trendy i zmiany zaimplementowane przez naukowców, które stają się nową normą.<sup>67</sup>

---

<sup>67</sup> <https://www.youtube.com/watch?v=OVH1Lim8gL8&t=4282s> Deep Learning State of the Art (2020) | MIT Deep Learning Series, 19.11.2020

### 2.1.1. Opis podstaw sieci neuronowych

Uczenie maszynowe to termin parasolowy, jedną z jego odmian jest sieć neuronowa i/lub głęboka sieć neuronowa. Jednak nie wszystkie metody uczenia maszynowego zostały opracowane poprzez nawiązanie do ludzkiego mózgu. Niektóre wykorzystują terminy matematyczne i statystyczne, wzmacniając je zdolnością komputera do przetwarzania wielkich zbiorów danych. Do tej grupy należą na przykład regresja liniowa lub drzewa decyzyjne.

Uczenie maszynowe to pewien rodzaj sztucznej inteligencji – programu, który uczy się, dzięki wykorzystaniu danych (jego działanie nie jest zdefiniowane przez człowieka). Modelem uczenia maszynowego nazywa się algorytm uczący przy użyciu danych. Po treningu składającym się z określonej liczby iteracji (w jednej iteracji wszystkie lub część danych wykorzystywane są do wygenerowania przez algorytm wyników), model wygeneruje finalne dane wynikowe. Tak wytrenowany model stworzy prognozę dla kolejnych dostarczanych mu danych. Wyróżnia się dwa główne typy modeli – klasyfikacyjne (określające przynależność danego przykładu do jednej z grup) oraz regresyjne (dopasowujące jedną wartość do konkretnych wartości atrybutów danego przykładu).<sup>68</sup>

O uczeniu maszynowym można mówić, gdy program komputerowy korzysta z doświadczenia  $E$  w ramach zadania  $T$  i jakiejś miary wydajności  $P$ , gdy spełniony jest warunek, że efektywność zadania  $T$ , mierzona przy pomocy miary  $P$ , wzrasta wraz z doświadczeniem  $E$ . Przy czym w uczeniu maszynowym doświadczeniem  $E$  zazwyczaj nazywa się zbiór danych treningowych i występują różne miary wydajności, zależne od stawianego zadania i cech zbioru danych, tworzących zbiór danych treningowych.<sup>69</sup>

Uczenie maszynowe jest skuteczne w przypadku, gdyby rozwiązanie go przy pomocy tradycyjnych algorytmów wymagało napisania zbyt wielkiej ilości zasad i warunków oraz w przypadku, gdy takie algorytmy musiałyby się relatywnie często zmieniać. Podobnie jest w przypadku zagadnień, w których algorytm byłby zbyt skomplikowany lub nie istnieje żaden znany algorytm, rozwiązujący dany problem. W związku z tym, że metody uczenia maszynowego potrafią same rozpoznawać zależności (między zmiennymi objaśniającymi i zmienną objaśnianą),

---

<sup>68</sup> <https://www.ibm.com/pl-pl/analytics/machine-learning>, 20.11.2020

<sup>69</sup> Mitchell, T., 1997, Machine Learning, McGraw-Hill, Nowy Jork, s. 2.

korzystanie z nich może przyczyniać się do odkrycia nowych, nieznanych wcześniej ludziom zależności. Lub takich, których człowiek nie jest w stanie dostrzec.<sup>70</sup>

Metody uczenia maszynowego dzieli się według kilku kategorii (niewykluczających się wzajemnie). Istotnym kryterium podziału jest rodzaj uczenia – wyróżniamy **uczenie nadzorowane**, **nienadzorowane**, **seminadzorowane** oraz **uczenie ze wzmocnieniem**. W **uczeniu nadzorowanym** dane treningowe algorytmu zawierają pożądane wartości docelowe (zwane etykietami). Przykładem może być zadanie rozpoznania obrazów z kotami. Zakładając, że w zbiorze danych 1/3 obrazów zawiera kota – te obrazy będą oznaczone wartością 1 (prawda), a reszta wartością 0. Do algorytmów uczenia nadzorowanego należą między innymi: k najbliższych sąsiadów, regresja liniowa, regresja logistyczna, maszyna wektorów nośnych, drzewa decyzyjne i lasy losowe oraz sieci neuronowe (nie wszystkie). W **uczeniu nienadzorowanym** nie nadaje się etykiet, czyli algorytm sam próbuje znaleźć zależność między danymi. Do algorytmów tej grupy należą: analiza skupień (algorytm centroidów, grupowanie hierarchiczne, maksymalizacja wartości oczekiwanej), wizualizacja i redukcja wymiarowości (analiza głównych składowych, węzeł t-SNE), algorytmy oparte o reguły asocjacyjne (ECLAT, Apriori). Algorytmy nienadzorowane mogą służyć na przykład do generowania reklam pod grupy odbiorców zidentyfikowane na danej stronie internetowej lub do wykrywania anomalii, na przykład w używaniu kart kredytowych, co pozwala zidentyfikować oszustwa i kradzieże środków. **Seminadzorowane sieci** uczone są częściowo etykietowanymi danymi – zazwyczaj więcej jest w nich danych nieoznaczonych. Przykładami takich sieci są na przykład RBM, które są wielowarstwowymi sieciami neuronowymi, z których pierwsze działają na zasadach nienadzorowanych, a ich wyniki są ulepszane przez zastosowanie warstw sieci nadzorowanych.<sup>71</sup>

**Uczenie przez wzmocnienie** różni się od powyższych metod, ponieważ nie polega na sprawdzaniu dopasowania danych do etykiet ani na próbie znalezienia zależności między danymi, a raczej na stworzeniu behawioralnego modelu uczenia się metodą prób i błędów. Algorytm otrzymuje informację zwrotną po przeanalizowaniu danych. W związku z metodą prób i błędów

---

<sup>70</sup> Géron, A., 2017, *Hands-On Machine Learning with Scikit-Learn and Tensorflow*, O'Reilly Media, Sebastopol, s. 3 - 13

<sup>71</sup> Ibidem.

sekwencja właściwych decyzji spowoduje wzmocnienie procesu, jako iż będzie ona najlepszym rozwiązaniem danego problemu.<sup>72</sup>

Kolejnym kryterium podziału uczenia maszynowego jest rozróżnienie sposobów nauki – wyróżnia się **batch learning** oraz **online learning**. **Batch learning** polega nauczeniu przy użyciu wszystkich dostępnych danych zbioru treningowego na raz. Taki proces zazwyczaj zajmuje dużo czasu i wymaga wykorzystania dużej mocy obliczeniowej, więc przeprowadza się go *offline*. System jest uczony, a później uruchamia się go w wersji produkcyjnej i wtedy już się nie uczy. Użycie nowych danych do nauczenia modelu rozwiniętego pod *batch learning* wymaga nauczenia całego modelu od zera (przy wykorzystaniu starych i nowych danych). W niektórych przypadkach, w których nie ma dostępu do dużej mocy obliczeniowej (jak na przykład w przypadku aplikacji na smartfon), stosuje się **online learning**, które polega na uczeniu systemu przyrostowo – dostarczając nowych danych w postaci sekwencji (pojedynczo lub w małych porcjach). Systemy uczące się w sposób *online* dobrze nadają się do zastosowań, w których nowe dane nieustannie wpływają i się zmieniają, na przykład w przypadku badania zależności cen giełdowych rozpatrując dane pochodzące na żywo z sesji (tikowe).<sup>73</sup>

Wiele problemów związanych ze sztuczną inteligencją można rozwiązać poprzez określenie właściwych atrybutów (zmiennych) i przekazanie ich prostemu algorytmowi uczenia maszynowego. Dla podania przykładu – przydatną cechą w rozpoznawaniu głosu mówcy z dźwięku jest wysokość tonu głosu, którą można zapisać w sposób formalny przy użyciu spektrogramu częstotliwości dźwięku. Ton pomaga w określeniu płci oraz wieku mówcy. Jednak w przypadku wielu zagadnień trudno jest zidentyfikować, które zmienne są dla nich istotne. Przykładem tego może być próba zidentyfikowania samochodów na zdjęciach – każdy samochód ma koła, więc sensownym byłoby określenie tego w modelu. Jednak trudno jest dokładnie sprecyzować jak wygląda koło z punktu widzenia wartości pikseli. Mimo dość prostego geometrycznego kształtu, obraz koła może być zniekształcony na przykład przez cień na nie padający lub przez słońce odbijające się w jego metalowej feldze.<sup>74</sup>

---

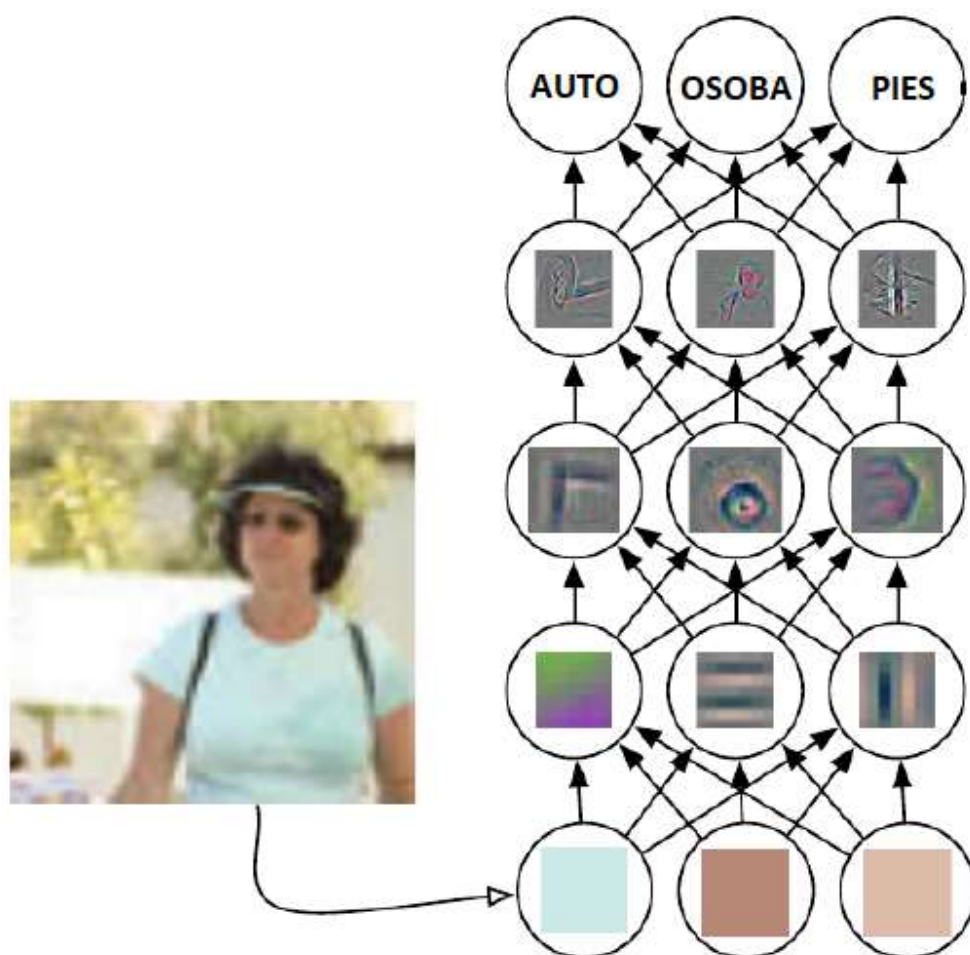
<sup>72</sup> <https://www.ibm.com/pl-pl/analytics/machine-learning> , 21.11.2020

<sup>73</sup> Géron, Op. Cit., s. 14 – 20

<sup>74</sup> Goodfellow, I. J., Bengio, Y., Courville, A., 2016, Deep Learning, MIT Press, Cambridge, s. 3 - 5

Głównym źródłem trudności w wielu rzeczywistych zastosowaniach sztucznej inteligencji jest to, że w prawdziwym (nieidealnym) świecie wiele czynników, które są istotne dla zagadnienia ulega różnym zakłóceniom i szumom – pojedyncze piksele zdjęcia czerwonego samochodu zrobionego nocą mogą mieć kolor bardzo zbliżony do czarnego. Głębokie uczenie (*Deep Learning*) rozwiązuje ten problem poprzez wprowadzenie warstwowego podejścia do problemu, w którym każda kolejna warstwa jest bardziej rozwinięta i składa się z elementów poprzedniej warstwy. Głębokie uczenie pozwala komputerowi zbudować skomplikowany obraz z mniejszych (lub mniej zawiłych) elementów z których się składa, jak zostało to przedstawione na rysunku 6.<sup>75</sup>

Rys. 6. Przykładowy schemat działania głębokiej sieci neuronowej rozpoznającej obraz.



Źródło: Opracowanie własne na podstawie: Goodfellow, I. J., Bengio, Y., Courville, A., 2016, Deep Learning, MIT Press, Cambridge, s. 6.

<sup>75</sup> Ibidem.

W XX wieku, gdy technologia uczenia maszynowego zaczęła się rozwijać, największym ograniczeniem ilości neuronów i połączeń między nimi były możliwości sprzętowe. W obecnych czasach jest to bardziej kwestia wyboru badacza. Niektóre sieci neuronowe mają niemal tyle połączeń neuronowych co kot, a powszechną praktyką jest stosowanie co najmniej tylu neuronów, ile znajduje się w mózgach mniejszych ssaków takich jak myszy. Rozrzucenie i tylko częściowa łączność między neuronami znajdującymi się w ludzkim mózgu sprawia, że wykorzystując sztuczną sieć neuronową dla nauczenia komputera jakiejś konkretnej zależności między danymi, wystarczy znacznie mniejsza liczba neuronów niż w przypadku ludzkiego mózgu (dla osiągnięcia takiej samej lub nawet większej skuteczności).<sup>76</sup>

### 2.1.2. O danych w sieciach neuronowych

Osiągnięcie zadowalających rezultatów przy użyciu modelu głębokiego uczenia wymaga od badacza umiejętności i wiedzy, ale im większa jest baza danych z której można wytrenować dany algorytm, tym łatwiej jest to osiągnąć. Algorytmy które dziś wykorzystuje się w uczeniu maszynowym niewiele zmieniły się od lat 1980tych, najważniejszym napędem rozwojowym tej technologii jest ilość danych, które można im dostarczać. Postępująca cyfryzacja społeczeństwa ułatwia ten proces, jako że coraz więcej czynności i codziennych obowiązków wykonywana jest za pomocą komputera. Okres *Big Data* przyczynił się do ułatwienia wykorzystania sieci neuronowych – głównie przez pozbycie się najbardziej obciążającego ich skuteczność błędu – estymacji statystycznej występującej w przypadku próby wyciągania wniosków z danych, bazując na małej ilości dostępnych informacji. W 2015 roku niepisaną zasadą było, że algorytmy nadzorowanego uczenia osiągają zadowalającą skuteczność, jeśli wykorzysta się około 5000 etykietowanych przykładów na kategorię, a zrówna się lub nawet przebije ludzką skuteczność, jeśli wykorzysta się zbiór danych liczący co najmniej 10 milionów obserwacji.<sup>77</sup>

Géron (2017), rozważając największe wyzwania i trudności w stosowaniu obecnie uczenia maszynowego, zwraca uwagę przede wszystkim na aspekty związane z danymi. Brak dostatecznej ilości danych jest jedną z głównych przeszkód skutecznego uczenia modelu. Zwiększanie bazy danych treningowych zazwyczaj przynosi lepsze rezultaty niż próby usprawniania algorytmu – pod warunkiem, że dane są dobrej jakości. Przez dane złej jakości rozumie się mocno zaszumione dane,

---

<sup>76</sup> Goodfellow i in., Op. cit., s. 21

<sup>77</sup> Goodfellow i in., Op. cit., s. 17-18

w których dużo jest losowości, częściowo wybrakowanych obserwacji lub błędów, które pojawiają się tam ze względu na złe dokonanie pomiaru. Algorytmy głębokiego uczenia są częściowo odporne na takie braki, ale im jest ich mniej, tym lepszy osiąga się rezultat. Dla większości specjalistów zajmujących się uczeniem maszynowym pierwsza część projektu to oczyszczanie danych – na przykład usuwanie skrajnych (mocno różniących się od pozostałych) wartości lub uzupełnianie brakujących obserwacji medianą wyliczoną dla wartości tej cechy w innych obserwacjach. Obserwacje powinny być również reprezentatywne dla nowych danych, dla których chce się dokonywać regresji albo klasyfikacji.<sup>78</sup>

### 2.1.3. Trenowanie modeli

Zrozumienie sposobu działania algorytmów uczenia maszynowego jest przydatne przy formułowaniu własnych sieci i wniosków płynących z ich wyników. Współcześnie istnieje wiele rodzajów oprogramowania z interfejsami, które ułatwiają użytkownikowi korzystanie z tej technologii (zazwyczaj odbierając część dowolności w tworzeniu sieci), ale pewien stopień ich znajomości jest konieczny dla zrozumienia otrzymywanych wyników. Sieci neuronowe korzystają z wiedzy i osiągnięć różnych dziedzin nauki – matematyki, **statystyki**, **zasad prawdopodobieństwa**, **analizy numerycznej** i innych, pokrewnych dziedzin. Szczególną istotność dla zagadnienia prezentują operacje na wartościach skalarnych, wektorach, macierzach czy tensorach oraz zrozumienie zależności różnego stopnia pomiędzy zmiennymi w funkcjach, w tym wiedza o pochodnych kolejnych stopni. **Pojęcia statystyczne** są z kolei istotne już w procesie przetwarzania danych, których używa się do uczenia sieci. Obserwacje na temat średniej, mediany, odchylenia, rozkładu wartości, kurtozy, czy statystycznej istotności mogą pomóc w oszacowaniu kroków potrzebnych przy przetwarzaniu danych, które będą stanowić atrybuty sieci oraz przy wyborze odpowiedniego dla zadania algorytmu. **Prawdopodobieństwo** jest istotne z punktu widzenia uczenia maszynowego, ponieważ algorytmy uczenia maszynowego zazwyczaj są stochastyczne, czyli do pewnego stopnia losowe i bazujące na niepewnych wynikach – co może być przewagą tej technologii. W rzeczywistości trudno jest mówić o wydarzeniach, które są absolutnie prawdziwe lub takich, które na pewno będą miały miejsce. Dla uczenia maszynowego szczególnie ważne z tego pola są zagadnienia dotyczące zmiennych losowych, rozkładu prawdopodobieństwa, gęstości prawdopodobieństwa, warunkowego prawdopodobieństwa,

---

<sup>78</sup> Géron, Op. Cit, s. 22 - 25



wartości oczekiwanej, wariancji czy kowariancji. **Analiza numeryczna** stanowi podstawę uczenia i usprawniania algorytmów uczenia maszynowego, które zazwyczaj polegają na optymalizacji zadanej funkcji (najczęściej na minimalizacji funkcji starty) – z tego też powodu wiedza o regule łańcuchowej, optymalizacji z ograniczeniami oraz o metodach optymalizacyjnych takich jak metoda najmniejszych kwadratów czy metoda gradientu prostego jest istotna dla zrozumienia i korzystania z algorytmów samouczących.<sup>79</sup> Wiedza z zakresu informatyki również ułatwia korzystanie z algorytmów i pozwala lepiej zrozumieć sposób, w jaki komputer tworzy neurony i przeprowadza z nich obliczenia (na przykład w jaki sposób dzieli porcje danych na poszczególne rdzenie układu GPU).

Większość algorytmów uczenia maszynowego służy do weryfikacji wcześniej postawionej hipotezy badawczej – sprawdza się zależności między obserwacjami i analizuje ich statystyki. Zauważając zależność, bądź spodziewając się, że w dana zależność istnieje, stawia hipotezę i próbuje zbudować model, który na podstawie próby (zbioru treningowego) nauczy się, jak można przedstawić tę zależność (lub raczej – jaką wartość przyjmie zmienna objaśniana dla nowej obserwacji tego samego problemu z konkretnymi wartościami parametrów). Jeżeli na zbiorze takich nowych obserwacji (zbiór testowy) model poprawnie przewidzi wartość zmiennej objaśnianej, można wnioskować, że istnieje zależność pomiędzy parametrami modelu. Najłatwiejsza do zauważenia i opisanie jest zależność liniowa, na przykład w przypadku, gdy próbuje się określić wpływ PKB na mieszkańca na satysfakcję z życia mieszkańca danego kraju:

$$\text{satysfakcja} = \theta_0 + \theta_1 * \text{PKB na mieszkańca} \quad (2.1)$$

Przeprowadzając analizę danych z OECD można zaobserwować, że mimo pewnego szumu w danych, istnieje dodatnia korelacja pomiędzy PKB a satysfakcją z życia, której model mógłby się nauczyć. W przypadku takiej regresji liniowej (zasady byłyby podobne również dla sieci neuronowych) zmienianie parametrów  $\theta_0$  oraz  $\theta_1$  pozwala na przedstawienie dowolnej zależności liniowej pomiędzy atrybutami. W celu nauczania algorytmu jakie wartości powinny przyjmować parametry, należy najpierw wybrać funkcję odpowiedniości (która zmierzy jak dokładnie model jest dopasowany) lub funkcję straty (która zbada, jak dużo brakuje modelowi do poprawnego dopasowania). W przypadku regresji liniowej często stosuje się sumę kwadratów reszt jako funkcję

---

<sup>79</sup> Goodfellow i in., Op. cit., s. 28 - 90

straty. Jeśli model wytrenowany na podstawie funkcji straty wykazuje niską trafność przewidywania wartości dla nowych obserwacji, może to oznaczać, że potrzebuje więcej atrybutów, lepszej jakości danych lub użycia innego typu modelu (być może zależność między danymi nie jest liniowa).<sup>80</sup>

Zrozumienie zasad działania uczenia się sieci neuronowej jest podstawą w dobrym doborze architektury pod konkretne zadanie oraz usuwaniu błędów i uzyskiwaniu lepszych wyników poprzez zmienianie hiperparametrów. Korzystanie z sieci neuronowych i zaawansowanych metod uczenia maszynowego zazwyczaj wymaga korzystania z wielu atrybutów dla uzyskania zadowalającego rezultatu – jak na przykład w modelu, w którym średnie wynagrodzenie wyjaśnione jest za pomocą wieku, płci, wielkości miasta zamieszkania, ilości dzieci i tak dalej. Każda kolejna zmienna objaśniająca podwyższa stopień wymiaru, w którym działa algorytm uczący sieci – dlatego też między innymi ze względów obliczeniowych zaleca się ograniczanie ilości atrybutów. Ponadto, wizualizowanie procesu działania algorytmów uczących w sposób łatwy da się przeprowadzić tylko dla problemu z dwiema zmiennymi objaśniającymi (wagi ich parametrów zestawione są z błędem – funkcją straty, wizualizując się w trzech wymiarach). Rozważając model liniowy:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_n x_n \quad (2.2)$$

w którym  $y$  to zmienna objaśniana,  $n$  to liczba zmiennych (atrybutów),  $x_i$  to wartość  $i$ -tej obserwacji danej zmiennej,  $\theta_j$  jest  $j$ -tym parametrem modelu, a  $\theta_0$  obciążeniem modelu, jego optymalizacja polegałaby na znalezieniu takich wartości parametrów  $\theta$ , dla których miara błędu byłaby najniższa. Zakładając, że jest to zadanie regresyjne, najczęściej korzysta się z błędu średniokwadratowego lub z średniej kwadratowej błędów. Ta pierwsza miara w przypadku powyżej opisanego modelu miałaby postać:

$$\text{MSE (Mean Squared Error)} = \frac{1}{m} \sum_{i=1}^m (\theta^T * x^{(i)} - y^{(i)})^2 \quad (2.3)$$

Równanie zostało zapisane w postaci wektorowej, więc  $\theta$  jest wektorem parametrów, zawierającym w sobie obciążenie i wartości parametrów  $\theta$  dla wszystkich atrybutów. W równaniu  $\theta^T$  oznacza postać transponowaną tego wektora, żeby można było przeprowadzić jego

---

<sup>80</sup> Géron, Op. Cit, s. 18 - 22

mnożenie skalarne z wektorem wartości zmiennych  $x$ . MSE jest dobrą miarą efektywności modelu, ponieważ wyciąga średnią z podniesionej do kwadratu (niezależnie, czy wartość jest mniejsza od właściwej czy większa, wykaże taką samą wartość) różnicy pomiędzy rzeczywistymi wartościami  $y$ , a ich estymacją dokonaną przez model. W przypadku regresji liniowej skuteczną metodą optymalizacji (odnalezienia takiego  $\theta$ , dla którego funkcja straty jest jak najmniejsza) jest klasyczna metoda najmniejszych kwadratów, którą można opisać wzorem:

$$\hat{\theta} = (X^T * X)^{-1} * X^T * y \quad (2.4)$$

gdzie  $y$  to wektor wartości zmiennej objaśnianej dla wszystkich obserwacji. Taki sposób trenowania algorytmu jest dokładny, ale powolny dla dużych zbiorów danych. Jeśli wektory i macierze omawiane w powyższym przykładzie nie zmieściłyby się na raz w pamięci komputera, nie można byłoby przeprowadzić optymalizacji za pomocą metody najmniejszych kwadratów.<sup>81</sup>

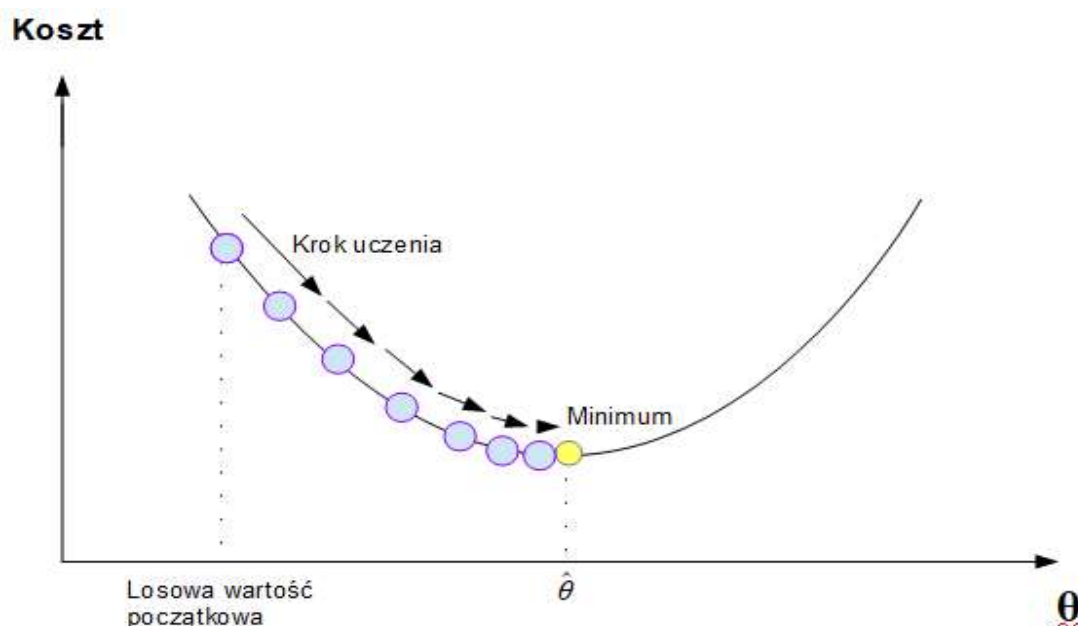
Dla dużych zbiorów danych w regresji liniowej i dla optymalizacji innych, bardziej złożonych obliczeniowo algorytmów, korzysta się z mniej dokładnych, przybliżających iteracyjnie metod (między innymi *stochastic gradient descent*, *batch gradient descent*, *AdaGrad*, *RMSProp*, *Adam*), których bazę stanowi metoda gradientu prostego. Metoda gradientu prostego jest generycznym optymalizacyjnym algorytmem zdolnym do odnajdywania optymalnych rozwiązań dla szerokiej gamy problemów. Najważniejszym jej założeniem jest iteratywne, stopniowe zmienianie parametrów w celu minimalizacji funkcji kosztu. Algorytm mierzy lokalny gradient funkcji błędu dla wartości wektora parametrów  $\theta$ , a następnie zmienia te parametry w taki sposób, że gradient będzie malał wraz z kolejnymi iteracjami. Minimum funkcji jest osiągnięte w momencie, gdy gradient wynosi zero. Wywołując tę metodę najczęściej nadaje się wektorowi parametrów  $\theta$  losowe wartości, które algorytm będzie stopniowo zmieniał, dążąc do optymalizacji funkcji poprzez minimalizację funkcji straty.<sup>82</sup>

---

<sup>81</sup> Géron, Op. Cit, s. 105 – 142.

<sup>82</sup> Ibidem.

Rys. 7. Symulacja działania metody gradientu prostego dla funkcji o jednym parametrze theta (lub wektorze parametrów theta).

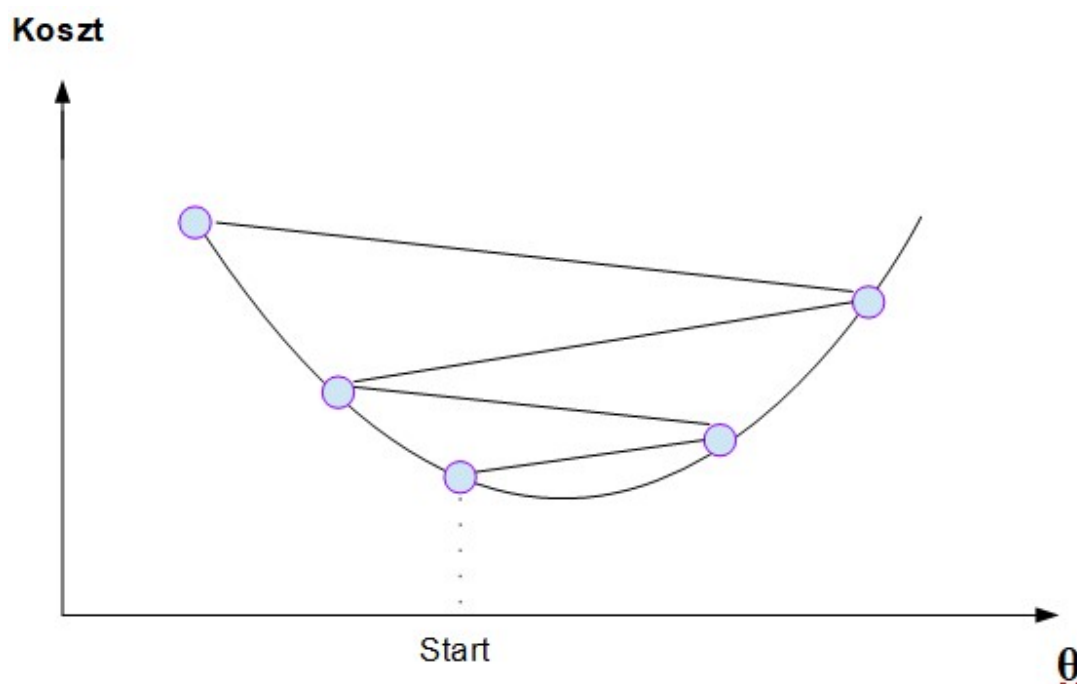


Źródło: Opracowanie własne na podstawie: Géron, A., 2017, Hands-On Machine Learning with Scikit-Learn and Tensorflow, O'Reilly Media, Sebastopol, s. 111.

Najistotniejszym hiperparametrem takiego algorytmu jest współczynnik uczenia, który steruje tym, jak dużej zmiany w wektorze parametrów może dokonać algorytm podczas jednej iteracji uczenia. Jeśli współczynnik będzie za mały – algorytm może wymagać zbyt wielu iteracji uczenia, żeby było to opłacalne. Jeśli byłby za duży – wartość funkcji straty może skakać z jednego ramiona paraboli na drugi i nigdy nie znaleźć najniższej wartości funkcji straty możliwej dla tej funkcji (jak na rysunku 8.).<sup>83</sup>

<sup>83</sup> Ibidem.

Rys. 8. Możliwy problem z odnalezieniem minimum funkcji przy zastosowaniu zbyt dużego współczynnika uczenia.

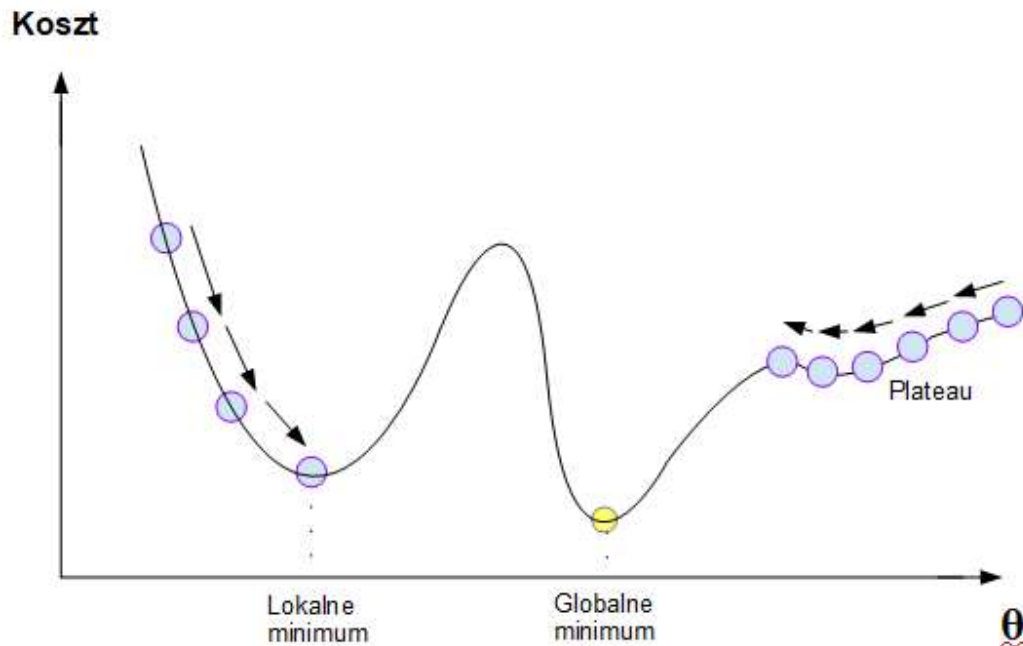


Źródło: Opracowanie własne na podstawie: Géron, A., 2017, Hands-On Machine Learning with Scikit-Learn and Tensorflow, O'Reilly Media, Sebastopol, s. 112.

Ponadto w rzeczywistości funkcje straty rzadko miewają regularne kształty jak te przedstawione na powyższych ilustracjach, bardziej prawdopodobne są niemonotoniczne i wypukłe funkcje (takie jak ta przedstawiona na rysunku 9.), co zmniejsza szanse algorytmu na odnalezienie globalnego minimum. W przypadku inicjalizacji parametru  $\theta$  jak po lewej stronie rysunku 9., algorytm może zatrzymać się w lokalnym minimum. Natomiast w przypadku inicjalizacji jak po prawej stronie algorytm długo będzie szukał globalnego minimum lub, przy zbyt małej ilości iteracji, nigdy go nie znajdzie.<sup>84</sup>

<sup>84</sup> Ibidem.

Rys. 9. Potencjalne trudności w odnalezieniu globalnego minimum.



Źródło: Opracowanie własne na podstawie: Géron, A., 2017, Hands-On Machine Learning with Scikit-Learn and Tensorflow, O'Reilly Media, Sebastopol, s. 113.

Implementacja metody gradientu prostego polega na obliczeniu gradientu dla funkcji straty w odniesieniu do każdego z parametrów modelu ( $\theta_1, \dots, \theta_n$ ), a więc na sprawdzeniu cząstkowych pochodnych parametrów względem tej funkcji:

$$\frac{\partial}{\partial \theta_j} MSE(\theta) = \frac{2}{m} \sum_{i=1}^m (\theta^T * x^{(i)} - y^{(i)}) x_j^{(i)} \quad (2.5)$$

Operację tę należy wykonać w stosunku do każdego parametru, a wyniki utworzą wektor gradientu funkcji straty, który można zapisać również w postaci:

$$\nabla_{\theta} MSE(\theta) = \frac{2}{m} X^T * (X * \theta - y) \quad (2.6)$$

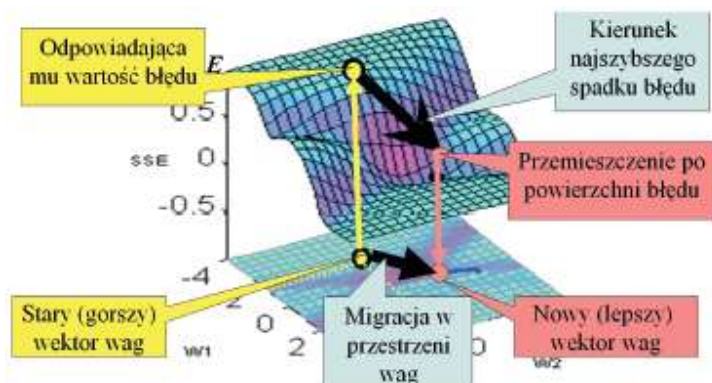
gdzie  $X$  stanowi macierz wszystkich wartości dla wszystkich obserwacji zmiennych objaśniających. Po obliczeniu wektora gradientów algorytm podejmuje decyzje w jaki sposób zmienić parametry, żeby zmniejszyć funkcję straty w następujący sposób:

$$\theta^{(kolejna\ iteracja)} = \theta - \eta \nabla_{\theta} MSE(\theta) \quad (2.7)$$

gdzie  $\eta$  oznacza współczynnik uczenia.<sup>85</sup>

Proces w formie graficznej dobrze oddaje wykres przedstawiony na rysunku 10. dla modelu z dwoma parametrami (osie X i Y), gdzie algorytm aktualizuje wagi (parametry) w taki sposób, by w kolejnej iteracji znaleźć się bliżej minimum dla funkcji straty przedstawionej na osi Z.<sup>86</sup>

Rys. 10. Istota procesu uczenia polega na posuwaniu się w kierunku spadku powierzchni błędu.



Źródło: Tadeusiewicz, R., 2007, Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#, Polska Akademia Umiejętności, Kraków, s. 124

W przypadku funkcji wypukłych (posiadających więcej niż jedno minimum lokalne) algorytm sprawdza także pochodne drugiego rzędu w celu określenia, w jaki sposób zmiana pierwszej pochodnej wpłynie na funkcję podstawową, co pozwala określić (w niektórych przypadkach), czy w chwili obecnej funkcja nie znajduje się w lokalnym minimum (zamiast globalnego). Jeśli algorytm bazuje na wielowymiarowej funkcji, konieczne jest zbadanie pochodnych drugiego rzędu dla wszystkich parametrów i stworzenie z nich hesjanu. Jeśli wszystkie wartości własne w takiej macierzy są dodatnie, punkt jest lokalnym minimum. Jeśli natomiast wszystkie wartości własne są negatywne, punkt jest lokalnym maksimum.<sup>87</sup>

#### 2.1.4. Trudności i wyzwania

Poza problemami związanymi ze zbyt małą ilością danych lub złą ich jakością, programista sieci neuronowych mierzy się także z kilkoma innymi wyzwaniami. Pierwszym jest **dobór odpowiednich atrybutów(zmiennych)**, które wpływają na zmienną objaśnianą. Mimo iż algorytmy są częściowo odporne na niewłaściwy dobór zmiennych, to zbyt duża liczba takich,

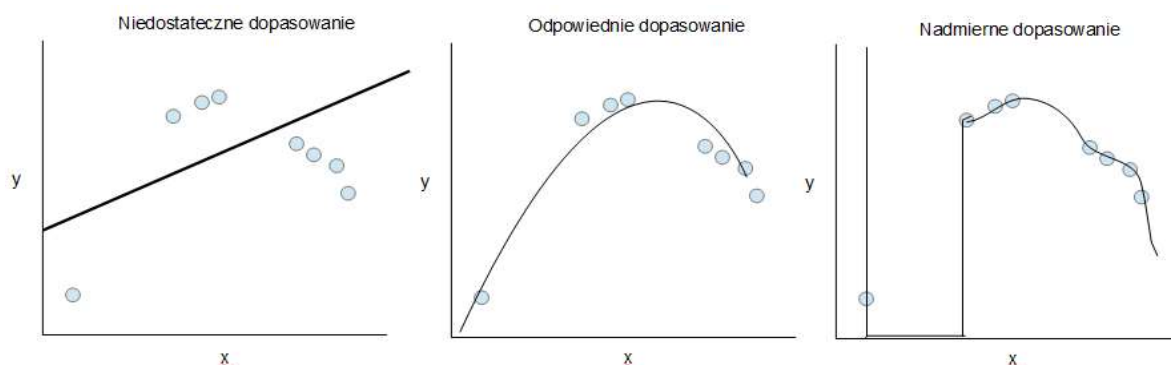
<sup>85</sup> Ibidem.

<sup>86</sup> Tadeusiewicz, R., 2007, Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#, Polska Akademia Umiejętności, Kraków, s. 124.

<sup>87</sup> Goodfellow i in., Op. Cit., s. 83 - 84

które nie są powiązane ze zmienną objaśnianą, powoduje pogorszenie się wyników takiej sieci. Kluczową częścią procesu korzystania z algorytmu uczenia maszynowego jest **inżynieria atrybutów (*feature engineering*)**, czyli proces doboru atrybutów oraz ich dostosowywania (na przykład tworzenie nowych atrybutów poprzez operację matematyczną na dwóch pozostałych). Kolejnym wyzwaniem jest **odpowiednie dopasowanie algorytmu (poprzez uczenie) do obserwacji**, czyli dopasowanie w taki sposób, by przy danych wsadowych, które wykorzystuje się do uczenia go, uzyskiwał dokładny wynik, a przy nowych danych, dla których chce się wykonać predykcję, dawał zadowalający wynik, który jest niewiele gorszy niż ten dla danych treningowych. W przypadku w którym algorytm bardzo dobrze dopasowuje się do danych treningowych, ale osiąga znacznie gorsze wyniki, jeśli zostaną podane mu nowe dane, nastąpiło **nadmierne dopasowanie**. W takim wypadku badacz powinien: ograniczyć liczbę iteracji uczenia, uprościć model (na przykład poprzez zmniejszenie liczby neuronów w sieci neuronowej), poprawić dane (lub dostarczyć lepsze dane) lub powiększyć zbiór danych służący do uczenia (lub zastosować kombinację wymienionych metod). Jeśli natomiast zbiór nie potrafi przewidywać trafnie nawet na danych, na których został nauczony, zachodzi **zjawisko niedostatecznego dopasowania**. W takim wypadku rozwiązaniem może być wybranie bardziej złożonego modelu (na przykład więcej neuronów lub przejście z regresji liniowej na logarymiczną), dodanie iteracji uczenia, zmiana/dodanie nowych atrybutów lub dodanie obserwacji do zbioru danych, który wykorzystywany jest do uczenia. Problem obrazuje rysunek 11., w którym punkty na wykresie to wartości zmiennej objaśnianej dla danych wartości zmiennej  $x$ .<sup>88</sup>

Rys. 11. Rodzaje dopasowania modelu do danych.

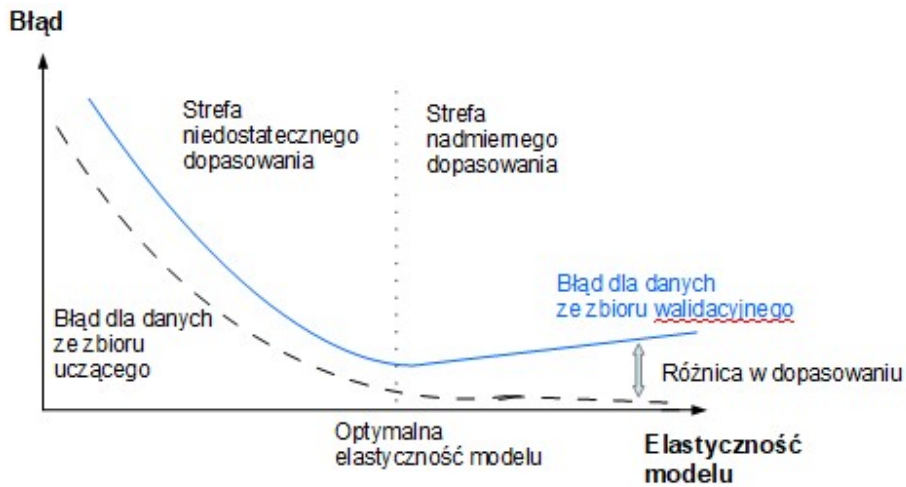


Źródło: Opracowanie własne na podstawie: Goodfellow, I. J., Bengio, Y., Courville, A., 2016, Deep Learning, MIT Press, Cambridge, s. 105

<sup>88</sup> Géron, Op. Cit, s. 25 - 28



Rys. 12. Błąd w relacji do elastyczności modelu.



Źródło: Opracowanie własne na podstawie: Goodfellow, I. J., Bengio, Y., Courville, A., 2016, Deep Learning, MIT Press, Cambridge, s. 107.

Elastyczność modelu, czyli zakres funkcji spośród których model ma dobierać takie o najlepszym dopasowaniu do danych, istotnie wpływa na rozbieżność pomiędzy dopasowaniem modelu do danych uczących, a danych ze zbioru walidującego. Im większa swoboda modelu w doborze funkcji, tym większa szansa na bardzo dobre dopasowanie do zbioru treningowego. W przypadku zbyt dużej elastyczności modelu dochodzi do nadmiernego dopasowania i model z niewielką poprawnością przewiduje wartości dla nowych, niewidzianych wcześniej danych. W przypadku zbyt małej elastyczności, model w ogóle nie będzie potrafił dopasować się do danych. Zaleca się, aby elastyczność modelu była odpowiednia dla: poziomu skomplikowania samego zagadnienia, które dany model analizuje oraz dla ilości danych w zbiorze treningowym, którymi badacz dysponuje.<sup>89</sup>

## 2.2. Rodzaje modeli

Istnieje wiele rodzajów algorytmów uczenia maszynowego. W najbardziej ogólnym sensie można dokonać ich podziału na metody klasyczne oraz na sieci neuronowe i ich pochodne. Ta pierwsza grupa składa się między innymi z: regresji liniowej, regresji logistycznej, naiwnego klasyfikatora bayesowskiego, maszyny wektorów nośnych, drzew decyzyjnych, K-najbliższego sąsiada, lasów losowych, wzmacniania gradientowego i K-średnich.<sup>90</sup> W części badawczej tej

<sup>89</sup> Goodfellow i in., Op. Cit., s. 103 – 107.

<sup>90</sup> <https://azure.microsoft.com/pl-pl/overview/machine-learning-algorithms/#uses>, 07.12.2020

pracy do przewidywania przyszłych wartości walorów zostaną wykorzystane sieci neuronowe, więc to na ich opisie skupiony będzie ten rozdział.

Podziału sieci neuronowych można dokonać ze względu na różne kryteria – najbardziej bazowym z nich jest liczba warstw – wyróżnia się sieci **jednowarstwowe** (posiadające tylko jedną warstwę neuronów) oraz **wielowarstwowe** (posiadające  $n$  warstw neuronów). Sieci z których obecnie korzysta się w rozwiązaniach komercyjnych najczęściej są wielowarstwowe (często są to sieci składające się z wielu mniejszych sieci neuronowych – uczenie wzmacnione). W zależności od tego, jak zbudowane są pojedyncze neurony sieci wielowarstwowej (i jak są między sobą połączone), wyróżnia się **perceptron wielowarstwowy**, **sieci rekurencyjne**, **sieci konwolucyjne** i **autoenkodery**. Przy czym każda z tych grup ma wiele wariantów, dostosowanych do różnych zastosowań i danych.<sup>91</sup> Ta praca będzie bazowała na wykorzystaniu rekurencyjnych sieci neuronowych, z powodów opisanych w dalszej części tego rozdziału. Sieci rekurencyjne zbudowane są z neuronów bazujących na zmodyfikowanej budowie perceptronu.

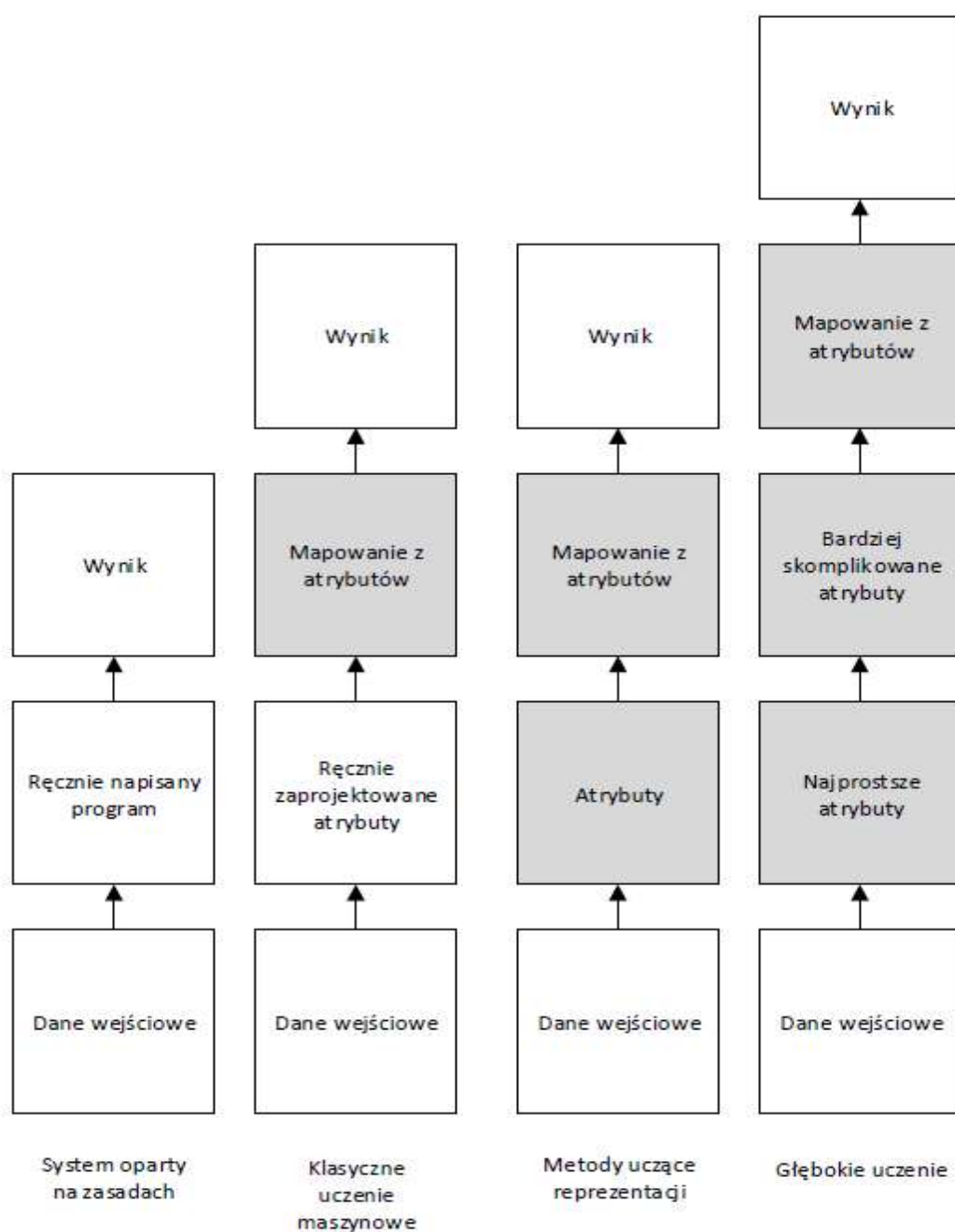
Goodfellow i in. (2016) dodają do podziału również metody uczące reprezentacji. Poniższa ilustracja wskazuje, że im bardziej skomplikowany model (i im więcej warstw posiadający), tym więcej potrafi nauczyć się bez ingerencji człowieka. Podstawowym sposobem rozwiązywania problemów jest wykorzystanie ręcznie napisanego programu, który wykonuje opisane polecenie, nie dostosowując się do niego w żaden sposób. Na rysunku 13. w innych metodach na szaro oznaczone są te części procesu, których algorytm jest w stanie nauczyć się sam na podstawie danych.<sup>92</sup>

---

<sup>91</sup> Goodfellow i in., Op. Cit., s. 1 – 25.

<sup>92</sup> Ibidem.

Rys. 13. Porównanie schematów algorytmów rozwiązujących problemy na bazie danych.



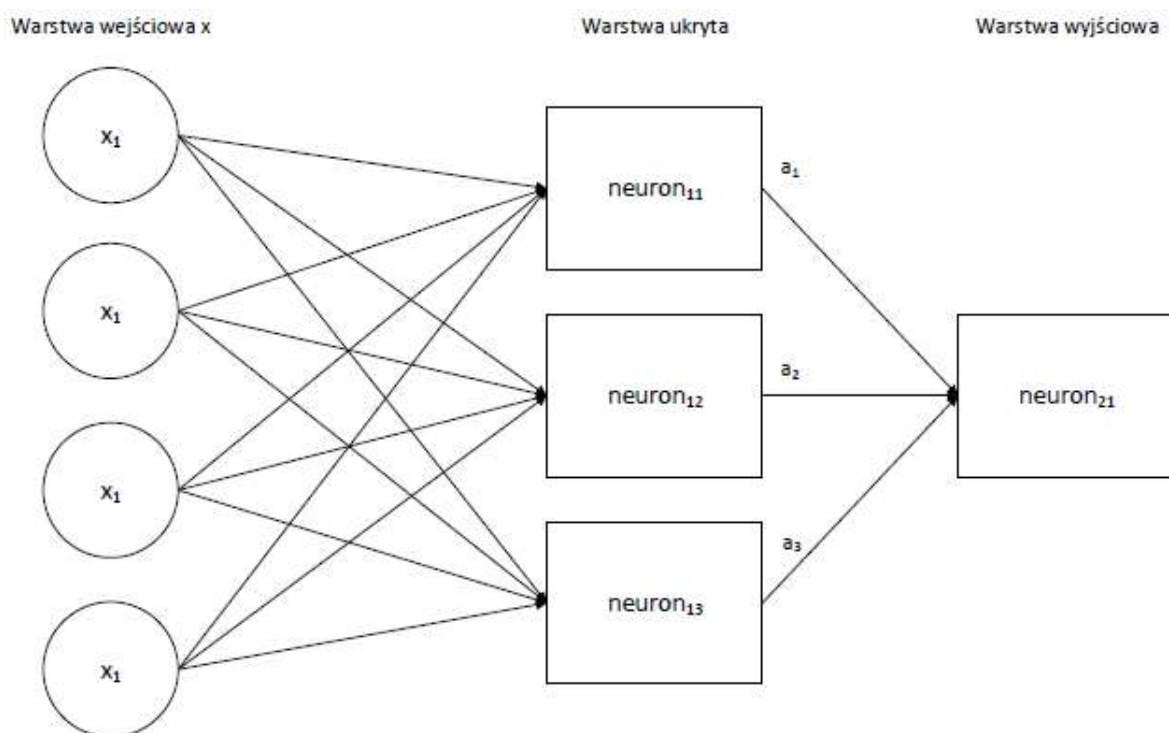
Źródło: Opracowanie własne. Na podstawie: Goodfellow, I. J., Bengio, Y., Courville, A., Deep Learning, MIT Press, Cambridge, s. 10.

### 2.2.1. Perceptron wielowarstwowy

Opis budowy neuronu w sieci rekurencyjnej powinien być poprzedzony analizą budowy perceptronu (i stworzonej z perceptronów sieci), ponieważ neurony w sieci rekurencyjnej są zmodyfikowaną wersją perceptronów.

Podstawowym przykładem uczenia głębokiego jest jednokierunkowa głęboka sieć neuronowa lub wielowarstwowy perceptron (*MLP*). Perceptron wielowarstwowy to po prostu matematyczna funkcja powiązująca zestaw danych wejściowych z wyjściowymi wartościami. Funkcja główna jest złożona z wielu prostszych funkcji. Można to rozumieć w taki sposób, że każda z tych mniejszych funkcji dostarcza nowego sposobu reprezentacji danych wejściowych.<sup>93</sup>

Rys. 14. Schemat budowy prostej, dwuwarstwowej sieci neuronowej.



Źródło: Opracowanie własne. Na podstawie Géron, A., 2017, Hands-On Machine Learning with Scikit-Learn and Tensorflow, O'Reilly Media, Sebastopol, s. 261.

<sup>93</sup> Goodfellow i in., Op. Cit. s. 5.

Wyróżnia się dwa główne typy perceptronów – liniowy, którego wynik (ostateczny wynik sieci lub cząstkowy wynik dla jednego neuronu, przekazywany do kolejnej warstwy sieci) to:

$$y = x_1w_1 + x_2w_2 + \dots + x_nw_n \quad (2.8)$$

czyli suma wartości obserwacji atrybutów pomnożonych przez wagi. Powyższa suma w postaci wektorowej ma postać:  $y = w^T * x$ . W przypadku perceptronu, wagi to parametry, których uczy się sieć (iteracyjnie poprawiając ich wartości na przykład przy pomocy metody gradientu prostego). Liniowość neuronu wiąże się jednak z pewnymi ograniczeniami: a) jeśli zadaniem danej sieci jest klasyfikacja, której dokonuje się poprzez określenie prawdopodobieństwa, że dany obiekt należy do danej grupy, nie ma pewności, że taki neuron będzie miał wartość w przedziale (0, 1). b) W przypadku sieci składającej się z bardzo dużej ilości warstw, w przypadku liniowych funkcji aktywacji neuronów w każdej warstwie może dojść do zjawiska eksplodujących gradientów – wartość wyniku ostatniej warstwy może być bardzo duża i nie dająca możliwości wnioskowania z niej bądź nawet nie mieszcząca się w pamięci komputera.<sup>94</sup>

W związku z powyższym w praktyce częściej stosuje się neurony nieliniowe, gdzie wartość sumy jest argumentem funkcji spłaszczającej (normalizującej) wynik, który w takiej formie jest przekazywany do kolejnej warstwy sieci neuronowej. Oznaczając  $x_1w_1 + x_2w_2 + \dots + x_nw_n$  przez  $\Sigma$ , funkcja będzie miała postać:

$$y = \varphi(\Sigma) \quad (2.9)$$

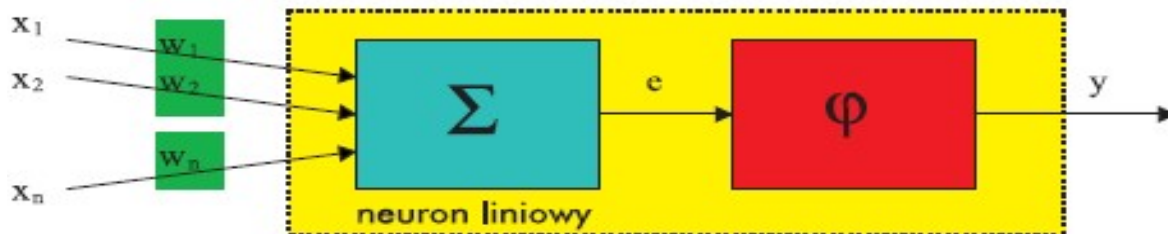
gdzie  $\varphi$  to jedna z funkcji aktywacji (przejścia).<sup>95</sup> Schemat obliczeniowy dla pojedynczego neuronu tego typu obrazuje rysunek 15. Takie same obliczenia wykonywane są dla każdego neuronu w każdej warstwie sieci neuronowej. Jeśli zadaniem danego wielowarstwowego perceptronu jest regresja, ostatni neuron (neuron wyjściowy) jest najczęściej liniowy, bez dodatkowej funkcji aktywacji, ponieważ ma on zaprezentować rzeczywistą wartość, którą badacz chce zaprognozować, a nie jej zniekształcony funkcją aktywacji wariant.

---

<sup>94</sup> Géron, Op. Cit, s. 253 - 264

<sup>95</sup> Ibidem.

Rys. 15. Struktura neuronu nieliniowego jako rozszerzenie neuronu liniowego o nieliniową funkcję przejścia.



Źródło: Tadeusiewicz, R., 2007, *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#*, Polska Akademia Umiejętności, Kraków, s. 198.

Zadaniem funkcji aktywacji jest unormowanie wyników wszystkich pojedynczych neuronów. Większość funkcji aktywacji przypisuje zmiennym wartości z niewielkiego zakresu (na przykład (0,1) lub (-1,1), co pozwala ograniczyć wzrost wartości dla kolejnych warstw neuronów. Funkcje aktywacji są również istotne w związku z propagacją wsteczną (optymalizacją funkcji bazowej), ponieważ są różniczkowalne w całej swojej dziedzinie (a ich pochodne zazwyczaj nie są zerem). Optymalizacja za pomocą metody gradientu prostego wymaga funkcji, z której w każdym miejscu można wyliczyć niezerową pochodną, ponieważ w przypadku zerowej (a więc braku zmiany wartości funkcji bazowej przy zmianie danego parametru) gradient nie będzie jednoznaczny w kwestii zwiększenia bądź zmniejszenia wagi dla danego parametru.<sup>96</sup>

Klasyczną funkcją aktywacji była funkcja sigmoidalna o postaci:

$$S(x) = \frac{e^x}{e^x + 1} \quad (2.10)$$

której zbiór wartości mieści się w przedziale (0,1) dla całej dziedziny. Obecnie używa się całej rodziny funkcji, a wybór funkcji aktywacji jest związany z zagadnieniem, do którego sieć jest zaaplikowana (oraz z eksperymentami metodą prób i błędów). Funkcje aktywacji często są prostymi transformacjami, używanymi, by w całej sieci neuronowej osiągnąć wysoce nieliniowe przekształcenia. Do obecnie używanych funkcji należą między innymi tanh:

$$T(x) = \tanh(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}} \quad (2.11)$$

z wartościami w przedziale (-1, 1) dla całej dziedziny; ReLu (*Rectified Linear Unit*):

<sup>96</sup> Goodfellow i in., Op. Cit., s. 158 - 164

$$R(x) = \max(0, x) \quad (2.12)$$

która jest wyjątkiem, ponieważ nie ma ograniczonej górnej wartości oraz nie jest różniczkowalna w punkcie zero. Natomiast jest łatwa dla obliczania i praktyka pokazuje, że dobrze sprawdza się zwłaszcza w przypadku głębokich sieci neuronowych; *Softmax*:

$$\text{Soft}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \text{ dla } i = 1, \dots, K \text{ oraz } x = (x_1, \dots, x_K) \quad (2.13)$$

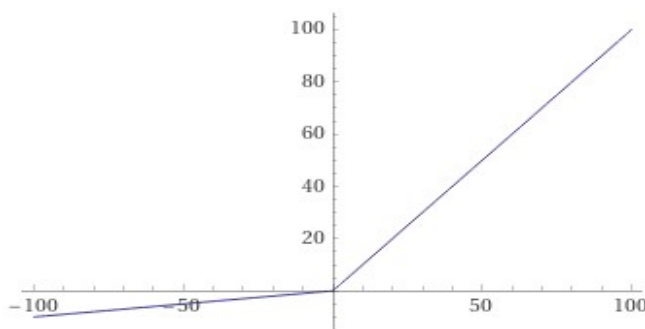
używany przede wszystkim w zadaniach klasyfikacyjnych, ponieważ jej wartości zawierają się w przedziale (0,1) dla każdej z n klas a suma tych wartości to 1. Wartość funkcji dla konkretnej klasy k oznacza prawdopodobieństwo z jakim dany element należy do klasy k.<sup>97</sup>

Jedną z niedawno zaaplikowanych funkcji aktywacji (2015) o dużej skuteczności, jest *LeakyReLU*, czyli wariant *ReLU*, w którym:

$$RL(x) = \max(\alpha x, x) \quad (2.14)$$

Wariant podstawowy tej funkcji aktywacji borykał się z problemem umierających neuronów, czyli takich, które nie dają wyników innych niż zero. Problem ten był szczególnie zauważalny w sieciach z dużym współczynnikiem uczenia. Jeśli w trakcie uczenia wagi zostaną zaktualizowane w taki sposób, że suma mnożenia wag z danymi wejściowymi neuronu jest negatywna, będzie to skutkowało neuronem dającym wynik w postaci zera. Wariant „cieknący” nadaje parametr  $\alpha$  (domyślna jego wartość to 0,01), który zabezpiecza neurony przed wymieraniem.<sup>98</sup>

Rys. 16. Wykres funkcji LeakyReLU dla dziedziny (-100, 100) i parametru  $\alpha=0,1$ .



Źródło: Opracowanie własne w programie WolframAlpha. Na podstawie: Géron, A., 2017, Hands-On Machine Learning with Scikit-Learn and Tensorflow, O'Reilly Media, Sebastopol, s. 280.

<sup>97</sup> Ibidem.

<sup>98</sup> Géron, Op. Cit, s. 279.

Istnieje wiele funkcji aktywacji i z każdym rokiem wykorzystuje się ich coraz więcej – badacze nieustannie sprawdzają wyniki sieci neuronowych przy wykorzystaniu różnych funkcji aktywacji. W tym rozdziale zostały omówione najbardziej podstawowe z nich lub takie, które są powszechnie stosowane.

Kryteria treningowe dla sieci neuronowych są bazowane albo na maksymalizowaniu prawdopodobieństwa poprawnej klasyfikacji albo na minimalizacji błędu szacunku liczbowego (w przypadku regresji). Do tego celu dobrze nadają się funkcje starty opisane we wcześniejszej części tego rozdziału, takie jak MSE dla regresji czy funkcja wiarygodności (dla klasyfikacji).<sup>99</sup>

Uczenie sieci neuronowych zawsze jest procesem iteracyjnym. Generalna zasada mówi, że im więcej iteracji, tym lepiej (zwłaszcza w przypadku dużych zbiorów danych), ale w przypadku bardziej zawiłych problemów istnieje moment, po którym zmniejsza się wartość funkcji straty mierzona tylko dla zbioru treningowego, natomiast wartość tej samej funkcji dla zbioru testowego rośnie (nadmierne dopasowanie). Wybór procedury optymalizacyjnej dla sieci neuronowej zależy w pewnej mierze od zadania, które ma wykonywać, ale w praktyce większość sieci jednokierunkowych uczy się przy zastosowaniu metody gradientu prostego.<sup>100</sup>

W bardziej zawiłych sieciach neuronowych, projektant sieci nie ma bezpośredniego wpływu na neurony w warstwach ukrytych – nie da się bezpośrednio wyznaczyć wielkości błędów dla neuronów w tych warstwach, ponieważ nauczyciel sieci podaje poprawne wartości tylko dla warstwy wyjściowej, natomiast sygnałów cząstkowych (wyników neuronów z warstw ukrytych) nie ma do czego porównać. Najszerzej stosowanym rozwiązaniem, pozwalającym poradzić sobie z tym problemem jest zastosowanie wstecznej propagacji błędów (*backpropagation*). Rozwiązanie to polega na odtworzeniu prawdopodobnych wartości błędów głębszych warstw sieci neuronowej za pomocą rzutowania wstecz błędów z warstwy wyjściowej. Rozważając każdy pojedynczy neuron warstw ukrytych, bierze się pod uwagę błędy wszystkich neuronów, do których sprawdzany neuron przesłał swój wynik cząstkowy i sumuje się je, biorąc również pod uwagę współczynniki wag między badanym neuronem, a tymi które korzystają z jego wyniku (im większa waga, tym bardziej wynik sprawdzanego neuronu wpływał na kolejny neuron).<sup>101</sup>

---

<sup>99</sup> Goodfellow i in., Op. Cit., s. 173 - 174

<sup>100</sup> Ibidem

<sup>101</sup> Tadeusiewicz, 2007, Op. Cit., s. 132.



Wsteczna propagacja błędu polega na różniczkowaniu kryterium treningowego  $J$  (który jest wartością skalarną) względem zestawu parametrów wag (wektora  $\theta$ ). Pochodna cząstkowa  $J$  względem  $\theta$  jest wyznacznikiem tego, jak  $\theta$  powinno być zwiększone lub zmniejszone, by zmniejszyć  $J$ . Propagacja wsteczna pomaga osiągnąć ten cel przy optymalnej zawłości obliczeniowej (żaden znany algorytm nie jest w stanie obliczyć gradientu szybciej). Pochodną cząstkową  $J$  względem  $\theta$  można poddawać rekursywnej dekompozycji biorąc pod uwagę kompozycje funkcji wiążących  $\theta$  z  $J$  poprzez wartości pośredniczące w przekazywaniu tych zależności. Podstawowym matematycznym narzędziem pozwalającym na badanie wpływu pochodnych jednej funkcji na drugą jest reguła łańcuchowa. Pochodna cząstkowa  $\frac{\partial y}{\partial x}$  sprawdza lokalny, liniowy wpływ zmiennej  $x$  na  $y$ . Jeśli  $x$  wpływa na  $y$ , które wpływa na  $z$ , należy zbadać, jak mała zmiana  $x$  przyczyni się do małej zmiany w  $z$  przez zmianę  $y$ . W przypadku sieci neuronowych, wynik  $z$  to funkcja straty:

$$z = J(g(\theta)) \quad (2.15)$$

a bada się gradient względem parametrów  $x = \theta$ . W przekazywaniu tego wpływu pośredniczą  $y = g(\theta)$ , takie jak wagi neuronów. Można zapisać to za pomocą równania, które wyznacza wpływ zmiany parametrów  $\theta$  na funkcję  $J$ :

$$\nabla_{\theta} J(g(\theta)) = \nabla_{g(\theta)} J(g(\theta)) \frac{\partial g(\theta)}{\partial \theta} \quad (2.16)$$

Równanie to jest prawdziwe również gdy  $J$ ,  $g$  lub  $\theta$  są wektorami a nie skalarami (w takim wypadku pochodne cząstkowe są macierzami Jacobiego odpowiedniej wymiarowości). Rozpatrując to równanie w wersji skalarnej, można przeprowadzić interpretację, że mała zmiana w  $\theta$  wpłynie na małą zmianę w  $g(\theta)$  poprzez przemnożenie przez  $\frac{\partial g(\theta)}{\partial \theta}$ , a mała zmiana w  $g(\theta)$  wpłynie na małą zmianę w  $J(g(\theta))$  poprzez przemnożenie przez  $\nabla_{g(\theta)} J(g(\theta))$ , w związku z tym stosunek zmiany  $J(g(\theta))$  do zmiany  $\theta$  jest wynikiem powyżej opisanych pochodnych cząstkowych.<sup>102</sup>

Innym, ważnym dla sieci neuronowych zagadnieniem jest randomizacja procesu uczenia – liczba obserwacji na podstawie których sieć neuronowa ma się uczyć najczęściej jest

---

<sup>102</sup> Goodfellow i in., Op. Cit., s. 174 – 176.

niewystarczająca dla osiągnięcia dobrych wyników przy jednorazowym wykorzystaniu tych danych. Najczęściej wszystkie obiekty prezentowane są kilku lub kilkadziesiątkrotnie. Niekorzystnym byłoby pokazywanie sieci danych treningowych w konkretnej sekwencji, ponieważ okresowe nauczanie może doprowadzić do pojawiania się cykli współczynników wagowych, a co za tym idzie, złego dopasowania sieci do nowych danych. Dostarczanie danych do sieci w losowej (lub semilosowej) kolejności pozwala pozbyć się tego problemu.<sup>103</sup>

Podczas procesu nauki głębokiej sieci neuronowej ważne jest normowanie wyników cząstkowych pojedynczych neuronów lub danych warstw. Efekt ten częściowo osiąga się poprzez funkcję aktywacji w danym neuronie, ale powszechną praktyką jest stosowanie także dodatkowych metod. Wiele z nich ma swoje zastosowanie również w klasycznych metodach uczenia maszynowego. Skuteczne normowanie wyników to zyskowna wymiana między obciążeniem a wariancją (zwiększenie pierwszego dla zmniejszenia drugiego). Wśród przykładów często stosowanych funkcji normujących wyniki wymienia się między innymi: **a) Rozkładem wag (*weight decay*)**, **b) Dodawanie danych wejściowych z szumem**, **c) Wczesne zatrzymanie (*Early Stopping*)**, **d) Dropout**. **Rozkład wag** polega ona na dodaniu parametru zmieniającego wynik funkcji straty, który zmusza algorytm uczący do optymalizacji funkcji straty przy zachowaniu jak najniższych wartości wag. **Dodawanie danych wejściowych z szumem** to metoda, którą stosuje się chcąc uodpornić sieć neuronową na słabej jakości dane wejściowe w fazie testowej. Można próbować dodać gorszej jakości dane wejściowe również w fazie uczenia, dzięki czemu sieć powinna nauczyć wnioskować z danych o gorszej jakości. Przykładem może być zdjęcie, na którym kot (zwierzę, które sieć ma rozpoznawać) stoi w cieniu. **Wczesne zatrzymanie (*Early Stopping*)** polega na przerwaniu uczenia w  $i$ -tej iteracji, w przypadku gdy minimalizowana funkcja straty nie zmalała od  $n$  okresów. **Dropout** to jedna z najczęściej używanych technik normowania – w każdej kolejnej iteracji uczenia każdy neuron (łącznie z neuronami wejściowymi, ale bez neuronów wyjściowych) z prawdopodobieństwem  $p$  nie będzie brany pod uwagę w tej iteracji. Ten zabieg pomaga osiągnąć lepszy ostateczny wynik, ponieważ neurony stają się bardziej odporne na niewielkie zmiany w danych wejściowych. W ten sposób uodparniają się na błędy i na zaszumione dane bardziej.<sup>104</sup>

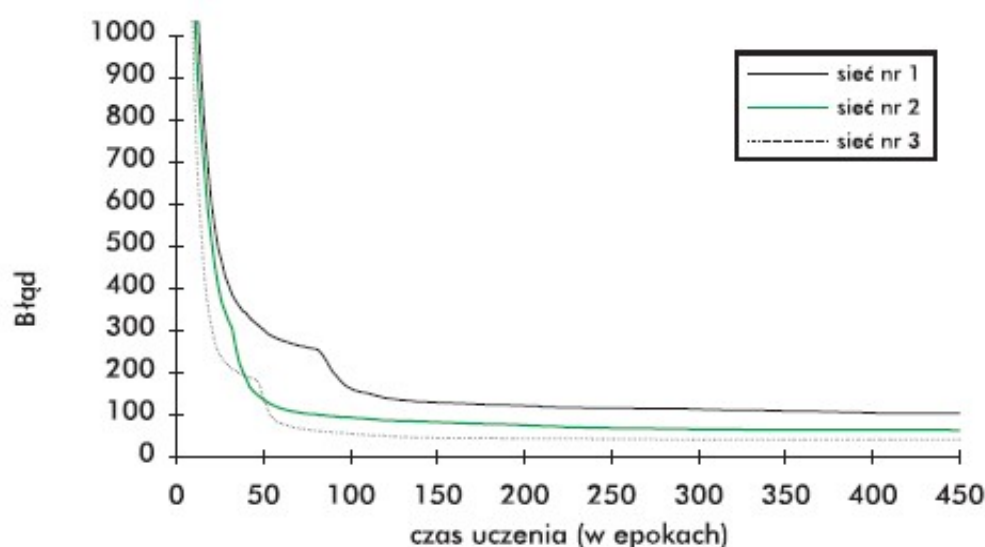
---

<sup>103</sup> Tadeusiewicz, 2007, Op. Cit., s. 133.

<sup>104</sup> Goodfellow i in., Op. Cit., s. 196 – 235.

Uczenie się sieci neuronowej jest procesem stochastycznym i wyniki identycznej sieci nauczanej wielokrotnie bez sztucznego blokowania losowości (*random seed generator*) będzie owocowało różnymi wynikami. Wagi w sieciach neuronowych często inicjalizuje się losowo (nauczyciel nie wie, który zestaw wag będzie najlepszym punktem startowym), a w związku ze sposobem działania metody gradientu prostego, różną długość czasu może zająć sieci dojście do tych samych wyników. Nierzadko sieci z odmiennymi wagami nie dochodzą ostatecznie do tego samego wyniku, choć przebieg uczenia jest podobny (stopniowe zmniejszanie się błędu wraz z kolejnymi epokami, które najszybsze jest w pierwszych kilkudziesięciu epokach, a potem znacznie zwalnia). Epoką w uczeniu nazywa się jedną iterację uczenia, w trakcie której sieć uczy się ze wszystkich danych wejściowych, przy czym aktualizacje wag mogą i najczęściej odbywają się więcej niż raz na epokę – zależnie od wielkości paczki danych (*mini-batch gradient descent*).

Rys. 17. Zmniejszanie się błędu w trakcie procesu uczenia dla różnych wartości początkowych wag w badanym modelu.



Źródło: Tadeusiewicz, R., 2007, *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#*, Polska Akademia Umiejętności, Kraków, s. 176

### 2.2.2. Rekurencyjne sieci neuronowe

Rekurencyjne sieci neuronowe (*RNN*) są typem sieci szczególnie popularnym w przewidywaniu przyszłych wartości. Ten rodzaj architektury wykorzystuje się między innymi w analizie szeregów czasowych takich jak kursy akcji (na podstawie przeszłych danych mogą wygenerować predykcję dla następnego dnia sesyjnego), autonomicznych systemach prowadzenia

pojazdów (gdzie przewidują trajektorię obiektów i pomagają unikać kolizji), przetwarzaniu języka naturalnego (na przykład przy tłumaczeniu maszynowym) czy analizie ładunku emocjonalnego tekstu (może pomóc ocenić, czy ocena opisowa filmu jest pozytywna lub w innym kontekście – sprawdzić, czy nagłówki prasowe z danego dnia są korzystne dla wyceny wybranego waloru giełdowego). Predykcyjne zdolności tego rodzaju sieci neuronowych sprawiają, że wykorzystuje się je z sukcesem również w zadaniach twórczych i są one w stanie na przykład wykreować obraz (uprzednio nauczony z bazy obrazów) poprzez przewidzenie, jaki rodzaj piksela powinien najprawdopodobniej wystąpić po wcześniej wprowadzonym pikselu.<sup>105</sup>

Sieci rekurencyjne działają z danymi wejściowymi ułożonymi w sekwencje różnej długości (czym różnią się od klasycznych sieci neuronowych czy sieci konwolucyjnych, gdzie dane wejściowe muszą być konkretnej długości). Omawiane w tym rozdziale sieci mogą przyjmować dane wejściowe w postaci na przykład całych zdań lub fragmentów nagrań dźwiękowych stanowiących jakąś całość.<sup>106</sup>

Jednymi z największych wyzwań przy projektowaniu rekurencyjnej sieci są kwestie eksplodujących lub znikających gradientów. Oprócz bazowej wersji *RNN* występuje także kilka jej wariantów – między innymi *LSTM* czy *GRU*.<sup>107</sup> Architektura neuronu typu *LSTM* jest szczególnie pomocna przy pracy nad danymi finansowymi, więc zostanie omówiona szczegółowo w dalszej części tego podrozdziału.

Kolejnym aspektem, w którym sieci *RNN* różnią się od perceptronu wielowarstwowego jest kierunek przepływu informacji. W przypadku wcześniej omawianego perceptronu, przepływ informacji jest jednokierunkowy, natomiast sieć *RNN* poza połączeniami neuronowymi i wynikami cząstkowymi znanymi z klasycznych sieci neuronowej, ma również połączenia skierowane wstecz. W najprostszym przypadku jednoneuronowej sieci rekurencyjnej będzie to oznaczało, że na każdym kroku czasowym  $t$  (zwanym również klatką) taki neuron będzie otrzymywał dane wejściowe  $x_{(t)}$  oraz swój własny wynik cząstkowy z poprzedniego kroku czasowego:  $y_{(t-1)}$ . Jeśli taki neuron zostanie rozwinęty w czasie, będzie miał postać przedstawioną na rysunku 18.<sup>108</sup>

---

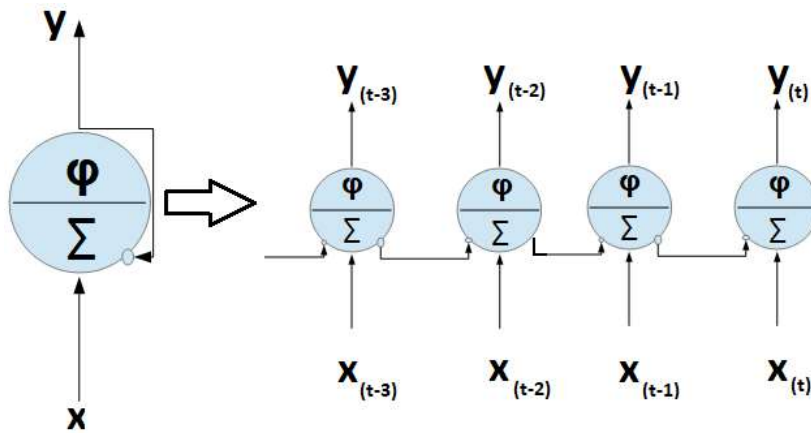
<sup>105</sup> Géron, Op. Cit, s. 379 - 381

<sup>106</sup> Ibidem.

<sup>107</sup> Ibidem.

<sup>108</sup> Ibidem.

Rys. 18. Schemat działania jednoneuronowej sieci rekurencyjnej rozłożonej w czasie.



Źródło: Opracowanie własne. Na podstawie Géron, A., 2017, Hands-On Machine Learning with Scikit-Learn and Tensorflow, O'Reilly Media, Sebastopol, s. 380.

W ten sam sposób można układać całe warstwy neuronów rekurencyjnych – w takim wypadku zamiast skalnych danych wejściowych i wyjściowych występują wektory. Każdy neuron rekurencyjny ma dwa zestawy wag – jeden dla danych wejściowych  $x$  dla kroku  $t$ , a drugi dla danych wyjściowych  $y$  z kroku  $t-1$ . Nazywając zestawy wag odpowiednio:  $w_x$  oraz  $w_y$  wynik obliczeń zachodzących w pojedynczym neuronie rekurencyjnym można zapisać za pomocą wzoru:

$$y(t) = \varphi(x(t)^T * w_x + y(t-1)^T * w_y + b) \quad (2.17)$$

gdzie  $b$  oznacza obciążenie modelu, a  $\varphi$  to funkcja aktywacji, co w wersji wektorowej będzie miało następującą postać:

$$Y(t) = \varphi([X(t) \ Y(t-1)] * W + b) \text{ dla } W = \begin{bmatrix} W_x \\ W_y \end{bmatrix} \quad (2.18)$$

gdzie  $Y(t)$  to macierz  $m \times n_{neuro\text{ny}}$  zawierająca dane wyjściowe warstwy w kroku czasowym  $t$ , a  $m$  to liczba obserwacji w mini-paczki (*mini-batch*) danych, a  $n_{neuro\text{ny}}$  to liczba neuronów w warstwie.  $X(t)$  to macierz  $m \times n_{wejs\text{ciowe}}$  zawierająca dane wejściowe wszystkich obserwacji, gdzie  $n_{wejs\text{ciowe}}$  to liczba atrybutów.  $W_x$  to macierz  $n_{wejs\text{ciowe}} \times n_{neuro\text{ny}}$  zawierająca wagi połączeń dla danych wejściowych danego kroku.  $W_y$  to macierz  $n_{neuro\text{ny}} \times n_{wejs\text{ciowe}}$  zawierająca wagi połączeń dla danych wyjściowych neuronów z poprzedniego kroku czasowego. Macierze wag najczęściej przedstawia się przy użyciu jednej macierzy  $W$  o kształcie  $(n_{wejs\text{ciowe}} +$

$n_{neuron}$ )  $\times n_{neuron}$  . Natomiast  $b$  to wektor o rozmiarze  $n_{neuron}$  zawierający obciążenia wszystkich neuronów.<sup>109</sup>

W związku z tym, że neuron sieci rekurencyjnej w kroku czasowym  $t$  jest funkcją wszystkich danych wejściowych z poprzednich kroków czasowych, można powiedzieć, że ma on swego rodzaju pamięć. Część sieci rekurencyjnej przechowuje stan pomiędzy kolejnymi krokami czasowymi. W przypadku niezawitych sieci, stan  $h$  może być równy danym wyjściowym  $y$  danego kroku czasowego, ale w bardziej rozbudowanych przypadkach jest to osobna zmienna.<sup>110</sup>

Sieci RNN mogą jednocześnie przyjmować sekwencyjne dane wejściowe i generować sekwencyjne dane wyjściowe. Z tego powodu często wykorzystywane są w analizowaniu danych finansowych opartych na szeregach czasowych – na przykład przyjmują dane dotyczące ceny akcji na giełdzie z ostatnich  $N$  dni, a danymi wyjściowymi są predykcje dotyczące kursu akcji w ciągu następnych  $N$  dni. Innym rozwiązaniem jest załadowanie do sieci wartości akcji z ostatnich  $N$  dni, a w danych wyjściowych uwzględnienie tylko przewidywanej wartości dla kolejnego dnia sesyjnego (sieć typu sekwencja do wektora). Jeszcze innym typem sieci rekurencyjnej jest sieć typu wektor do sekwencji, nazywany również dekodery. Sieci tego typu są często wykorzystywane się w tłumaczeniu maszynowym.<sup>111</sup>

Kształt danych w sieci RNN zależy od ich typu – analizując szereg czasowy z wieloma zmiennymi, danymi wejściowymi sieci mogą być zmienne  $x_1 \dots x_n$  dla danego kroku czasowego  $t$  (w takim wypadku dane wejściowe są dwuwymiarowe: liczba kroków czasowych razy liczba zmiennych  $x$ ) lub okienka czasowe, które składają się z ustalonej liczby kroków czasowych (w takim wypadku dane wejściowe są trójwymiarowe: liczba okienek czasowych razy liczba kroków czasowych w okienku razy liczba zmiennych dla jednego kroku czasowego).<sup>112</sup>

W związku z tym, że sieć rekurencyjna może przyjmować dane wejściowe o różnym kształcie, istnieje podział na sieci statyczne i dynamiczne. W pierwszym z typów deklaruje się długość sekwencji danych wejściowych i sieć będzie przyjmowała tylko dane o tej długości – może to być przydatne, gdy badacz analizuje dane z szeregu czasowego i chce sprawdzić jak dane z  $n$

---

<sup>109</sup> Ibidem.

<sup>110</sup> Géron, Op. Cit, s. 382.

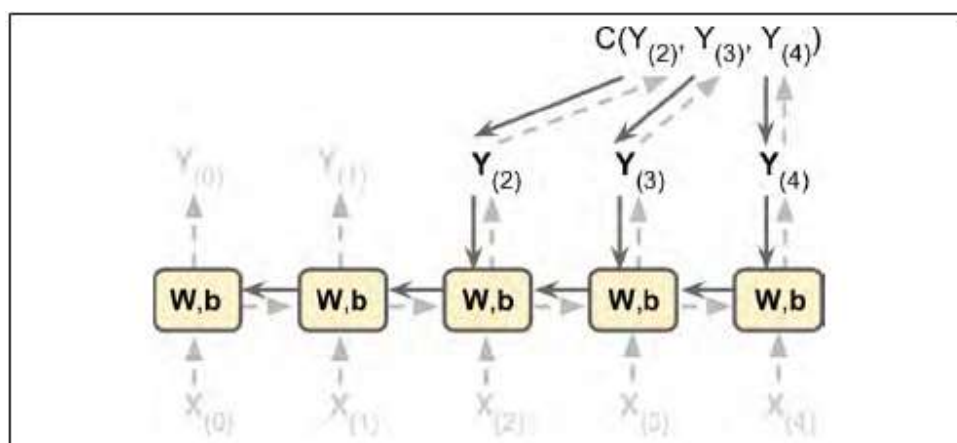
<sup>111</sup> Géron, Op. Cit, s. 382 - 383

<sup>112</sup> Goodfellow i in., Op. Cit., s. 308 - 313

ostatnich dni wpływają na wynik w dniu  $n + 1$ . W innych przypadkach, takich jak tłumaczenie maszynowe, koniecznym jest używanie dynamicznej sieci rekurencyjnej, jako iż zdania w każdym języku przyjmują różne długości i sieć musi być w stanie przyjmować sekwencje o różnych kształtach.<sup>113</sup>

Sieci rekurencyjne, tak jak inne typy sieci neuronowych, również mogą być optymalizowane poprzez zastosowanie metod opartych na gradientach, a aktualizacji wag dokonuje się przez propagację wsteczną w czasie (*backpropagation through time*). Tak samo jak w przypadku zwykłej propagacji wstecznej, dane najpierw przechodzą od wejścia do wyjścia poprzez rozwiniętą (w czasie) sieć, następnie sekwencja wynikowa jest weryfikowana w odniesieniu do funkcji straty (dla wyniku z każdego kroku czasowego), a później gradienty obliczone przy użyciu funkcji straty są przesyłane wstecz przez rozwiniętą w czasie sieć, a parametry modelu są zaktualizowane przy ich użyciu.<sup>114</sup>

Rys. 19. Schemat działania sieci rekurencyjnej z 5 krokami czasowymi dla danych wejściowych i 3 krokami w sekwencji wyjściowej.



Źródło: Géron, A., 2017, Hands-On Machine Learning with Scikit-Learn and Tensorflow, O'Reilly Media, Sebastopol, s. 389.

Istotnymi problemami pojawiającymi się przy projektowaniu głębokiej sieci rekurencyjnej są:

<sup>113</sup> <http://www.wildml.com/2016/08/rnns-in-tensorflow-a-practical-guide-and-undocumented-features/>, 04.01.2021

<sup>114</sup> Goodfellow i in., Op. Cit., s. 314 - 315

- nadmierne dopasowanie danych do zbioru treningowego, które można ograniczyć poprzez zastosowanie *dropoutu*. *Dropout* to prawdopodobieństwo (dla części lub wszystkich neuronów w sieci, poza warstwą wyjściową), że neuron nie zostanie aktywowany w danej iteracji uczenia.
- problem znikających/eksplodujących gradientów pojawia się przy trenowaniu sieci rekurencyjnej przy użyciu długich okienek czasowych, co można ograniczyć na przykład przez normalizację danych wejściowych lub wykorzystanie szybszych algorytmów optymalizacyjnych. Natomiast w przypadku zbyt długich sekwencji, sieć nadal może borykać się z tym samym problemem i/lub uczyć się zbyt wolno. W takim wypadku badacz powinien skrócić długość sekwencji i ponowić proces uczenia dla zmienionych danych wejściowych. Minusem takiego rozwiązania jest brak możliwości uchwycenia długoterminowych trendów w danych.
- w przypadku długich sekwencji tradycyjne sieci rekurencyjne stopniowo tracą pamięć na temat informacji zawartych w pierwszych danych wejściowych – z każdym kolejnym krokiem dane ulegają transformacjom, które sprawiają, że ślad wpływu pierwszych danych traci znaczenie. Problem ten jest szczególnie istotny, jeśli najważniejsze dla sieci dane zawierają się w pierwszych kilku danych wejściowych, a sekwencja jest zbyt długa, żeby ślad tych informacji został przekazany do dalszych kroków czasowych.<sup>115</sup>

### 2.2.3. Sieci LSTM

Sieci rekurencyjne są powszechnie stosowane w obecnych pracach wykorzystujących sieci neuronowe i niejednokrotnie osiągają lepsze wyniki niż inne typy architektury. Z definicji powinny one być zdolne do przechowywania danych z wcześniejszych kroków czasowych i wykorzystywania ich w analizie późniejszych. Jak zostało opisane w poprzednim rozdziale – przy dłuższych sekwencjach (okienkach czasowych) dane z pierwszych kroków mogą być w zbyt dużej mierze utracone, by realnie wpływać na ostatnie z nich. Prowadzone są nieustanne prace nad usprawnianiem istniejących już modeli – jednym z nich było opisanie i wykorzystanie w 2015 roku neuronów LSTM (*Long Short-Term Memory*). Nazwę tę można przetłumaczyć jako długotrwała pamięć krótkotrwała.<sup>116</sup>

Dzięki swojej budowie sieci z neuronami LSTM są w stanie przechowywać długoterminowe zależności. Domyślnie wszystkie rekurencyjne sieci neuronowe mają formę

---

<sup>115</sup> Géron, Op. Cit, s. 399-400

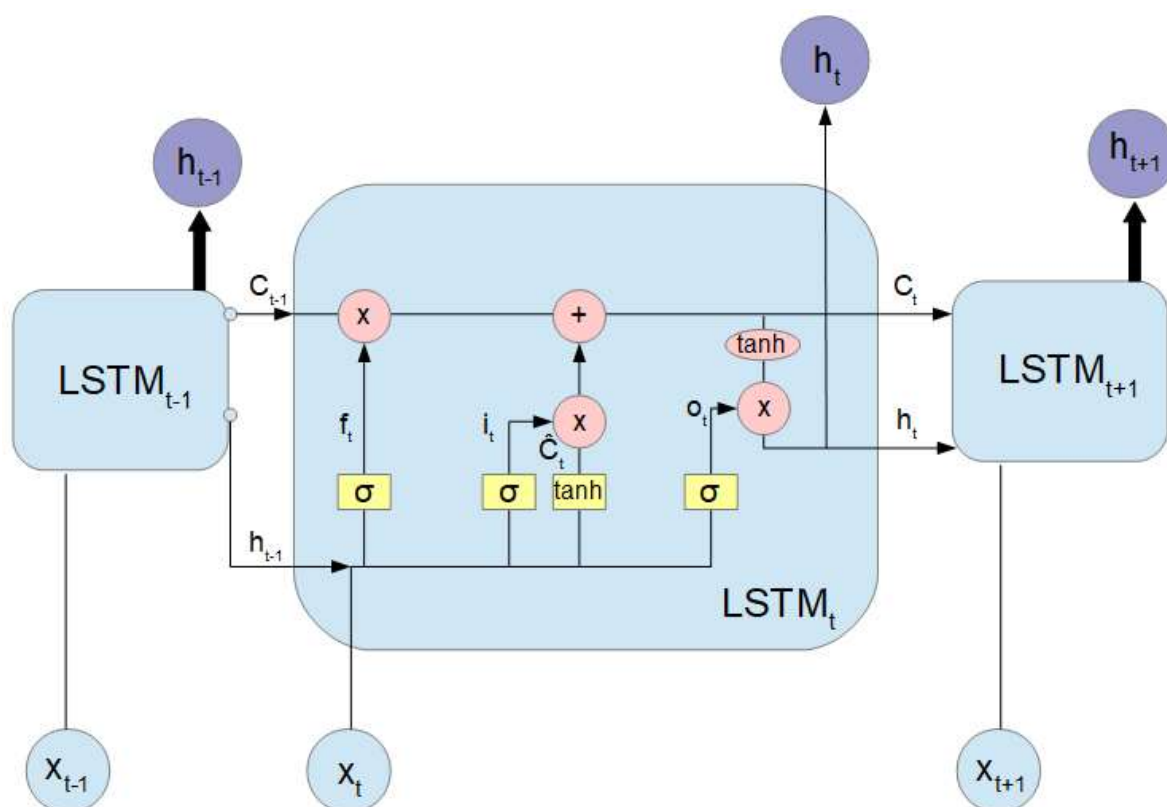
<sup>116</sup> Géron, Op. Cit, s. 401 - 403



łańcucha powtarzających się modułów sieci neuronowej. W standardowych sieciach RNN ten powtarzający się moduł ma prostą budowę (w postaci na przykład pojedynczej warstwy z funkcją aktywacji  $\tanh$ ). Sieci LSTM również mają budowę łańcuchową, ale powtarzający się moduł zamiast pojedynczej warstwy sieci neuronowej, ma ich cztery.<sup>117</sup>

Elementem odróżniającym sieć LSTM od standardowej RNN jest *cell state* (stan komórki), czyli pamięć długotrwała modelu, którą można wyobrazić sobie jako poziomą linię, biegnącą przez wszystkie neurony (rozwinęte w czasie) sieci LSTM. Każdy neuron zmienia jej wartość w zależności od istotności przechowywanych danych – czyli decyduje jaka część przekazanego stanu ma zostać „zapomniana”, a jaka część nowych danych ma zostać przekazana do kolejnych neuronów jako stan komórki.<sup>118</sup>

Rys. 20. Schemat neuronu LSTM.



Źródło: Opracowanie własne w programie OpenOffice Draw.

<sup>117</sup> <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, 06.01.2021

<sup>118</sup> Ibidem.

Rysunek 20. przedstawia schemat działania pojedynczego neuronu LSTM – pozioma linia na górze to stan komórki, który zmieniany jest w jednej komórce dwa razy (operacjami liniowymi) – najpierw stan przekazany z poprzedniej komórki jest przemnażany przez wynik funkcji z bramki zapominającej (*forget gate*), a później dodawany jest do niego wynik funkcji z bramki danych wejściowych (*input gate*). Pierwszym krokiem w przepływie danych przez LSTM jest bramka zapominająca – na podstawie pamięci krótkotrwałej z poprzedniego neuronu ( $h_{t-1}$ ) oraz danych wejściowych dla bieżącego neuronu ( $x_t$ ) sieć neuronowa „decyduje” za pomocą warstwy sigmoidalnej (bramki zapominającej, która sama w sobie jest siecią neuronową) jaka część stanu komórki powinna zostać zapomniana. Warstwa sigomoidalna daje wynik pomiędzy 0 a 1, gdzie 0 oznacza całkowite usunięcie obecnego stanu komórki, a 1 oznacza brak usuwania czegokolwiek z obecnego stanu komórki. Funkcja, która kontroluje proces zapominania ma postać: <sup>119</sup>

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.19)$$

W kolejnym kroku komórka LSTM decyduje jaka część nowych informacji (z pamięci krótkotrwałej oraz danych wejściowych dla danego kroku) zostanie zapamiętana w pamięci długotrwałej (stanie komórki). Ten krok podzielony jest na etapy: a) po pierwsze warstwa sigmoidalna o nazwie bramka danych wejściowych (*input gate layer*) decyduje, które wartości zostaną zaktualizowane, b) a następnie warstwa z funkcją tanh tworzy wektor z wartościami kandydatami ( $\hat{C}_t$ ), które po przemnożeniu przez wynik bramki sigmoidalnej zostaną dodane do obecnego stanu komórki. Zarówno warstwa sigmoidalna jak i tanh to osobne sieci neuronowe. Funkcje, które kontrolują te procesy mają postać:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2.20)$$

$$\hat{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)^{120} \quad (2.21)$$

Ostatnim krokiem jest obliczenie danych wyjściowych neuronu LSTM i tym samym pamięci krótkotrwałej modelu. Na wynik komórki mają wpływ: pamięć krótkotrwała z poprzedniego kroku ( $h_{t-1}$ ), dane wejściowe z obecnego kroku ( $x_t$ ) oraz zaktualizowany w poprzednich etapach stan komórki ( $C_t$ ). Najpierw  $x_t$  i  $h_{t-1}$  przechodzą przez warstwę sigmoidalną, która decyduje o tym, jaka część tych danych znajdzie się w wyniku (po przemnożeniu przez wagi),

---

<sup>119</sup> Ibidem.

<sup>120</sup> Ibidem.

następnie  $C_t$  przechodzi przez funkcję  $\tanh$  (która tutaj nie jest osobną warstwą sieci neuronowej, a jedynie operacją matematyczną, która standaryzuje wynik sprawiając, że mieści się on w przedziale  $(-1, 1)$ ), a na końcu wyniki tych dwóch operacji są przez siebie przemnażane. Ostateczny wynik staje się danymi wyjściowymi neuronu oraz pamięcią krótkotrwałą sieci, która przekazywana jest do kolejnego neuronu  $LSTM_{t+1}$  jako  $h_t$ . Funkcje, które kontrolują te procesy mają postać:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.22)$$

$$h_t = o_t \otimes \tanh(C_t)^{121} \quad (2.23)$$

Minusem sieci LSTM jest ich większe zapotrzebowanie na moc obliczeniową (niż w przypadku standardowych sieci rekurencyjnych), jako że w pojedynczym neuronie znajduje się kilka innych sieci neuronowych, uczących się, które dane są istotne i należy je zachować w pamięci na dłużej, a które nie.<sup>122</sup>

### 2.3. Przykłady zastosowania sieci neuronowych w finansach i ekonomii

W tym rozdziale zostanie zaprezentowany krótki przegląd prac, które opisują wykorzystanie sieci neuronowych w finansach i ekonomii, ze szczególnym uwzględnieniem tych, które zajmują się prognozowaniem wartości walorów na rynku kapitałowym.

Adya i Collopy (1998) przeprowadzili analizę 48 prac naukowych, w których porównywano trafność przewidywania wartości sieci neuronowych z trafnością przewidywania innych, bardziej konwencjonalnych metod. Odrzucono badania, w których przyjęto niewłaściwe założenia lub niemożliwym było ich powtórzenie w celu weryfikacji skuteczności – w ten sposób zredukowano liczbę prac do porównania do 22. W 19 z nich (86%) sieci neuronowe osiągnęły lepsze rezultaty w prognozowaniu przyszłych wartości niż metody alternatywne.<sup>123</sup> Należy dodać, że porównywane prace zostały opublikowane w latach 1988 – 1994, a od tamtego czasu w dziedzinie sieci neuronowych dokonano wielu przełomowych odkryć.

---

<sup>121</sup> Ibidem.

<sup>122</sup> Ibidem.

<sup>123</sup> Adya, M., Collopy, F., 1998, How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting*, 17(5-6), 481–495.

Li i Ma (2010) zestawiają kilkadziesiąt prac naukowych o tematyce ekonomicznej i finansowej, w których używa się sieci neuronowych. Zauważają, że sieci neuronowe są wolne od założeń statystycznych i bardziej odporne na brakujące oraz błędne dane. Dzięki iteracyjnemu nauczaniu i zdolności do oszacowania funkcji opisujących nawet wielowymiarowe zależności, są dobrym narzędziem do przewidywania wartości oraz wykrywania anomalii w finansach. Wiele wcześniejszych prac udowodniło, że sieci neuronowe są skuteczniejsze od tradycyjnych metod ekonometrycznych nie tylko w przewidywaniu kursu wymiany walut n-dni w przód, ale także w przewidywaniu kierunku zmiany (w krótkim okresie). Li i Ma wymieniają także kilkanaście prac, w których uzyskano zadowalające wyniki przewidywania wartości na rynku kapitałowym, a kierowanie się strategią inwestycyjną opartą o sygnały generowane przez sieci neuronowe pozwalało uzyskiwać lepsze zwroty z inwestycji niż w oparciu o sygnały generowane przez inne metody statystyczne. Ponadto, sieci neuronowe zostały z sukcesem wykorzystane w analizie ryzyka kredytowego oraz w wykrywaniu anomalii (co pozwala blokować oszukańcze transakcje).<sup>124</sup>

Saad i in. (1998) założyli, że przewidywanie wartości w krótkim okresie czasu jest trafniejsze i zbudowali trzy sieci neuronowe w oparciu o różne architektury (opóźniona w czasie sieć neuronowa, rekurencyjna sieć neuronowa oraz probabilistyczna sieć neuronowa), których zadaniem było zidentyfikować, czy w następnych 22 dniach wartość akcji danych spółek wzrośnie o więcej niż 2% (co przy odpowiednim wykorzystaniu odpowiedniego efektu dźwigni może dać inwestorowi realny zarobek). Sieci nie miały więc przewidywać konkretnych wartości, a raczej klasyfikować trend (wzrost powyżej 2% versus mniejszy wzrost lub spadek).<sup>125</sup>

Na rynek kapitałowy wpływają dwa rodzaje czynników – losowe i chaotyczne. Losowe są nieprzewidywalne, ale chaotyczne w wielu przypadkach mogą zostać poprawnie sklasyfikowane przez model sieci neuronowej, która na podstawie historycznych oszacuje funkcję, która opisuje te zależności. Ze względu na wielowymiarowość problemów z rynku kapitałowego, trudno ten efekt osiągnąć przy pomocy konwencjonalnych (czyli liniowych lub niskowymiarowych) metod. Dzięki zastosowaniu odpowiednich funkcji straty (penalizujących fałszywie pozytywne przewidywania

---

<sup>124</sup> Li, Y., Ma, W., 2010, Applications of Artificial Neural Networks in Financial Economics: A Survey. IEEE 2010 3rd International Symposium on Computational Intelligence and Design (ISCID) - Hangzhou, China (2010.10.29-2010.10.31), s. 211–214.

<sup>125</sup> Saad, E.W., Prokhorov, D.V., Wunsch, D.C., 1998, Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks, IEEE Transactions on Neural Networks, 9(6), s. 1456.

bardziej niż utracone szanse), sieci wykazywały niewielki odsetek fałszywie pozytywnych klasyfikacji, co jest niezwykle ważne, gdy handluje się z wykorzystaniem efektu dźwigni. W przypadku kursu akcji firmy IBM, w testowanym okresie było 150 rzeczywistych okresów, w których można było osiągnąć zysk. Trzy badane sieci zidentyfikowały kolejno: 38, 37 oraz 78 z nich, przy czym fałszywie pozytywnej klasyfikacji dokonały tylko w: 1, 3, 1 przypadkach. Porównywalnie dobre rezultaty osiągały w przypadku akcji innych spółek.<sup>126</sup>

Zhu i in. (2008) zauważają, że wielu naukowców udokumentowało skuteczność zastosowania sieci neuronowych w przewidywaniu wartości indeksów giełdowych. Najważniejszym czynnikiem pozwalającym na skuteczne przewidywanie jest dobór danych wejściowych. Sieci neuronowe przeprowadzające regresje na szeregach czasowych mogą wykorzystywać jedną zmienną (tylko kurs danej akcji) lub wiele zmiennych. Wykorzystanie większej ilości zmiennych pozwala sieci na zidentyfikowanie innych istotnych informacji (na przykład wartości indeksów giełdowych, wskaźników analizy technicznej czy danych z analizy fundamentalnej spółek), ale wiąże się z koniecznością zgromadzenia większej ilości danych i odpowiedniego doboru zmiennych, które mogą mieć realny wpływ na prognozowaną wartość. W badaniu sprawdzono, jak zmieniają się wyniki modeli przewidujących przyszłe wartości indeksów giełdowych (jeden dzień w przód), gdy uwzględną się w nich wolumen obrotu (który nieliniowo wpływa na kurs akcji).<sup>127</sup>

W badaniu wykorzystano sieci neuronowe z trzema warstwami, w których znajdowało się 5, 10, 15 lub 20 neuronów. Do tak zbudowanych sieci zaimplementowano dane z rynków kapitałowych – wartości indeksów giełdowych DJIA, NASDAQ i STI oraz wartości poszczególnych akcji wchodzących w skład tych indeksów, po czym wytrenowano je ponownie, ale tym razem do danych wejściowych dołożono również informacje o wolumenie obrotu. Wykazano że sieci, które mają dostęp do danych o wolumenie obrotu uczą się szybciej (dochodzą do optymalnych efektów w mniejszej ilości epok). Ponadto, sieci z większą liczbą neuronów osiągają większą skuteczność przewidywania. Jeśli chodzi o wyniki na zbiorach treningowych – we wszystkich przypadkach lepiej radziły sobie sieci, które miały dostęp do danych o wolumenie obrotu. Natomiast w przypadku zbiorów testowych wyniki nie były już tak spójne – w przypadku

---

<sup>126</sup> Ibidem.

<sup>127</sup> Zhu, X., Wang, H., Xu, L., Li, H., 2008, Predicting stock index increments by neural networks: The role of trading volume under different horizons, *Expert Systems with Applications*, 34(4), s. 3043 – 3044.

sieci operujących na danych z krótkiego horyzontu czasowego dodanie wolumenu obrotu nie poprawiło jej wyników. W przypadku sieci badających długi horyzont czasowy – dane o wolumenie obrotu okazały się poprawiać wyniki sieci (najlepszy okazał się model, który miał dostęp do danych o wolumenie obrotu 3 spółek z danego indeksu, których wolumen obrotu wykazywał największą korelację z obrotem na akcjach całego indeksu).<sup>128</sup>

Mimo że sieci LSTM rozpowszechniły się dopiero w 2015, w związku z ich możliwościami analizy danych z szeregów czasowych oraz zdolności do uchwycenia zależności długoterminowych pomiędzy zmiennymi objaśniającymi a zmiennymi objaśnianymi, są często stosowane w pracach naukowych dotyczących zagadnień finansowych i ekonomicznych.<sup>129</sup> LSTM to nie jedyna architektura uczenia maszynowego, którą powszechnie wykorzystuje się (z zadowalającymi wynikami) do badania zagadnień ekonomicznych. Do tej grupy zaliczają się także sieci GAN, GRU czy niektóre rodzaje sieci konwolucyjnych. W części badawczej zdecydowano się na wykorzystanie sieci LSTM w związku z ich powszechnie uznaną skutecznością w analizie szeregów czasowych oraz przez wzgląd na to, że sieć sama uczy się na ile ważne są dane z okresów wcześniejszych.

Namini i in. (2018) zestawiają prognozowanie (regresję) przyszłych wartości dla szeregów czasowych za pomocą LSTM oraz za pomocą kilku stochastycznych modeli takich jak: ARMA, ARIMA czy VAR (zwłaszcza wariantów analizujących wiele zmiennych). ARIMA jest najczęściej stosowanym modelem do analizowania szeregów czasowych i ma wysoką skuteczność. Natomiast ARIMA ma niewielką zdolność do wykrywania nieliniowych zależności między zmiennymi (do czego zdolne są głębokie sieci neuronowe).<sup>130</sup>

W badaniu użyto miesięcznych danych z przedziału od stycznia 1985 do sierpnia 2018 roku dla indeksów: Nikkei 225, NASDAQ, Hang Seng, S&P500, Dow Jones oraz danych makroekonomicznych, w tym między innymi: dane dotyczące budownictwa mieszkaniowego, ważony wolumenem handlu indeks dolara amerykańskiego, indeks żywności i napojów, czy podaż pieniądza M1. Dla wszystkich wymienionych zmiennych wybrano wartość zamknięcia, a pozostałe (otwarcie, najwyższa, najniższa) odrzucono. Wyselekcjonowano 70% z danych na zbiór

---

<sup>128</sup> Zhu i in., Op. Cit., s. 3045 - 3054

<sup>129</sup> Géron, Op. Cit., s. 379.

<sup>130</sup> Siami-Namini, S., Siami-Namini A., 2018, "Forecasting Economics and Financial Time Series: ARIMA vs. LSTM." *ArXiv abs/1803.06386*. s. 3 - 5

treningowy i 30% na zbiór testowy, a za miarę skuteczności modelu wykorzystano RMSE (pierwiastkowy błąd średniokwadratowy). Modele zostały zbudowane tak, że przewidują miesiąc do przodu (wartość zamknięcia następnego miesiąca), korzystając ze wszystkich dostępnych (wcześniejszych) w danej iteracji miesięcy (metoda *walk-forward model validation*). W przewidywaniu wartości indeksu giełdowego dla następnego miesiąca osiągnięto znacznie celniejsze wyniki za pomocą modelu LSTM niż przy użyciu modelu ARIMA - RMSE było mniejsze o 84-87%, wyniki sieci LSTM były lepsze dla każdego ze sprawdzanych indeksów.<sup>131</sup>

Heaton et al. (2018) wysnuwają tezę, że wszystkie informacje istotne dla predykcji danych ekonomicznych znajdują się w dostępnych i zebranych danych, ale przez to jak gigantyczne są to ilości i jak bardzo rozproszone po różnych zbiorach danych – trudno jest wykorzystywać je do ręcznej analizy. Ponadto, obecnie funkcjonujące modele tradycyjne nie potrafią (lub są mało skuteczne) w odnajdywaniu skomplikowanych nieliniowych zależności w danych. Zwrócono także uwagę na to, że istnieje wiele modeli prognozujących przyszłe wartości, dla których brakuje teoretycznego uzasadnienia, bo teoria która uzasadniałaby dane zjawisko, nie została jeszcze opisana. Przynajmniej część z powyższych problemów powinno dać się rozwiązać przez zastosowanie sieci neuronowych – dane wejściowe mogą być rozszerzone, by zawrzeć wszystkie informacje mające powiązanie z problemem; nieliniowości i zawile interakcje pomiędzy danymi wejściowymi są wychwytywane przez sieci neuronowe; łatwiej kontrolować nadmierne dopasowanie.<sup>132</sup>

W badaniu porównuje się prognozowanie wartości przy pomocy klasycznych modeli (*PCA*, *RRR*, *LDA*) do prognozowania za pomocą sieci neuronowych, zauważając istotną przewagę tych drugich – używając więcej niż dwóch warstw ukrytych w sieci neuronowej można stworzyć model, który przeanalizuje wszystkie dostępne dane, zależności między nimi oraz weźmie pod uwagę nieliniowe powiązania. W takim sensie sieci neuronowe mogą być postrzegane jak liniowe funkcje składające się z rekursywnie zdefiniowanych nieliniowych funkcji. Dzięki temu nadają się do prognozowania wartości przy użyciu wielu zmiennych objaśniających.<sup>133</sup>

---

<sup>131</sup> Namini i in. , Op. Cit. s. 8 – 15.

<sup>132</sup> Heaton, J. B., Polson, N.G., Witte, J.H., 2018, "Deep Learning in Finance". ArXiv abs/1602.06561 s. 2.

<sup>133</sup> Heaton et al, Op. Cit. s. 13 – 18.

Analiza ryzyka kredytowego to kolejny obszar, w którym sieci neuronowe znajdują zastosowanie. Łącząc zmienne różnego typu (między innymi dane finansowe z danymi księgowymi oraz danymi tekstowymi – transkryptami spotkań, podczas których omawiane są wyniki przedsiębiorstwa) i używając funkcji *softmax* w warstwie wyjściowej można określić sytuację finansową danego podmiotu.<sup>134</sup>

Hansson (2017) porównuje skuteczność sieci LSTM (różnej głębokości) z konwencjonalnymi modelami wykorzystywanymi w analizie szeregów czasowych (ARMA-GJRARCH) w przewidywaniu zwrotów inwestycji giełdowych. Swoją analizę przeprowadził na trzech indeksach – S&P 500 (USA), Bovespa 50 (Brazylia) oraz OMX 30 (Szwecja). Model ARMA-GJRARCH jest powszechnie wykorzystywany w analizie szeregów czasowych dla danych finansowych, więc stanowi dobry punkt odniesienia dla sprawdzenia wyników sieci LSTM. Sprawdzono jak powyżej opisane modele oraz sieci LSTM (jednowarstwowe oraz głębokie) poradziły sobie z regresją – przewidywaniem wartości indeksu jeden dzień w przód oraz z klasyfikacją – przewidywaniem, czy następnego dnia wartość indeksu będzie wyższa czy niższa.<sup>135</sup>

W badaniu sprawdzono dwie różne sieci LSTM – pierwsza z nich była jednowarstwowa – z jedną warstwą ukrytą składającą się z czterech neuronów LSTM. Druga z nich to głębsza sieć składająca się z 3 warstw ukrytych (zawierających kolejno: 4, 50, 100 neuronów LSTM). Przy czym druga i trzecia warstwa głębszej sieci zostały poddane zabiegowi *dropout* z szansą na pominięcie wyniku pojedynczych neuronów na poziomie 50%. Obie sieci dla zadania regresji miały neuron z liniową aktywacją w warstwie wyjściowej, a dla zadania klasyfikacji – neuron z funkcją aktywacji *softmax*.

Obie sieci LSTM dały wyniki zbliżone do modeli ARMA i GJRARCH, które były zbliżone do średniej kursu w całym okresie (takiej sieci nie można więc użyć do tworzenia strategii inwestycyjnej). Suma kwadratów reszt wszystkich sprawdzanych w tym badaniu modeli była bardzo zbliżona, natomiast ARMA-GJRARCH osiągnął minimalnie lepszy wynik dla indeksu SP500.<sup>136</sup>

---

<sup>134</sup> Ibidem.

<sup>135</sup> Hansson, M., 2017, On stock return prediction with LSTM networks. LUP Student Papers, Lund, s. 1 i 19 - 21.

<sup>136</sup> Hansson., Op. Cit., s. 21 – 31.



W zadaniu klasyfikacji obie sieci LSTM w danym okresie dla indeksu S&P 500 przewidywały nieustanny wzrost kursu, a głęboka sieć LSTM tak sam wynik dała również dla indeksu Bovespa. W przypadku indeksu OMX – oba modele dały mniej regularne wyniki. Najdokładniejszym uzyskanym wynikiem było przewidywanie kierunku zmian w kursie indeksu na następny dzień dla OMX przy użyciu głębokiej sieci LSTM z funkcją aktywacji softmax w warstwie wyjściowej, gdzie osiągnął skuteczność przewidywania na poziomie 55,3% (statystycznie istotny na poziomie  $\alpha = 50\%$  w teście dwumianowym). W związku z tym, że dla SP500 oraz Bovespa sieć LSTM przewiduje ciągły wzrost kursu, strategia inwestycyjna, która polega na kupowaniu indeksu, jeśli następnego dnia przewiduje się wzrost jego wartości oraz sprzedawaniu, jeśli następnego dnia przewiduje się spadek jego wartości będzie zawsze zalecała kupno, a więc wartość inwestycji będzie podążała za rzeczywistymi zmianami wartości indeksu. W przypadku OMX głęboka sieć LSTM przewidywała wzrosty i spadki wartości z 55,3% skuteczności i dzięki temu strategię inwestycyjną dawały 50%-100% lepsze zwroty niż indeks. W tym przypadku sieć LSTM może być dobrym narzędziem do wspomagania strategii inwestycyjnej, natomiast należy zwrócić uwagę na to, że dla OMX różnice między treningowym zbiorem danych a testowym zbiorem danych były najmniejsze (więc sieci łatwiej było generalizować dla testowych danych).<sup>137</sup>

Wenjie i in. (2020) uważają, że klasyczne modele takie jak VAR, BVAR, ARIMA i GARCH osiągają małą skuteczność w prognozowaniu przyszłych wartości szeregów czasowych dla kursów giełdowych w związku z dużym zaszumieniem danych oraz w związku z tym że zależności między zmiennymi objaśniającymi, a objaśnianymi podatne są na zmiany wraz z upływem czasu. Pojedyncza sieć neuronowa również nie jest rozwiązaniem optymalnym dla prognozowania i proponują wykorzystanie możliwości łączenia różnych rodzajów sieci w jednym modelu, tworząc sieć składającą się z neuronów LSTM oraz konwolucyjnych (dla optymalnego wykorzystania informacji znajdujących się w danych wejściowych).<sup>138</sup>

W badaniu wykorzystano dane dla *Shanghai Composite Index* (cena otwarcia, zamknięcia, najniższa, najwyższa i zamknięcia oraz wolumen obrotu i procentowa zmiana wartości), żeby nauczyć model CNN-LSTM przewidywać cenę zamknięcia indeksu jeden dzień w przód. Oprócz

---

<sup>137</sup> Ibidem.

<sup>138</sup> Lu, W., Li, J., Li, Y., Sun, A., Wang, J.. 2020. "A CNN-LSTM-Based Model to Forecast Stock Prices". Complexity. vol. 2020. Article ID 6622927, s. 1 – 2.

modelu CNN-LSTM takie same dane dostarczyli do innych modeli sieci neuronowych – MLP, CNN, RNN, LSTM oraz CNN-RNN, które służyły za punkty odniesienia. Wybrano trzy metryki do porównywania modeli - średni błąd absolutny ( $MAE$ ), pierwiastkowy błąd średniokwadratowy ( $RMSE$ ) oraz współczynnik determinacji  $R^2$ .<sup>139</sup>

Dane wejściowe modelu miały postać trójwymiarowego wektora (Brak, 10, 8), w którym 10 to długość okienka czasowego (liczba dni sesyjnych), a 8 to liczba atrybutów w każdym dniu sesyjnym. Tak zbudowany wektor danych wejściowych najpierw trafiał do warstwy konwulucyjnej, później do warstwy *max pool*, następnie do 32-neuronowej warstwy LSTM, a na końcu do 64-neuronowej warstwy w pełni połączonej. Warstwą wyjściową sieci stanowił jeden neuron przewidujący wartość zamknięcia dla kolejnego dnia sesyjnego. W okresie zawartym w zbiorze testowym indeks *Shanghai Composite Index* osiągał wartości z przedziału (2500, 3500), a najlepszym ze sprawdzanych modeli był model CNN-LSTM, który miał  $MAE$  na poziomie 27,564,  $RMSE$  na poziomie 39,688, a jego  $R^2$  wynosiło 0,9646. Wenjie i in. konkludują, że z dokładnością na takim poziomie zaprezentowany model może służyć jako narzędzie pomagające inwestorom dokonywać decyzji inwestycyjnych.<sup>140</sup>

Minami (2018) zauważa trudności w prognozowaniu wartości giełdowych związane z tym, że dane historyczne dotyczące cen walorów nie są jedynymi czynnikami wpływającymi na zmiany cen akcji czy indeksów – inwestorzy posługują się różnymi metodami, takimi jak określanie wartości wewnętrznej akcji, modelem zdyskontowanych przepływów, analizą techniczną, czy czynnikami emocjonalno-behawiorystycznymi (na przykład kupując akcje ulubionej firmy lub kupując akcje w związku z dużym wolumenem obrotu na jakimś walorze). Trafność prognoz można zwiększyć, jeśli oprócz danych historycznych dotyczących samego indeksu giełdowego, wykorzysta się również dane makroekonomiczne dla badanego okresu oraz informacje publikowane przez spółki, które są istotne dla ich sytuacji finansowej. Minami proponuje architekturę złożoną z neuronów LSTM, które sprawdzają się w analizie szeregów czasowych o długim horyzoncie oraz uwzględniają długo i krótkoczasowe zależności pomiędzy danymi.<sup>141</sup>

---

<sup>139</sup> Wenjie i in., Op. cit. s. 4 – 10.

<sup>140</sup> Ibidem.

<sup>141</sup> Minami, S., 2018, "Predicting Equity Price with Corporate Action Events Using LSTM-RNN". Journal of Mathematical Finance, 8, s. 58 - 60

Model sprawdzono na akcjach jednego przedsiębiorstwa – Tsugami Corporation (by móc skutecznie wykorzystać publikowane przez nie informacje). Oprócz danych pochodzących z informacji prasowych, wykorzystano także informacje dotyczące historycznych cen waloru oraz danych makroekonomicznych. Dane makroekonomiczne, które ogłaszane są miesięcznie lub kwartalnie, zostały poddane interpolacji funkcjami sklejanymi, by zastosować je w ujęciu dziennym. Model wykorzystuje 1082 dni sesyjne (1082 obserwacje), z czego 70% należy do zbioru uczącego, a 30% do zbioru testowego, a miarą jego skuteczności był RMSE. Badanie pokazało, że sieć LSTM wykorzystująca tylko dane z informacji prasowych miała niższy pierwiastkowy błąd średniokwadratowy niż sieć wykorzystująca tylko historię cen tego waloru. Natomiast sieć, która wykorzystywała wszystkie dostępne dane (historia ceny, informacje prasowe oraz dane makroekonomiczne) osiągnęła gorsze rezultaty niż sieci z ograniczoną liczbą atrybutów (przy 500 epokach uczenia RMSE wynosiło 15,3 dla wszystkich danych, 15,0 dla historii cen oraz 14,9 dla danych z informacji prasowych).<sup>142</sup>

Gamboa (2017) zauważa w swojej pracy, że tradycyjne głębokie sieci neuronowe (i ich odmiany) rzadko radzą sobie lepiej wraz ze wzrostem ilości warstw sieci. Natomiast często dobrym sposobem zwiększania skuteczności przewidywania sieci neuronowej jest: zastosowanie uczenia nienadzorowanego lub ułożenie architektury składającej się z kilku/wielu osobnych sieci neuronowych, w której wynik jednej sieci stanowi dane wejściowe kolejnej. Gamboa zauważa, że w analizie szeregów czasowych (pod kątem wykrywania anomalii) wysoką skuteczność osiągają nałożone na siebie sieci składające się z neuronów LSTM.<sup>143</sup>

---

<sup>142</sup> Minami, Op. cit., s. 60 - 63

<sup>143</sup> Gamboa, J., 2017, "Deep Learning for Time-Series Analysis", ArXiv abs/1701.01887, s.

### **3. ZASTOSOWANIE MODELU SIECI LSTM DO BUDOWY SYSTEMU TRANSAKCYJNEGO DLA KONTRAKTU NA INDEKS NASDAQ-100 W LATACH 2003 – 2019 – BADANIE EMPIRYCZNE**

W poprzednich rozdziałach pracy opisano teorię dotyczącą tego, jakie ogłoszenia makroekonomiczne oraz dane z rynków oddziałują na kursy akcji i indeksów giełdowych. Opisano także nowoczesne narzędzia, które mogą posłużyć do analizy tych danych. Trzeci rozdział będzie stanowił opis badania empirycznego, w którym narzędzia te zostaną użyte do weryfikacji postawionej we wprowadzeniu hipotezy, że sieci zbudowane w oparciu o neurony LSTM, wykorzystujące historyczne dane, mogą stanowić skuteczne narzędzie do generowania sygnałów inwestycyjnych.

#### **3.1. Opis modelowanego waloru**

Zdecydowano się przeprowadzić badanie na wartości kontraktów futures na indeks NASDAQ-100, który obejmuje akcje 100 największych, niefinansowych spółek z giełdy NASDAQ. Wybór tego kontraktu podyktowany był tym, że do właściwego funkcjonowania sieci neuronowej potrzebny jest instrument o wystarczająco dużej płynności, o którym można zgromadzić odpowiednią ilość danych. Ponadto, badanie przeprowadzono stosując relatywnie nowe i stale rozwijane metody statystyczne, służące do prognozowania, co poniekąd łączy się z indeksem technologicznym. Można stwierdzić, że taki model bada tylko jeden sektor gospodarki (wszystkie spółki w skład jego wchodzące są związane z nowoczesnymi technologiami). Spółki z jednego sektora zazwyczaj podobnie reagują na konkretne ogłoszenia makroekonomiczne, co powinno ułatwić sieci neuronowej znajdowanie powiązań pomiędzy konkretnym ogłoszeniem makroekonomicznym, a zmianą wartości indeksu (kontraktu na indeks). Co ważne – w indeksie tym nie ma żadnych spółek z sektora finansowego, które zachowują się inaczej i są bardziej wrażliwe na niektóre ogłoszenia makroekonomiczne (na przykład na zmiany stóp procentowych).

Kolejnym powodem, dla którego wybrano NASDAQ-100 jest jego większa zmienność w porównaniu do innych dużych indeksów ze Stanów Zjednoczonych (Dow Jones oraz S&P 500). Rysunek 21. przedstawia wahania największych indeksów z giełd amerykańskich. Wahania indeksu NASDAQ-100 są znacznie większe niż w przypadku S&P 500 oraz Dow Jones. Większa

zmienność wiąże się z większymi możliwościami zysku oraz straty. Można zaobserwować, że w latach 1999 - 2020 NASDAQ-100 wykazywał bardzo silny trend wzrostowy, co może zaburzać działanie modelu i proporcjonalnie zwiększać liczbę generowanych sygnałów kupna.

Rys. 21. Porównanie zmienności trzech największych indeksów giełdowych Stanów Zjednoczonych.



Źródło: <https://capital.com/nasdaq-100-vs-sp-500> (dostęp na 20.08.2020)

Pełna nazwa kontraktu to E-mini Nasdaq-100 Futures (NQ). Wybrano ten kontrakt z powodów opisanych powyżej oraz dlatego, że jest jednym z największych kontraktów futures opartych na amerykańskich indeksach, nie jest jednak największym z nich. Dla przykładu w styczniu 2019 roku zawarto dziennie średnio 530.000 kontraktów na E-mini Nasdaq-100 futures oraz aż 1.600.000 kontraktów na E-mini S&P 500 futures (instrument o największym obrocie wśród tych na giełdzie CME bazujących na indeksach). E-mini Nasdaq-100 futures był drugim najczęściej zawierany kontrakt, a w innych miesiącach 2019 roku wartości te rozkładały się w podobnych proporcjach.<sup>144</sup>

Kontrakt E-mini Nasdaq-100 Futures (NQ) jest płynnym instrumentem pochodnym, którym handluje się w trzymiesięcznych cyklach z terminem wygaśnięcia przypadającym zazwyczaj na trzeci piątek miesiąca. E-mini Nasdaq-100 Futures (NQ) posiada mnożnik 20, czyli za każdy punkt wartości indeksu NASDAQ 100 inwestor zyskuje (traci) 20\$. Najmniejsza możliwa

<sup>144</sup> [https://www.cmegroup.com/media-room/press-releases/2019/2/04/cme\\_group\\_averaged178millioncontractsperdayinjanuary2019.html](https://www.cmegroup.com/media-room/press-releases/2019/2/04/cme_group_averaged178millioncontractsperdayinjanuary2019.html), 20.08.2020.

rejestrowana zmiana to 0,25 punkta indeksowego ( $0,25 * 20\$ = 5\$$ ).<sup>145</sup> Jednak do badania zostanie wykorzystane wystandaryzowane notowanie kontraktu, które pozwala na pozbycie się problemu różnych terminów wykonania poszczególnych kontraktów, sporządzenie wykresu kontynuacyjnego dla dłuższego okresu czasu i wyliczenie wskaźników analizy technicznej dla całego okresu badania. Dane zostały pobrane z serwisu stooq.pl.<sup>146</sup>

W związku z dynamicznie rosnącą wartością indeksu NASDAQ 100, w maju 2019 roku CME zdecydowało się na wprowadzenie do obiegu nowego instrumentu: Micro E-mini Nasdaq-100 (MNQ). Od wersji opisywanej w poprzednim akapicie różni się jedynie mnożnikiem kontraktu (2\$ razy wartość indeksu NASDAQ 100 zamiast 20\$ razy jego wartość). Dzięki temu stał się on alternatywą dla mniejszych inwestorów, którzy nie chcą ryzykować wyższych kwot.<sup>147</sup> Marża utrzymania intraday dla kontraktu micro w lipcu 2020 roku wynosiła 1500\$, a dla kontraktu mini 15.000\$.<sup>148</sup>

Odrzucono modelowanie kontraktu micro, ponieważ mimo dużej płynności (w pierwszej połowie 2020 roku średnio dziennie obracano 579.305 kontraktami micro<sup>149</sup>), instrument ten jest na rynku zbyt krótko, by uzyskać wystarczająco dużo obserwacji dla autoregresji przy częstotliwości obserwacji wynoszącej 1 dzień.

### 3.2. Dane modelu

Głównym przyjętym założeniem w wybieraniu danych do modelu było wykorzystanie wiedzy na temat czynników kształtujących ceny na rynku kapitałowym oraz informacji pochodzących z analizy technicznej (poprzez historię cen dla modelowanego waloru – bez sygnalizowania sieci, jakich kształtów powinna szukać). Wykorzystano więc wskaźniki analizy technicznej, obliczone dla kursu kontraktu E-mini Nasdaq-100 Futures oraz dane makroekonomiczne, które mogą się przyczyniać do wyceny kursów akcji na giełdzie.

Wyselekcjonowano szereg czasowy z przedziału dat 02.01.2003 – 07.11.2019, a każdą obserwację stanowił jeden dzień. Początkowo było więc 6154 obserwacji. Wybór takiego zakresu

---

<sup>145</sup> [https://www.cmegroup.com/trading/equity-index/us-index/e-mini-nasdaq-100\\_contract\\_specifications.html](https://www.cmegroup.com/trading/equity-index/us-index/e-mini-nasdaq-100_contract_specifications.html), 20.08.2020.

<sup>146</sup> <https://stooq.pl/q/?s=nq.f>, 20.08.2020.

<sup>147</sup> [https://www.cmegroup.com/trading/equity-index/us-index/micro-e-mini-nasdaq-100\\_contract\\_specifications.html](https://www.cmegroup.com/trading/equity-index/us-index/micro-e-mini-nasdaq-100_contract_specifications.html), 20.08.2020.

<sup>148</sup> <https://tickertape.tdameritrade.com/trading/micro-e-mini-index-futures-earnings-season-17619>, 20.08.2020.

<sup>149</sup> <https://tickertape.tdameritrade.com/trading/micro-e-mini-index-futures-earnings-season-17619>, 20.08.2020.

dat jest uzasadniony chęcią wykorzystania jak najdłuższego okresu czasowego, ale rozpoczynającego się po okresie gwałtownych wahań ceny wywołanych bańką internetową. Zdecydowano się uwzględnić kryzys z 2008 roku, żeby sieć neuronowa miała szansę uodpornić się na zmianę koniunktury i duże spadki wartości walorów.

Dla każdego dnia uwzględniono tylko jedną obserwację danego waloru – dla danych giełdowych takich jak: kursy walut, wartości indeksów, wartości surowców wykorzystano wartość, którą stanowiła średnia kursu otwarcia i kursu zamknięcia. Zabieg ten miał na celu ograniczenie ilości zmiennych, które niewiele wnoszą do modelu. Ponadto, w przybliżeniu można uznać, że wartość średnia z danego dnia to dobry punkt odniesienia dla decyzji inwestycyjnych podejmowanych przez inwestora. Dla danych makroekonomicznych, które ogłaszane są raz na tydzień, raz na miesiąc, lub rzadziej wykorzystano tę samą wartość dla każdego kolejnego dnia aż do następnego jej ogłoszenia. Innym sposobem radzenia sobie z miesięcznymi danymi w szeregu czasowym o dziennym interwale jest zastosowanie interpolacji wielomianowymi funkcjami sklejanymi, co pozwoliłoby na uzyskanie nieznacznie różniących się obserwacji dla każdego dnia. W przypadku tego modelu zdecydowano się nie korzystać z tej opcji, ponieważ dane wejściowe sieci będą miały postać sekwencji w formie okienek czasowych dla kilku-kilkunastu dni sesyjnych. Dzięki temu sieć może nauczyć się, jak zmiana danego parametru makroekonomicznego wpływa na walory na giełdzie w momencie jej ogłoszenia.

Tabela 1. Początkowo wybrane zmienne.

Nazwa zmiennej w modelu	Ticker	Opis zmiennej
AUDUSD	AUDUSD	Kurs dolara australijskiego względem dolara amerykańskiego
CPIUS	CPIYUS.M	Wskaźnik cen towarów i usług konsumpcyjnych rok do roku - miara inflacji (dane miesięczne)
CrudeOilF	CL.F	Cenę kontraktu futures na baryłkę ropy naftowej notowaną na NYMEX
CrudeOilFV	CL.F	Wolumen obrotu powyższym kontraktem CL.F
DJC	^DJC	Wartość indeksu Dow Jones Composite
DJCV	^DJC	Wolumen obrotu na powyższym indeksie
DJI	^DJI	Wartość indeksu przemysłowego Dow Jones
DJIV	^DJI	Wolumen obrotu na powyższym indeksie
DJT	^DJT	Wartość indeksu transportowego Dow Jones
DJTV	^DJT	Wolumen obrotu na powyższym indeksie
DJU	^DJU	Wartość indeksu spółek użyteczności publicznej
DJUV	^DJU	Wolumen obrotu na powyższym indeksie
FEDFunds	-	Stopy procentowe FED
MNGDP	-	Nominalny krajowy produkt brutto w ujęciu miesięcznym dla USA
MRGDP	-	Realny krajowy produkt brutto w ujęciu miesięcznym dla USA
NASDAQC	^NDQ	Wartość indeksu Nasdaq Composite
NASDAQCV	^NDQ	Wolumen obrotu na powyższym indeksie
NASDAQ100	^NDX	Wartość indeksu Nasdaq 100 (100 największych spółek na Nasdaq)
NASDAQ100V	^NDX	Wolumen obrotu na powyższym indeksie
S&P500	^SPX	Wartość indeksu zawierającego 500 największych spółek notowanych na giełdach amerykańskich
S&P500V	^SPX	Wolumen obrotu na powyższym indeksie
Unemprate	UNRTUS.M	Stopa bezrobocia w USA (dane miesięczne)
USDCHF	USDCHF	Kurs franka szwajcarskiego względem dolara amerykańskiego
USDCNY	USDCNY	Kurs chińskiego yuana względem dolara amerykańskiego
USDEUR	USDEUR	Kurs euro względem dolara amerykańskiego
USDGBP	USDGBP	Kurs funta brytyjskiego względem dolara amerykańskiego
USDJPY	USDJPY	Kurs jena japońskiego względem dolara amerykańskiego
VIX	^VIX	Wartość indeksu zmienności CBOE - miary oczekiwanej zmienności na opcjach S&P 500
CRB	^CRY	Wartość indeksu CRB, którego składowymi są ceny dóbr (m.in. złoto i ropa naftowa)
NASDAQ100F	NQ.F	Wartość indeksu dla kontraktu E-mini na Nasdaq 100
NASDAQ100FV	NQ.F	Wolumen obrotu na powyższym indeksie
NASDAQ100FCHANGE	-	Procentowa zmiana względem poprzedniego dnia sesyjnego dla indeksu NQ.F

Źródło: Opracowanie własne.

Tak zbudowana baza danych musiała zostać odpowiednio dostosowana, nim można było wykorzystać ją jako dane wsadowe sieci neuronowej. Pierwszym krokiem było usunięcie dni niesesyjnych dla kontraktu futures na Nasdaq 100 (czyli takich, w których nie ma notowania tego waloru) dla wszystkich zmiennych, po czym dodano kolumnę opisaną w tabeli jako



NASDAQ100FCHANGE, czyli zmiana procentowa kursu waloru z poprzedniej dnia sesyjnego na bieżący. Następnie usunięto pierwszy wiersz w związku z brakiem możliwości wyliczenia dla niego zmiany procentowej z poprzedniej sesji. W związku z tym, że dni sesyjne dla walorów różnią się (w zależności od kraju w którym są notowane, świąt narodowych i innych czynników), po usunięciu niesesyjnych dni NASDAQ100F, dla innych walorów wciąż pozostawały dni, w których brakowało wartości. Żeby nie tracić cennych z punktu widzenia sieci danych, zdecydowano się zastosować jedną z polecanych przez Gerona (2017)<sup>150</sup> technikę radzenia sobie z brakującymi danymi (dla zmiennych opisujących) i uzupełniono je średnią kroczącą z 8 okalających (zawierających obserwacje) dni. Kolejnym krokiem było usunięcie kolumny NASDAQ100F, jako iż, z punktu widzenia sieci, dostarcza dokładnie tych samych danych co NASDAQ100FCHANGE. Na końcu stworzono jeszcze jedną kolumnę poprzez przesunięcie NASDAQ100FCHANGE jeden dzień sesyjny w przód, w celu stworzenia etykiet dla uczenia nadzorowanego.

Po przetestowaniu kilku sieci neuronowych ze zmiennymi opisanymi w Tabeli 1. wyniki okazały się niezadowalające, więc zdecydowano się na dołożenie większej ilości zmiennych – kolejne wskaźniki makroekonomiczne, najważniejsze indeksy giełdowe z innych krajów oraz wskaźniki analizy technicznej.

Użyto kilku powszechnie stosowanych wskaźników: MACD z 12 i 26 sesji z linią sygnałną z 9 sesji, oscylator *Ultimate*, Wskaźnik siły względnej, Wskaźnik zmiany ROC z 15 sesji, *Commodity channel index* oraz %R Williamsa – wszystkie obliczono w programie Excel dla kursu NASDAQ100F. Zastosowano MACD z linią sygnałną, by sieć mogła nauczyć się, w których momentach przecinania się średnich kroczących z linią sygnałną, należy podejmować decyzje inwestycyjne (nie użyto etykiet, żeby umożliwić sieci nauczanie się właściwych interpretacji). Oscylator *Ultimate* został dodany, by sieć mogła wytrenować się w rozpoznawaniu „wykupienia” indeksu, a RSI oraz ROC miały umożliwić sieci analizę kierunku zmiany dla określonej ilości sesji. Z kolei CCI pozwala zidentyfikować cykliczne trendy, co może ułatwić sieci określenie przyszłych wartości w danym cyklu.

---

<sup>150</sup> Geron, Op. Cit., s. 60 - 61

Tabela 2. Zmienne dodane do modelu.

Nazwa zmiennej w modelu	Ticker	Opis zmiennej
M1	-	Tygodniowa podaż pieniądza M1 w USA
HOUSESTART	-	Liczba rozpoczętych budów domów w USA (tygodniowo)
BALANCETRADE	-	Bilans handlowy USA (dobra i usługi), dane miesięczne
INDPRO	-	Indeks produkcji przemysłowej USA, gdzie wartość z 2012 stanowi 100
PCONSUMPEXPEND	-	Wydatki na konsumpcję USA (miesięcznie w miliardach USD)
DAX	^DAX	Wartość głównego indeksu giełdy niemieckiej DAX
DAXV	^DAX	Wolumen obrotu na powyższym indeksie
FTSE100	^UKX	Wartość indeksu giełdy londyńskiej FTSE 100
FTSE100V	^UKX	Wolumen obrotu na powyższym indeksie
SHCOMP	^SHC	Wartość indeksu Shanghai Composite
HANGSENGIND	^HSI	Wartość indeksu Hang Seng giełdy koreańskiej
NIKKEI225	^NKX	Wartość indeksu Nikkei 225 giełdy japońskiej
NIKKEI225V	^NKX	Wolumen obrotu na powyższym indeksie
CrudeOilSpot	CL.C	Spotowa cena baryłki ropy naftowej notowana w USD
GOLD	XAUUSD	Cena jednej uncji złota wyrażona w USD
GOLDF	ZGG1	Cena kontraktów Futures na złoto wyrażona w USD za uncję złota
COPPERF	HG.F	Cena kontraktów Futures na miedź wyrażona w USD za funt miedzi
PPI1982US	-	Indeks wskaźnika cen dóbr produkcyjnych w USA (gdzie 1982 przedstawia wartość 100)
MACD(12,26)	-	Wskaźnik zbieżności i rozbieżności średnich ruchomych kursu NASDAQ100F (z 12 i 26 sesji)
Signal	-	Linia sygnałowa wykorzystywana do odczytywania informacji z MACD (z 9 sesji)
ULTOSC	-	Oscylator ultimate dla kursu NASDAQ100F
RSI	-	Wskaźnik siły względnej dla kursu NASDAQ100F
ROC15	-	Wskaźnik zmiany ROC z 15 sesji dla kursu NASDAQ100F
CCI(20;0,015)	-	Commodity channel index dla kursu NASDAQ100F
WILLIAMS%RANGE(14)	-	%R Williamsa dla kursu NASDAQ100F

Źródło: Opracowanie własne.

Dane dla zmiennych, przy których nie podano odnośników w Tabeli 2., zostały pobrane z serwisu stooq.pl. Po zebraniu wszystkich danych i poddaniu ich obróbce w opisany powyżej sposób, zbiór miał format *DataFrame* o kształcie: (4298, 57) – 57 zmiennych, w tym etykiety, czyli procentowa zmiana kursu indeksu na jeden dzień w przód (ale wyrażona w formie liczby pomnożonej przez 100 – to jest wzrost wartości o 2,81% w tabeli z danymi wejściowymi miał postać: 2,81, a nie 0,0281) dla 4298 dni sesyjnych.

Tabela 3. Fragment danych po wstępnej obróbce.

Date	FEDFunds	MNGDP	MRGDP	NASDAQ	NASDAQCV	NASDAQ10	NASDAQ100V	S&P500	S&P500V	Unemprate
2003-01-02 00:00:00	0,013	11147,64	13608,55	1365,89	322448256	1011,75	225884205	895,36	1365777825	0,058
2003-01-03 00:00:00	0,0112	11147,64	13608,55	1384,72	287505248	1028,05	201682459	908,255	1256444397	0,058
2003-01-06 00:00:00	0,0122	11147,64	13608,55	1405,755	392142752	1047,75	275033331	919,205	1595444527	0,058
2003-01-07 00:00:00	0,012	11147,64	13608,55	1427,915	438527264	1067,515	308028070	925,835	1716888841	0,058
2003-01-08 00:00:00	0,0129	11147,64	13608,55	1412,18	364952512	1053	255615798	915,72	1630666714	0,058
2003-01-09 00:00:00	0,0129	11147,64	13608,55	1426,465	422332992	1066,33	295787722	919,555	1733666702	0,058
2003-01-10 00:00:00	0,0125	11147,64	13608,55	1435,67	414156512	1073,61	289880702	926,27	1650444516	0,058
2003-01-13 00:00:00	0,0126	11147,64	13608,55	1453,885	400120256	1092,12	241410515	927,945	1551444480	0,058
2003-01-14 00:00:00	0,0124	11147,64	13608,55	1453,03	385463008	1088,62	233217550	928,655	1532666690	0,058
2003-01-15 00:00:00	0,013	11147,64	13608,55	1449,92	433902240	1085,28	265452643	925,055	1591222234	0,058
2003-01-16 00:00:00	0,0124	11147,64	13608,55	1432,155	394408736	1068,82	239745617	916,715	1705111135	0,058
2003-01-17 00:00:00	0,012	11147,64	13608,55	1388,78	409215488	1029,505	251550877	906,615	1509111087	0,058

Źródło: Opracowanie własne.

### 3.3. Opis wykorzystanych narzędzi

Wstępne przetwarzanie danych, wyliczenie wskaźników analizy technicznej i zebranie ich w jedną bazę zostało wykonane za pomocą programu Excel 2019. Tak stworzona baza danych została wczytana jako *DataFrame* do edytora języka Python przy użyciu biblioteki *Pandas*. Dalsze operacje na danych, takie jak uzupełnianie brakujących wartości, odbywały się przy użyciu języka Python i z wykorzystaniem takich bibliotek jak: *Pandas*, *NumPy*, *SciKitLearn*, *Math*, *Random*, *Collections*, *Seaborn* czy *Matplotlib*.

Modele sieci neuronowej zostały stworzone w języku programowania Python (w edytorze PyCharm) przy użyciu bibliotek: *Pandas*, *Numpy*, *Tensorflow* oraz *Keras*. *Keras* to interfejs biblioteki *Tensorflow*, który ułatwia korzystanie z niej poprzez minimalizowanie ilości akcji, jakie musi podjąć użytkownik.<sup>151</sup> *Tensorflow* to jedna z najczęściej stosowanych bibliotek w uczeniu maszynowym. Została opracowana przez firmę Google i udostępniona do powszechnego użytku w 2015 roku. Zdecydowano się na wybór tej biblioteki w związku z powszechnie uznaną prędkością jej działania oraz szeroką gamą możliwości, jakie oferuje użytkownikowi.<sup>152</sup> Wszystkie powyżej opisane biblioteki działają na licencji *open source*.

Do poszukiwania optymalnych hiperparametrów modelu wykorzystano biblioteki języka Python: *Hyperopt* oraz *Hyperas*. Pierwsza z nich wykorzystuje twierdzenie Bayesa do oszacowania najlepszych hiperparametrów (z podanego zakresu lub zmiennych dyskretnych), a druga jest

<sup>151</sup> <https://keras.io/>, 25.01.2021

<sup>152</sup> <https://www.tensorflow.org/about>, 25.01.2021

*wrapperem*, który ułatwia korzystania z *Hyperopt* i pozwala zastosować go do modelu zaprojektowanego w *Keras*.

### 3.4. Zmienne modelu

Przed przystąpieniem do tworzenia modelu sprawdzono korelację Pearsona pomiędzy zmiennymi objaśniającymi, a zmienną objaśnianą. W tym przypadku zmienną objaśnianą jest zmienna *future*, czyli zmiana procentowa na wartości indeksu dla kontraktu E-mini futures Nasdaq 100 dla dnia kolejnego. W związku z tym największą korelację dodatnią wykazuje ona ze zmianą procentową w danym dniu (*NASDAQ100FCHANGE*). Grupą wykazującą istotną korelację były także wskaźniki analizy technicznej (jako iż są liniowymi przekształceniami kursu akcji) - %R Williamsa, CCI, ROC, RSI, oscylator Ultimate. Mimo że do tej grupy należy także MACD i linia sygnałna, te dwie zmienne nie wykazywały istotnej korelacji ze zmienną objaśnianą. Oprócz powyższych, istotną korelację ( $\geq 0,05$  lub  $\leq -0,05$ ) wykazały tylko 4 zmienne: wolumen obrotu na indeksie DAX, bilans handlowy USA, wolumen obrotu na indeksie NASDAQ100 oraz wskaźnik inflacji CPI w USA.

Mimo pozornie niskiej korelacji wielu zmiennych ze zmienną objaśnianą, zdecydowano się nie usuwać z modelu żadnych zmiennych – korelacja Pearsona wykazuje jedynie liniowe zależności między zmiennymi, natomiast sieci neuronowe zdolne są do wychwytywania nieliniowych zależności, które mogą być istotne w przypadku tego modelu.

Tabela 4. Korelacja pomiędzy zmiennymi objaśniającymi a zmienną objaśnianą.

Zmienna	future	Zmienna	future	Zmienna	future
AUDUSD	-0,01	NASDAQ100	0	FTSE100	-0,01
CPIUS	-0,06	NASDAQ100V	-0,08	FTSE100V	-0,02
PPI1982US	-0,01	S&P500	0	SHCOMP	-0,02
CrudeOilF	-0,03	S&P500V	-0,04	HANGSENGIND	-0,02
CrudeOilFV	-0,01	Unemprate	0,02	NIKKEI225	-0,02
CrudeOilSpot	-0,03	USDCHF	0	NIKKEI225V	-0,01
DJC	0	USDCNY	-0,01	GOLD	0,01
DJCV	-0,03	USDEUR	0,02	GOLDF	0,01
DJI	0	USDGBP	0,03	COPPERF	-0,01
DJIV	-0,04	USDJPY	-0,01	MACD(12,26)	-0,01
DJT	0	VIX	-0,02	Signal	-0,02
DJTV	-0,03	CRB	-0,04	ULTOSC	0,15
DJU	-0,01	M1	0,01	RSI	0,19
DJUV	-0,04	HOUSESTART	0	ROC15	0,15
FEDFunds	-0,02	BALANCETRADE	0,05	CCI(20;0,015)	0,17
MNGDP	0	PCONSUMPEXPEND	0	WILLIAMS%RANGE(14)	0,26
MRGDP	0	INDPRO	-0,02	NASDAQ100FV	-0,11
NASDAQC	0	DAX	-0,01	NASDAQ100FCHANGE	0,42
NASDAQCV	-0,1	DAXV	-0,07		

Źródło: Opracowanie własne.

### 3.5. Proces budowania sieci neuronowej dla zadania regresji.

W celu stworzenia danych wejściowych modelu, najpierw podzielono przygotowaną wcześniej bazę danych na trzy zbiory – treningowy (85% obserwacji), walidacyjny (10% obserwacji) i testowy (5% najnowszych obserwacji). W związku z tym podzielone zbiory zawierały kolejno: 3654, 430 oraz 214 obserwacji. Następnie w każdym ze zbiorów wyodrębniono dwa podzbiory – X (zbiór zmiennych objaśniających) oraz Y (zmienna objaśniana). W X znajdują się wszystkie zmienne poza future, a Y składa się tylko z future. W wyniku podziału łączna liczba podzbiorów wynosiła 6 (trainX, trainY, devX, devY, testX, testY).

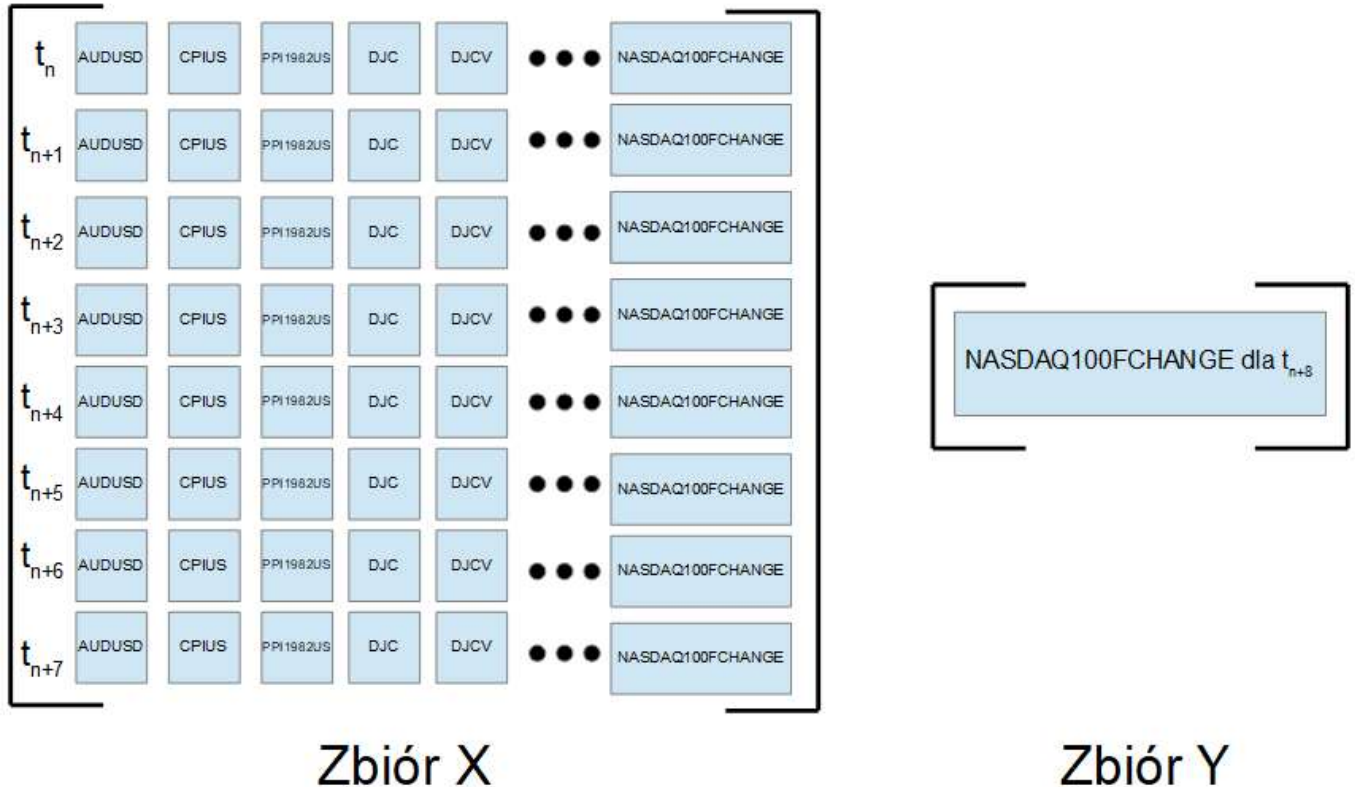
Zbiorów etykietowych nie poddawano dalszej obróbce – etykiety w zadaniu regresji powinny mieć oryginalną formę. Natomiast zbiory trainX, devX i testX poddano standaryzacji polegającej na zastosowaniu funkcji MinMaxScaler, która przekształca wartości wszystkich zmiennych tak, by mieściły się w przedziale [-1, 1]. Ten zabieg jest przeprowadzany dla każdej zmiennej z osobna. Wartości skrajne (którym przypisywane są wartości -1 dla minimalnej i 1 dla

maksymalnej) brane są ze zbioru treningowego, w związku z tym jeśli obserwacja dla danej zmiennej w zbiorze walidacyjnym i testowym jest wyższa niż maksymalna wartość tej zmiennej w zbiorze treningowym lub niższa niż minimalna wartość tej zmiennej w zbiorze treningowym, po ustandaryzowaniu może mieć wartość wyższą niż 1 lub niższą niż -1.

Standaryzacja przeprowadzona w ten sposób również jest korzystna dla sieci neuronowej, ponieważ zmienne, nawet jeśli mają wartości spoza ustalonego przedziału, wciąż znajdują się w tych samych rzędach wielkości, co jest korzystne z punktu widzenia uczenia sieci neuronowej. Natomiast zastosowanie wszystkich obserwacji ze wszystkich podzbiorów, by obserwacje rzeczywiście znajdowały się w przedziale  $[-1, 1]$  sprawiłoby, że informacje o rozkładzie zbioru testowego „przeciekłyby” do modelu, w związku z czym sprawdzanie jak dobrze model przewidyje na danych testowych, nie byłoby adekwatne, ponieważ model miałby już część informacji na temat tego zbioru.

Kolejnym elementem procesu było przygotowanie sekwencji danych wejściowych modelu. Dla każdego z podzbiorów z osobna przeprowadzono te same działania: utworzono kolejny zbiór danych składający się z obserwacji dla określonej liczby kroków czasowych (SEQ\_LEN) w postaci trójwymiarowego szyku danych biblioteki *Numpy*, o kształcie  $(56 \times \text{SEQ\_LEN} \times \text{liczba obserwacji w podzbiorze} - \text{SEQ\_LEN})$ , gdzie 56 to liczba zmiennych objaśniających, a SEQ\_LEN to zmienna kontrolująca długość okienka czasowego, w postaci którego dane mają trafiać do sieci. Następnie, do każdej takiej sekwencji  $56 \times \text{SEQ\_LEN}$  dopasowano etykietę, a więc procentową zmianę waloru, która nastąpi dzień po ostatnim dniu z danego okienka czasowego. Zbiory: uczący, walidacyjny i testowy składały się z tylu okienek widocznych na rysunku 22., ile wynosił wynik dla: liczby obserwacji w podzbiorze – SEQ\_LEN. Eksperymentowano z okienkami czasowymi różnej długości, ale najlepsze wyniki osiągnęto dla  $\text{SEQ\_LEN} = 8$ . W takim wypadku podczas jednej iteracji do sieci trafiało 8 zmiennych z jedną etykietą, a każda z 8 zmiennych to krok czasowy składający się z 56 zmiennych.

Rys. 22. Pojedyncza sekwencja danych wejściowych w postaci okienka czasowego dla  $SEQ\_LEN = 8$  wraz z jej etykietą.



Źródło: Opracowanie własne.

Szyki w powyższej formie stanowiły właściwe dane wejściowe modelu (treningowe dla stopniowej optymalizacji sieci, walidacyjne dla sprawdzenia, czy model dobrze generalizuje do nowych danych oraz testowe do ostatecznej weryfikacji generalizacji na nowych danych). Modele o najlepszych metrykach poddano na końcu testowi poprzez symulację zaaplikowania wygenerowanych przy ich pomocy sygnałów do kupna/sprzedaży prognozowanego waloru i sprawdzenie rentowności takich działań.

Glosariusz skrótów, zmiennych, metryk i hiperparametrów, które znajdują się w tym rozdziale i nie zostały zdefiniowane wcześniej znajduje się w załączniku A pracy. W załączniku B opisano, jakie wartości hiperparametrów wybrano dla poszczególnych modeli. Załącznik C stanowi kompletne zestawienie wyników dla trzech podejść uczenia każdego z modeli.

Korzystając z funkcji biblioteki Keras – Sequential(), ułożono pierwszą sieć rekurencyjną, której hiperparametry zostały wybrane arbitralnie (wzorując się na modelach w podobnych

zastosowaniach). Wybierano takie funkcje, które w generalnych zastosowaniach radzą sobie statystycznie najlepiej, żeby uzyskany w ten sposób wynik stanowił punkt odniesienia dla kolejnych wariantów.

Elementem wspólnym wszystkich stworzonych modeli było to, że przewidywały tylko jedną sesję w przód (w zadaniu regresji – zmiany procentowej na następny dzień, a w zadaniu klasyfikacji - czy kurs istotnie wzrośnie, istotnie zmaleje, czy też jego zmiana będzie nieistotna). Wszystkie modele otrzymują dane wejściowe w postaci dwuwymiarowej sekwencji (okienko czasowe), a ich wynikiem jest wektor (lub skalar).

Pierwszy model bazował na rozwiązaniach, które najczęściej sprawdzają się dla analizy szeregów czasowych. Stworzony został dla danych w postaci okienek czasowych z pięciu dni sesyjnych -pojedyncza sekwencja wejściowa miała więc 280 zmiennych. Zaletą tak zbudowanego zbioru treningowego jest to, że neurony LSTM rozpatrują zależności pomiędzy zmiennymi objaśniającymi a zmienną objaśnianą nie tylko dla tylu dni sesyjnych, z ilu składa się okienko czasowe. Są one również zdolne do zapamiętywania stanów pomiędzy poszczególnymi okienkami w obrębie jednej paczki danych treningowych (*batch*), dzięki czemu powinny lepiej odnajdywać zależności między danymi historycznymi, a przyszłymi wartościami walorów.

Pierwszy z ułożonych modeli miał 4 warstwy ukryte i warstwę wyjściową:

1. Warstwę LSTM z 224 neuronami,
2. Warstwę LSTM z 448 neuronami,
3. Warstwa w pełni połączona z 224 zwyczajnymi neuronami,
4. Warstwa w pełni połączona z 8 zwyczajnymi neuronami
5. Warstwa wyjściowa – jeden w pełni połączony neuron, którego wynik był jednocześnie wynikiem zadania regresji.

Posiłkowano się zaleceniami, żeby układać sieci tak, by liczby neuronów w poszczególnych warstwach były swoimi wielokrotnościami i żeby stopniowo redukować ilości neuronów w ostatnich warstwach. Inne parametry ustawiono w następujący sposób: rozmiar paczki danych treningowych (liczba sekwencji załadowana do sieci neuronowej przed aktualizacją wag) wynosiła 64, jako zalecana wartość startowa przy testowaniu sieci neuronowych. Wykorzystano domyślny wariant LSTM, który resetuje wagi po każdej paczki danych treningowych (*batch*). Liczbę epok



uczenia ustawiono na 100 (tyle razy wszystkie sekwencje treningowe powinny przejść przez sieć), z tym że ustawiono wcześniejsze przerywanie uczenia (*early stopping*) w przypadku gdy wartość funkcji straty dla zbioru walidacyjnego nie spada od 15 epok. Dzięki temu udaje się ograniczyć nadmierne dopasowanie modelu do danych treningowych. Jeśli model przerwie uczenie przed ostatnią epoką, przywracany jest zestaw wag, dla których wartość funkcji straty zbioru walidacyjnego była najniższa i na tych wagach model jest ostatecznie testowany.

Dzięki zastosowaniu trzech odrębnych zbiorów: treningowego, walidacyjnego i testowego udało się przeprowadzić podwójną weryfikację skuteczności sieci. Model ucząc się na danych treningowych, co epokę sprawdzał, jak dobrze generalizuje do danych ze zbioru walidacyjnego (które wcześniej nie były mu udostępnione). Natomiast po osiągnięciu optymalnej generalizacji dla nowych danych, do najlepiej sprawdzającego się modelu załadowano również dane wejściowe zbioru testowego, na które nie miał wcześniej żadnej ekspozycji. W związku z czym sprawdza się, jak model generalizowałby dla zastosowania w rzeczywistych warunkach i uzyskuje się trzy zestawy metryk (dla każdego ze zbiorów), dzięki którym można porównywać modele.

Nie zmieniano domyślnej budowy neuronów LSTM w modelu. Natomiast dla neuronów w warstwę w pełni połączonej zastosowano funkcję aktywacji ReLU, by wyniki neuronów nie były liniowymi przekształceniami danych wejściowych i wag oraz w celu uniknięcia eksplodujących/znikających gradientów (opisanych w 2. rozdziale pracy).

Minimalizowaną w procesie uczenia funkcją straty na bazie której wyliczano gradienty był błąd średniokwadratowy (MSE). Do iteracyjnej optymalizacji zadanej funkcji straty model korzystał z metody opartej na metodzie gradientu prostego – Adam (*Adaptive Moment Estimation*), który uznawany jest skuteczny w większości zastosowań, więc posłużył za punkt startowy. Współczynnik uczenia (*learning rate*) ustalono na poziomie  $lr = 0,001$ .

Oprócz metryki modelu, jaką jest funkcja straty, co epokę mierzono także średni błąd absolutny (MAE) oraz RMSE dla zbioru treningowego i walidacyjnego (w celu obserwowania postępów w uczeniu się sieci), a po zakończeniu uczenia, wszystkie trzy metryki sprawdzono także dla zbioru testowego. Zdecydowano się na użycie tych trzech miar ponieważ: MSE dobrze współpracuje z gradientowymi metodami optymalizacji oraz bardziej penalizuje większe błędy (podnosząc je do kwadratu), MAE ze względu na możliwość dosłowniejszej interpretacji (błędy

nie są podnoszone do kwadratu) i RMSE, by przewrócić miarę MSE do możliwości rzeczywistej interpretacji (pierwiastkując wynik, który wcześniej podniesiony był do kwadratu).

Funkcja ucząca (model.fit) biblioteki Keras ma argument *shuffle* (tasuj), który domyślnie jest aktywowany i w takiej konfiguracji przy treningu model dostaje dane wejściowe w postaci paczek danych uczących ułożonych w losowej kolejności – jest to pożądany zabieg przy większości sieci, aczkolwiek nie jest zalecany przy analizie szeregów czasowych, dlatego został zdezaktywowany, dzięki czemu dane trafiają do sieci chronologicznie.

W związku ze stochastyczną naturą algorytmów uczenia maszynowego, model o identycznych hiperparametrach będzie miał nieznacznie różniące się wyniki przy każdym uczeniu od zera (między innymi przez fakt losowej inicjalizacji wag). Dlatego weryfikacja wyników danego modelu wymaga więcej niż jednego podejścia. W przypadku tej pracy zdecydowano się na nauczanie wybranych modeli 3 razy i wybór najlepszego z nich do dalszych porównań.

Tabela 5. Wyniki modelu 1.

	Metryka/Zbiór	MSE	MAE	RMSE
I PRÓBA	Trening	0,358	0,435	0,598
	Walidacja	1,257	0,771	1,121
	Test	0,578	0,684	0,760
	Epoka	6		
II PRÓBA	Trening	0,516	0,512	0,718
	Walidacja	0,711	0,599	0,843
	Test	0,566	0,562	0,752
	Epoki	5		
III PRÓBA	Trening	1,015	0,728	1,007
	Walidacja	0,708	0,597	0,842
	Test	<b>0,555</b>	<b>0,559</b>	<b>0,745</b>
	Epoki	1		

Zródło: Opracowanie własne.

W przypadku pierwszego modelu najlepsze okazało się podejście trzecie, w którym model osiągnął najniższą funkcję straty dla zbioru walidacyjnego w pierwszej epoce (uczył się do piętnastej, w której *Early Stopping* zakończyło naukę). Sam fakt, że model osiągnął najlepszy rezultat w pierwszej epoce nie jest obiecujący. Przy RMSE na poziomie 0,745 i MAE na poziomie 0,5592 w zbiorze testowym nie można by zbudować działającego narzędzia wspomagającego

inwestowanie przy użyciu tego modelu. Takie wyniki metryk w uproszczeniu oznaczają, że przewidywana przez model wartość zmiany procentowej na następny dzień różniła się od rzeczywistej o 0,559 do 0,745 punktu procentowego, przy czym średnia z wartości bezwzględnych rzeczywistych zmian z dnia na dzień na NQ.F wynosiła 0,571%.

Poza mierzeniem metryk, napisany program generował również automatycznie arkusz Excel z przewidywanymi przez model wartościami zmiany na kolejny dzień sesyjny na wybranym walorze, rzeczywistymi zmianami na kolejny dzień sesyjny na wybranym walorze, wartością kontraktu futures e-mini na Nasdaq 100 z danego dnia oraz z sygnałem generowanym przez sieć (*Buy* jeśli przewidywana zmiana na kontrakcie jest wyższa niż 0,25%, *Sell* jeśli przewidywana zmiana na kontrakcie jest niższa niż -0,25% oraz *Hold* dla wartości pomiędzy -0,25 a 0,25%). Jak można zaobserwować na Tabeli 6., pierwszy model przewidywał podobne wartości dla całego okresu (niewielkie zmiany waloru), przez co w żadnym z dni sesyjnych nie przekroczył progu rekomendacji *Buy* lub *Sell*.

Tabela 6. Wyniki z pierwszych 12 dni sesyjnych dla najlepszego podejścia modelu 1.

Date	Predicted	Actual	IndexValue	Signal
09.01.2019	0,074	0,540	6575,75	Hold
10.01.2019	0,081	0,096	6611,25	Hold
11.01.2019	0,080	-0,689	6617,63	Hold
14.01.2019	0,081	0,576	6572,00	Hold
15.01.2019	0,079	0,915	6609,88	Hold
16.01.2019	0,082	0,369	6670,38	Hold
17.01.2019	0,085	0,937	6695,00	Hold
18.01.2019	0,093	-0,503	6757,75	Hold
22.01.2019	0,094	-0,969	6723,75	Hold
23.01.2019	0,092	0,156	6658,63	Hold
24.01.2019	0,095	1,033	6669,00	Hold
25.01.2019	0,084	0,076	6737,88	Hold

Źródło: Opracowanie własne.

W związku z niewielką skutecznością modelu w generalizowaniu do nowych danych, zdecydowano się zmienić trzy z hiperparametrów w modelu – wielkość paczki danych uczących (batch size) ustawiono na 32, czyli model częściej aktualizuje wagi i resetuje swój stan, zmieniono SEQ\_LEN na 7 (być może model lepiej dopasuje się do danych, mając do dyspozycji dłuższe okienko czasowe) oraz dodano dropout, czyli pomijanie wyników części neuronów. Użyto

ustandaryzowanych wartości dropout – 30% szans na pominięcie neuronów w warstwach LSTM oraz 20% szans na pominięcie neuronów w warstwach tradycyjnych (poza neuronem wyjściowym).

Tabela 7. Wyniki modelu 2. z okienkiem czasowym z 7 sesji.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,527	0,520	0,726
	Walidacja	0,583	0,550	0,764
	Test	0,512	0,550	0,715
	Epoka	22		
<b>II PRÓBA</b>	Trening	0,501	0,503	0,708
	Walidacja	0,564	0,512	0,751
	Test	0,506	0,541	0,712
	Epoki	29		
<b>III PRÓBA</b>	Trening	0,502	0,507	0,709
	Walidacja	0,495	0,482	0,703
	Test	<b>0,378</b>	<b>0,454</b>	<b>0,615</b>
	Epoki	25		

Źródło: Opracowanie własne.

Tabela 8. Wyniki z pierwszych 10 dni sesyjnych dla najlepszego podejścia modelu 2.

<b>Date</b>	<b>Predicted</b>	<b>Actual</b>	<b>IndexValue</b>	<b>Signal</b>
11.01.2019	0,080	-0,689	6617,63	Hold
14.01.2019	0,050	0,576	6572,00	Hold
15.01.2019	0,624	0,915	6609,88	Buy
16.01.2019	-0,207	0,369	6670,38	Hold
17.01.2019	0,536	0,937	6695,00	Buy
18.01.2019	0,230	-0,503	6757,75	Hold
22.01.2019	-0,305	-0,969	6723,75	Sell
23.01.2019	0,428	0,156	6658,63	Buy
24.01.2019	0,293	1,033	6669,00	Buy
25.01.2019	0,486	0,076	6737,88	Buy

Źródło: Opracowanie własne.

Na tabelach 7. oraz 8. widać wyraźną poprawę wyników względem pierwszego modelu, zwłaszcza przy trzeciej próbie uczenia. Choć odchylenia na poziomie 0,45 – 0,61 punktu procentowego wciąż nie są w pełni zadowalające, wyniki uzyskane w ten sposób mogłyby posłużyć już do zbudowania narzędzia wspomagającego inwestycje (co zostanie przetestowane w dalszej części pracy).

W związku z osiągnięciem lepszego wyniku, postanowiono sprawdzić, jak tasowanie paczek danych wejściowych (batch) składających się z siedmiosesyjnych okienek czasowych wpłynie na wyniki sieci. Zmieniono więc argument funkcji model.fit o nazwie shuffle na True, bez zmieniania reszty hiperparametrów. Źródło sugeruje, że model przewidujący wartości w szeregu czasowym powinien osiągać lepsze wyniki przy chronologicznym ułożeniu danych (nawet w modelu niezapamiętującym stanów pomiędzy paczkami danych wejściowych).<sup>153</sup>

Tabela 9. Wyniki modelu 3. z okienkiem czasowym z 7 sesji z tasowaniem paczki danych.

	Metryka/Zbiór	MSE	MAE	RMSE
<b>I PRÓBA</b>	Trening	0,550	0,524	0,741
	Walidacja	0,473	0,468	0,688
	Test	0,393	0,469	0,627
	Epoka	20		
<b>II PRÓBA</b>	Trening	0,512	0,508	0,716
	Walidacja	0,454	0,459	0,674
	Test	<b>0,359</b>	<b>0,444</b>	<b>0,599</b>
	Epoki	34		
<b>III PRÓBA</b>	Trening	0,580	0,531	0,762
	Walidacja	0,465	0,475	0,682
	Test	0,386	0,471	0,622
	Epoki	16		

Źródło: Opracowanie własne.

Tabela 10. Wyniki z pierwszych 10 dni sesyjnych dla najlepszego podejścia modelu 3.

Date	Predicted	Actual	IndexValue	Signal
11.01.2019	0,314	-0,689	6617,63	Buy
14.01.2019	0,238	0,576	6572,00	Hold
15.01.2019	0,987	0,915	6609,88	Buy
16.01.2019	0,222	0,369	6670,38	Hold
17.01.2019	0,640	0,937	6695,00	Buy
18.01.2019	0,554	-0,503	6757,75	Buy
22.01.2019	-0,691	-0,969	6723,75	Sell
23.01.2019	0,285	0,156	6658,63	Buy
24.01.2019	0,303	1,033	6669,00	Buy
25.01.2019	0,515	0,076	6737,88	Buy

Źródło: Opracowanie własne.

<sup>153</sup> <https://machinelearningmastery.com/stateful-stateless-lstm-time-series-forecasting-python/>, 09.02.2021

Wyniki dla modelu, gdzie paczki danych były przetasowane i podawane do modelu w niechronologicznej kolejności okazały się być lepsze (zarówno średni wynik dla trzech prób uczenia, jak i najlepszy wynik sieci) niż w przypadku chronologicznej kolejności sekwencji wejściowych. Może to jednak wynikać z małej próby porównawczej (3 podejścia do uczenia). W związku z tym, że jest to niezgodne z podanym wcześniej źródłem oraz sprzeczne z intuicją, wykorzystanie tasowania paczek danych zbioru treningowego zostanie poddane dalszym testom.

W związku ze znaczną poprawą wyników sieci po zwiększeniu SEQ\_LEN z 5 do 7, przeprowadzono serię eksperymentów, nauczając modele z coraz dłuższym okienkiem czasowym. Przetestowano modele z długością okienka na poziomie: 8, 9, 10, 11 oraz 14. Każdy z modeli nauczano trzy razy. Najlepszy wynik (we wszystkich trzech metrykach) osiągnął model z SEQ\_LEN = 8, co zaobserwować można na tabeli 11. oraz na tabeli 12. Średnie wyniki modeli stopniowo pogarszały się wraz z dalszym wydłużaniem okienka.

Tabela 11. Wyniki modelu 4. z okienkiem czasowym z 8 sesji (ceteris paribus).

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,595	0,551	0,771
	Walidacja	0,477	0,468	0,690
	Test	<b>0,338</b>	<b>0,424</b>	<b>0,582</b>
	Epoka	22		
<b>II PRÓBA</b>	Trening	0,554	0,525	0,744
	Walidacja	0,489	0,475	0,699
	Test	0,399	0,465	0,632
	Epoki	23		
<b>III PRÓBA</b>	Trening	0,546	0,522	0,739
	Walidacja	0,450	0,458	0,671
	Test	0,378	0,448	0,615
	Epoki	31		

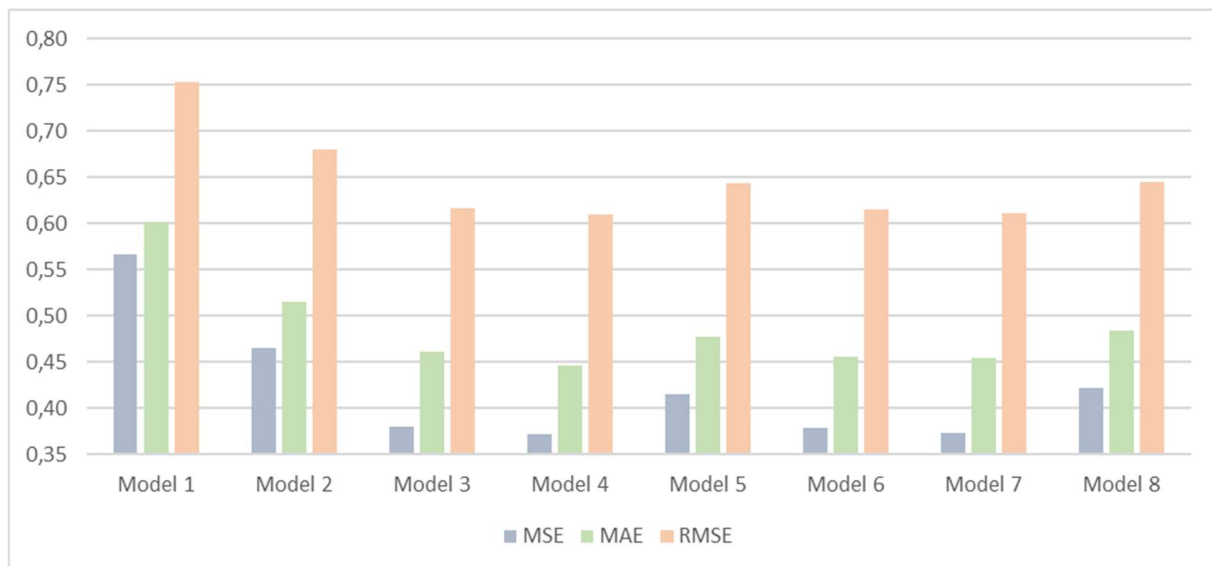
Źródło: Opracowanie własne.

Tabela 12. Wyniki z pierwszych 9 dni sesyjnych dla najlepszego podejścia modelu 4.

Date	Predicted	Actual	IndexValue	Signal
14.01.2019	0,252	0,576	6572,00	Buy
15.01.2019	1,246	0,915	6609,88	Buy
16.01.2019	0,047	0,369	6670,38	Hold
17.01.2019	0,804	0,937	6695,00	Buy
18.01.2019	0,443	-0,503	6757,75	Buy
22.01.2019	-0,572	-0,969	6723,75	Sell
23.01.2019	0,353	0,156	6658,63	Buy
24.01.2019	0,423	1,033	6669,00	Buy
25.01.2019	0,602	0,076	6737,88	Buy

Źródło: Opracowanie własne.

Rys. 23. Wykres metryk zbioru testowego dla średnich z trzech podejść uczenia.



Źródło: Opracowanie własne.

Po wyuczeniu 8 pierwszych modeli (każdy model został wytrenowany trzykrotnie), sporządzono zestawienie średnich wartości metryk dla zbioru testowego. Jak można zaobserwować na rysunku 23., najlepsze wyniki regresji dał model z danymi wejściowymi w postaci sekwencji składających się z danych z 8 dni sesyjnych. Ten sam model osiągnął również podejście z najniższymi wartościami metryk (MSE: 0,338; MAE: 0,424; RMSE: 0,582 Modele 3. – 8. Różnią się między sobą jedynie długością okienka czasowego (kolejno: 7, 8, 9, 10, 11, 14 dni sesyjnych).

Trening jednego wariantu odbywa się średnio w 40 epokach, a przeprowadzenie obliczeń w jednej epoce zajmuje 23 sekundy (na domowym komputerze stacjonarnym). Pojedynczy trening

trwa więc mniej więcej 15 minut. Przy założeniu, że każdy model należy przetestować co najmniej trzykrotnie, daje to 45 minut bez żadnej zmiany hiperparametrów. Nawet zakładając, że nie zmienia się ilości warstw w modelu ani rodzajów neuronów w warstwach, istnieje 13 hiperparametrów istotnie wpływających na wyniki modelu: długość sekwencji, wielkość paczki danych wejściowych, liczba epok, liczba neuronów w pierwszej, drugiej, trzeciej i czwartej warstwie, funkcje aktywacji w warstwach w pełni połączonych, rodzaj funkcji optymalizacyjnej, współczynnik uczenia, tasowanie paczek danych (binarny) oraz zastosowanie *dropoutu* (binarny, ale istnieje możliwość dobrania różnych współczynników będących szansami na pominięcie neuronów z zakresu  $(0,1)$ ). Niemożliwe byłoby przetestowanie wszystkich kombinacji powyższych parametrów poprzez nauczanie kolejnych sieci. W związku z tym zdecydowano się na przeprowadzenie optymalizacji hiperparametrów (znalezienie optymalnych poprzez zastosowanie zewnętrznego modelu) w celu polepszenia wyniku głównego modelu oraz zawężenia obszaru poszukiwań optymalnych rozwiązań. Rozważano trzy podejścia do optymalizacji hiperparametrów:

- *Grid search* – polega na wyszukiwaniu wyczerpującym w ręcznie zdefiniowanym obszarze hiperparametrów algorytmu uczącego, ale wymaga dużych nakładów obliczeniowych i daje gorsze wyniki w wielowymiarowych problemach.
- *Random Search* – wyszukiwanie optymalnych hiperparametrów spośród losowych wartości – sprawdza się w przypadku, gdy tylko niewielka ilość parametrów istotnie wpływa na wyniki modelu – w przypadku badanego modelu testy z ręcznym dostrajaniem hiperparametrów pokazały, że wiele hiperparametrów wpływa na wynik.
- Optymalizacja bayesowska – algorytm, który podejmuje próby oszacowania optymalizowanej funkcji na podstawie poznanych wcześniej wartości. Dziedziną jest zbiór hiperparametrów, a zbiorem wartości funkcja straty głównego modelu (MSE). Algorytm ten jest mniej obliczeniowo wymagający od poprzednich alternatyw oraz osiąga dobre wyniki w sytuacji, gdy funkcja ma wiele lokalnych optimów.<sup>154</sup> Z tych powodów zdecydowano się na wykorzystanie jej w tej pracy.

---

<sup>154</sup> <https://www.blog.bitpeak.pl/post/inteligentne-uczenie-sztucznej-inteligencji-czyli-optymalizacja-bayesowska>, 11.02.2021



Ze względu na zawilosc algorytmu, trudno jest przeprowadzić jego ręczną implementację, ale dzięki open sourcowym bibliotekom w Pythonie można przeprowadzić bayesowskie poszukiwanie optymalnych hiperparametrów w predefiniowanym zakresie lub w predefiniowanych punktach (dla poszczególnych hiperparametrów), bazując na modelu stworzonym w Kerasie. Przy użyciu Hiperas i Hyperopt przeprowadzono optymalizację wszystkich hiperparametrów poza:

- liczbą epok – wcześniej trenowane algorytmy nigdy nie przekroczyły setnej (ustawionej jako ostatniej) epoki, bo zawsze powstrzymywał je przed tym early stopping z cierpliwością ustawioną na 15 epok
- długością okienka czasowego sekwencji wejściowej – nie pozwalały na to ograniczenia bibliotek. Dodatkowo, jego optymalna długość została ustalona wcześniej na poziomie 8 dni sesyjnych poprzez wielokrotną zmianę tego hiperparametru przy ceteris paribus.

Model optymalizacji bayesowskiej rozpatrywał następujące wartości lub przedziały dla hiperparametrów:

- Liczba neuronów LSTM w pierwszej warstwie: 8, 16, 32, 56, 84, 112, 224, 336, 448
- Szansa na pominięcie neuronu w pierwszej warstwie: przedział (0; 0,6)
- Liczba neuronów LSTM w drugiej warstwie: 56, 84, 112, 168, 224, 336, 448, 512, 600
- Szansa na pominięcie neuronu w drugiej warstwie: przedział (0; 0,6)
- Liczba neuronów w trzeciej warstwie: 8, 16, 32, 56, 84, 112, 224, 336, 448
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU, Liniowa, Leaky ReLU
- Szansa na pominięcie neuronu w trzeciej warstwie: przedział (0; 0,6)
- Liczba neuronów w czwartej warstwie: 2, 4, 8, 16, 32
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU, Liniowa, Leaky ReLU
- Szansa na pominięcie neuronu w czwartej warstwie: przedział (0; 0,6)
- Wybór algorytmu optymalizacyjnego spośród: Adam, Adagrad, Adadelta, RMSprop
- Wybór współczynnika uczenia dla optymalizatorów: 0.0001, 0.001, 0.01, 0.1
- Wielkość paczek danych(*batch size*): 8, 16, 32, 64, 128, 256, 512
- Wybór czy paczki danych powinny być przetasowane przed uczeniem modelu: Tak, Nie

Pozostałe założenia i hiperparametry modelu pozostały niezmienione. Długość okienka czasowego sekwencji ze zbioru uczącego ustawiono na 8, model uczy się maksymalnie 110 epok, przy czym może przerwać uczenie wcześniej, jeśli główna metryka (funkcja straty dla zbioru walidacyjnego) nie odnotowała poprawy od 12 epok. W obrębie wyżej wymienionych przedziałów i wartości hiperparametrów, model optymalizacji bayesowskiej wybierał kombinacje, które następnie poddawał ocenie, przy czym maksymalna liczba iteracji została ustawiona na 70. Na koniec procesu model wypisał optymalne hiperparametry dla tak zdefiniowanych przedziałów:

- Liczba neuronów LSTM w pierwszej warstwie: 448
- Szansa na pominięcie neuronu w pierwszej warstwie: 0,1187
- Liczba neuronów LSTM w drugiej warstwie: 336
- Szansa na pominięcie neuronu w drugiej warstwie: 0,2022
- Liczba neuronów w trzeciej warstwie: 8
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: Leaky ReLU
- Szansa na pominięcie neuronu w trzeciej warstwie: 0,330
- Liczba neuronów w czwartej warstwie: 8
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: Liniowa
- Szansa na pominięcie neuronu w czwartej warstwie: 0,4417
- Wybór algorytmu optymalizacyjnego: RMSprop
- Wybór współczynnika uczenia dla optymalizatorów: 0,001
- Wielkość paczek danych (*batch size*): 32
- Wybór czy paczki danych powinny być przetasowane przed uczeniem modelu: Tak

Hiperparametry modelu dostosowano do powyższych wartości i przeprowadzono 3 testy – średnia wyników dla trzech podejść uczenia Modelu 9 była niższa (lepiej) niż dla wszystkich wcześniej badanych modeli w przypadku metryk MSE (o 2,7%) oraz RMSE (1,3%), a gorsza od najlepszej średniej w przypadku MAE (o 0,5%). Modelem dającym najlepszy jednostkowy wynik nadal pozostawał Model 4 (przed zastosowaniem bayesowskiej optymalizacji hiperparametrów). Zdecydowano się podjąć jeszcze jedną próbę optymalizacji hiperparametrów – tym razem z zawężonym zakresem w obszarze wcześniej wytypowanym przez bayesowską optymalizację jako najlepszy. Ustalono nowe przedziały:

- Liczba neuronów LSTM w pierwszej warstwie: 336, 352, 386, 412, 448, 480, 512
- Szansa na pominięcie neuronu w pierwszej warstwie: przedział (0,1; 0,3)
- Liczba neuronów LSTM w drugiej warstwie: 224, 264, 302, 320, 336, 368, 400
- Szansa na pominięcie neuronu w drugiej warstwie: przedział (0,15; 0,35)
- Liczba neuronów w trzeciej warstwie: 2, 4, 8, 16, 24, 32, 40, 56
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU, Liniowa, Leaky ReLU
- Szansa na pominięcie neuronu w trzeciej warstwie: przedział (0; 0,15)
- Liczba neuronów w czwartej warstwie: 4, 8, 10, 14, 16, 24, 32
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU, Liniowa, Leaky ReLU
- Szansa na pominięcie neuronu w czwartej warstwie: przedział (0,35; 0,5)
- Wybór współczynnika uczenia dla optymalizatorów: przedział (0,0008; 0,01)
- Wielkość paczek danych (*batch size*): 16, 20, 28, 32, 48, 56, 64
- Wybór czy paczki danych powinny być przetasowane przed uczeniem modelu: Tak, Nie

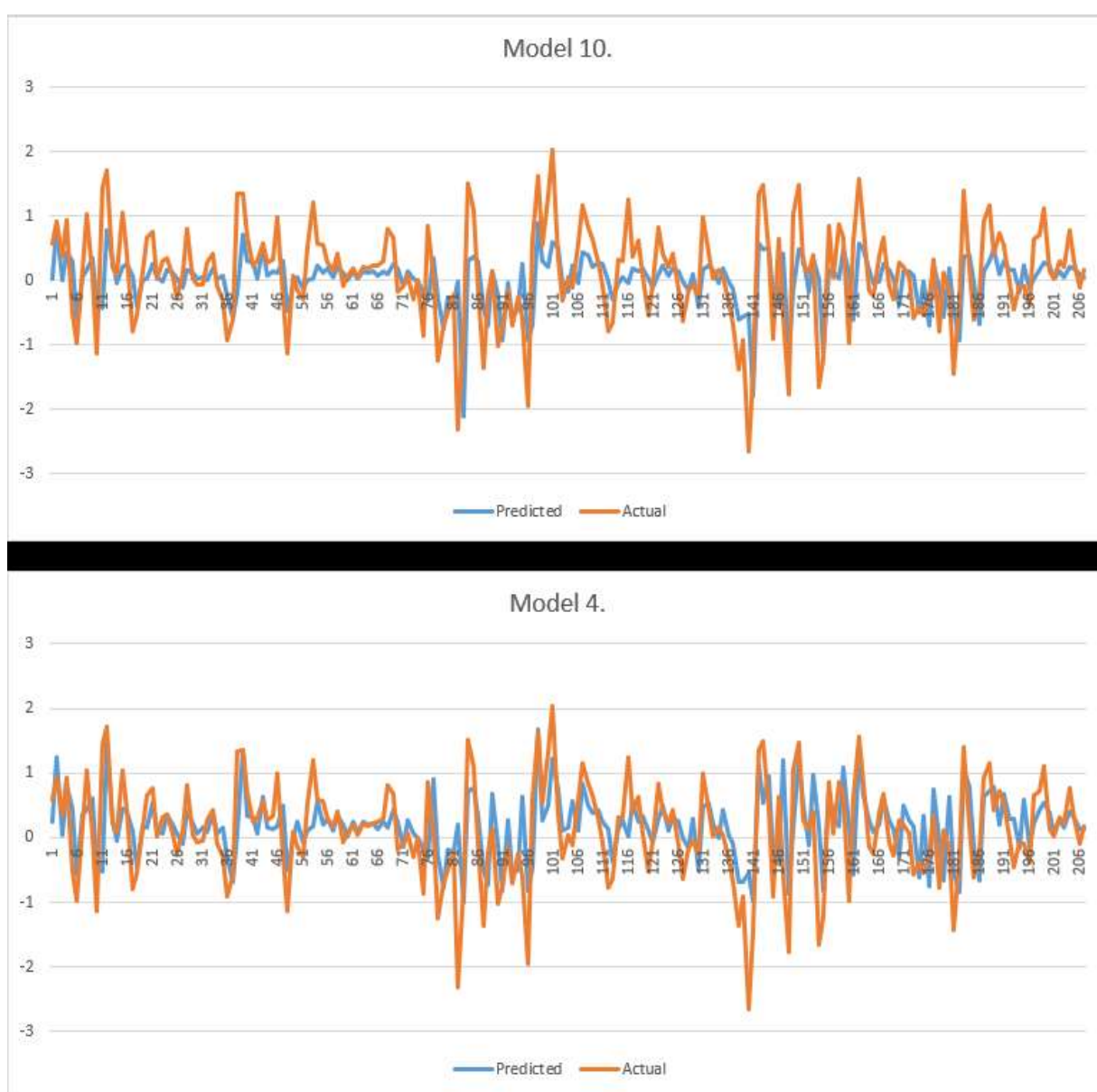
Optymalne hiperparametry w zawężonym przedziale, które model wybrał:

- Liczba neuronów LSTM w pierwszej warstwie: 480
- Szansa na pominięcie neuronu w pierwszej warstwie: 0,1135
- Liczba neuronów LSTM w drugiej warstwie: 302
- Szansa na pominięcie neuronu w drugiej warstwie: 0,3433
- Liczba neuronów w trzeciej warstwie: 32
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU
- Szansa na pominięcie neuronu w trzeciej warstwie: 0,0017
- Liczba neuronów w czwartej warstwie: 32
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: Liniowa
- Szansa na pominięcie neuronu w czwartej warstwie: 0,4966
- Wybór współczynnika uczenia dla optymalizatorów: 0,0029
- Wielkość paczek danych (*batch size*): 48
- Wybór czy paczki danych powinny być przetasowane przed uczeniem modelu: Tak

Zbudowano Model 10 w oparciu o powyższe parametry – najlepsza z trzech prób nauczania go wciąż dawała gorsze wyniki, niż wcześniej prezentowany model 4, ale średni wynik z trzech

prób i najlepszy wynik wśród nich dawały lepsze wyniki, niż poprzednio wytrenowany model (przed zawężeniem obszaru poszukiwania hiperparametrów). Uśrednione metryki dla zbioru testowego wynosiły kolejno: MSE 0,348, MAE 0,437, RMSE 0,590 – wyniki te były lepsze od poprzedniego modelu o około 3%, a od czwartego modelu (w którym wytrenowano najskuteczniejszy model) o około 4%. Najlepsze podejście pozostawało o 0,5% gorsze od najlepiej wytrenowanego do tego momentu modelu.

Rys. 24. Porównanie dwóch najlepszych modeli – wartości przewidziane przez model dla zbioru testowego (niebieska linia) i wartości rzeczywiste (pomarańczowa linia).



Źródło: Opracowanie własne.

### 3.6. Proces budowania sieci neuronowej dla zadania klasyfikacji

Rysunek 24. pokazuje, że sieci w zadaniu regresji wykazywały akceptowalną trafność w niektórych dniach sesyjnych, a w innych wydawały się być opóźnione względem wartości rzeczywistych. Budowanie narzędzia wspomagającego inwestycje na podstawie sieci neuronowej z zadaniem regresji wymaga od sieci bardzo precyzyjnych wyników, by podpowiedzi modelu były trafne. Jedną z metod mogących usprawnić działanie takiego narzędzia (dzięki zmniejszeniu koniecznej dokładności modelu) jest zastosowanie pseudoklasyfikacji.

Wstępny proces obróbki danych pozostał identyczny do przeprowadzonego w modelu z zadaniem regresji – pierwszym różniącym się krokiem było utworzenie dodatkowej kolumny w zbiorze danych, o nazwie FSignal – jeżeli kurs następnego dnia sesyjnego wzrasta o więcej niż 0,25 procenta, FSignal przyjmuje wartość 0 (potencjalny sygnał Kupuj), jeśli spadnie o więcej niż 0,25 procenta, FSignal przyjmuje wartość 1 (potencjalny sygnał Sprzedaj), jeśli zmiana wartości kontraktu mieści się w przedziale  $(-0,25; 0,25)$ , FSignal przyjmuje wartość 2 (potencjalny sygnał Trzymaj). Zdecydowano się na taki podział, by uwzględnić koszty transakcyjne, którymi obarczeni są inwestorzy oraz by model nie reagował zbyt pochopnie na małe zmiany w kursie kontraktu. Po utworzeniu kolumny FSignal, usunięto kolumnę zawierającą zmianę wartości kontraktu jeden dzień do przodu (future), by nie podpowiadała modelowi właściwej klasyfikacji.

Zmienne objaśniające były tożsame z tymi w modelu z zadaniem regresji, ale zmieniły się zmienne objaśniane – w modelu klasyfikacji była to kolumna FSignal, czyli informacja o tym, czy kurs następnego dnia istotnie wzrośnie, istotnie spadnie, czy zostanie na podobnym poziomie. Finalnie model powinien przewidywać, którą z tych trzech klas przypisać nowym danym.

W pierwszym podejściu wypróbowano model, który osiągnął najlepszą średnią wyników w metrykach dla zbioru testowego w zadaniu regresji – jedyną różnicą była warstwa wyjściowa, gdzie zamiast pojedynczego w pełni połączonego neuronu (bez funkcji aktywacji), który dawał ostateczny wynik regresji, zastosowano trzyneuronową warstwę z funkcją aktywacji *Softmax*. Funkcja *Softmax* określa, jaka jest szansa na to, że nastąpi jedno z możliwych zdarzeń, określonych w zmiennych objaśniających (w tym przypadku określi jaka jest szansa, że etykieta powinna być zerem (Kupuj), jedynką (Sprzedaj) lub dwójką (Trzymaj)). Każda klasa otrzymuje wartość prawdopodobieństwa (suma wartości dla wszystkich klas to zawsze 1), a klasa o najwyższym prawdopodobieństwie wystąpienia, zostaje wybrana jako ostateczna dana wyjściowa.

Do oceny skuteczności modeli wykorzystano metryki: *sparse categorical cross entropy* (jednocześnie funkcja straty) oraz dokładność. Ta pierwsza jest miarą różnicy pomiędzy dwoma rozkładami prawdopodobieństwa dla danego zestawu zdarzeń, a ta druga określa procent poprawnie sklasyfikowanych wektorów w stosunku do całkowitej ich liczby.

Tabela 13. Wyniki modelu klasyfikacji (ModelC 1.) zbudowanego z takimi hiperparametrami jak Model 10.

	<b>Metryka/Zbiór</b>	<b>Sparse Categorical Cross Entropy</b>	<b>Accuracy</b>
<b>I PRÓBA</b>	Trening	0,894	0,593
	Walidacja	0,852	0,601
	Test	0,882	<b>0,594</b>
	Epoka	13	
<b>II PRÓBA</b>	Trening	0,922	0,575
	Walidacja	0,894	0,612
	Test	0,900	0,594
	Epoki	11	
<b>III PRÓBA</b>	Trening	0,911	0,572
	Walidacja	0,852	0,629
	Test	<b>0,877</b>	0,585
	Epoki	9	

Źródło: Opracowanie własne.

Tabela 14. Wyniki z pierwszych 9 dni sesyjnych dla najlepszego podejścia ModelC 1.

<b>Date</b>	<b>Predicted</b>	<b>Actual</b>	<b>PercentDiff</b>	<b>IndexValue</b>
14.01.2019	Hold	Buy	0,576	6572,00
15.01.2019	Buy	Buy	0,915	6609,88
16.01.2019	Hold	Buy	0,369	6670,38
17.01.2019	Buy	Buy	0,937	6695,00
18.01.2019	Buy	Sell	-0,503	6757,75
22.01.2019	Sell	Sell	-0,969	6723,75
23.01.2019	Hold	Hold	0,156	6658,63
24.01.2019	Buy	Buy	1,033	6669,00
25.01.2019	Buy	Hold	0,076	6737,88

Źródło: Opracowanie własne.

Tabele 13. i 14. pokazują wyniki wyżej opisanego modelu – w swoim najlepszym podejściu osiągnął on dokładność na poziomie 59,42% na zbiorze testowym. Błędna klasyfikacja sygnału Sprzedaj jako Trzymaj (w sytuacji w której walor nie jest w portfelu) lub Kupuj jako Trzymaj (w sytuacji w której walor jest w portfelu) nie będzie błędą podpowiedzią inwestycyjną, więc

rzeczywista skuteczność sygnałów przewidywanych przez sieć będzie wyższa, niż wynika to z metryk. Każdy wynik powyżej 50% oznacza, że model może być skutecznym narzędziem we wspomaganiu inwestora. Natomiast wynik może być zaburzony tym, że w okresie testowym prawidłowe klasy nie rozkładały się równomiernie. Sygnał Kupuj wystąpił 87 razy, sygnał Sprzedaj 54 razy, a sygnał Trzymaj 66 razy.

Drugi model klasyfikacji, jaki poddano testom, bazował na hiperparametrach, przy których model z zadaniem regresji osiągnął najlepszy wynik (Model 4.). Jednak w przypadku zadania klasyfikacji żadna z trzech prób nauczania tego modelu nie okazała się lepsza modelu z wcześniejszymi hiperparametrami, a średnia dla metryk zbioru testowego okazała się gorsza o 2,51% dla *Sparse Categorical Cross Entropy* oraz o 2,72% dla dokładności.

W celu próby dalszego polepszenia wyniku modelu klasyfikacji, zdecydowano się zastosować bayesowską optymalizację hiperparametrów (korzystając z tych samych wyborów/przedziałów w wartościach hiperparametrów, co przy pierwszym podejściu do takiej optymalizacji w zadaniu regresji). Na koniec procesu model wypisał optymalne hiperparametry dla tak zdefiniowanych przedziałów:

- Liczba neuronów LSTM w pierwszej warstwie: 336
- Szansa na pominięcie neuronu w pierwszej warstwie: 0,0998
- Liczba neuronów LSTM w drugiej warstwie: 84
- Szansa na pominięcie neuronu w drugiej warstwie: 0,0283
- Liczba neuronów w trzeciej warstwie: 56
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU
- Szansa na pominięcie neuronu w trzeciej warstwie: 0,2795
- Liczba neuronów w czwartej warstwie: 16
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU
- Szansa na pominięcie neuronu w czwartej warstwie: 0,3778
- Wybór algorytmu optymalizacyjnego: RMSprop
- Wybór współczynnika uczenia dla optymalizatorów: 0,001
- Wielkość paczek danych (*batch size*): 32
- Wybór czy paczki danych powinny być przetasowane przed uczeniem modelu: Tak

Po dostosowaniu hiperparametrów do powyższych wartości i wytrenowaniu modelu trzykrotnie, okazało się, że dzięki osobnej optymalizacji hiperparametrów dla zadania klasyfikacji, udało się usprawnić zarówno średnią metryk z trzech treningów dla zbioru testowego, jak i wytrenować najlepszy do tej pory model. Średnia dla metryk zbioru testowego wyniosła: SCCE 0,875; Dokładność 0,607 (poprawa o około 4% w przypadku pierwszej metryki i 6% w przypadku drugiej względem poprzedniego modelu). Najlepszy model cechował się dokładnością przewidywania klasy na poziomie 62,63%. Zdecydowano się wykorzystać taką samą procedurę usprawniania modelu jak w przypadku zadania regresji i przeprowadzono kolejną optymalizację hiperparametrów na zawężonych przedziałach:

- Liczba neuronów LSTM w pierwszej warstwie: 224, 262, 286, 304, 336, 352, 364
- Szansa na pominięcie neuronu w pierwszej warstwie: przedział (0,05; 0,15)
- Liczba neuronów LSTM w drugiej warstwie: 56, 68, 72, 84, 96, 108, 112
- Szansa na pominięcie neuronu w drugiej warstwie: przedział (0; 0,1)
- Liczba neuronów w trzeciej warstwie: 32, 44, 50, 56, 62, 70, 84
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU, Liniowa, Leaky ReLU
- Szansa na pominięcie neuronu w trzeciej warstwie: przedział (0,2; 0,35)
- Liczba neuronów w czwartej warstwie: 8, 12, 16, 20, 24, 32
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: ReLU, Liniowa, Leaky ReLU
- Szansa na pominięcie neuronu w czwartej warstwie: przedział (0,3; 0,45)
- Wybór współczynnika uczenia dla optymalizatorów: przedział (0,0008; 0,01)
- Wielkość paczek danych(*batch size*): 16, 24, 32, 40, 46, 52
- Wybór czy paczki danych powinny być przetasowane przed uczeniem modelu: Tak, Nie

Optymalizacja bayesowska wybrała następujące wartości hiperparametrów jako optymalne:

- Liczba neuronów LSTM w pierwszej warstwie: 336
- Szansa na pominięcie neuronu w pierwszej warstwie: 0,0867
- Liczba neuronów LSTM w drugiej warstwie: 96
- Szansa na pominięcie neuronu w drugiej warstwie: 0,0436
- Liczba neuronów w trzeciej warstwie: 70



- Wybór funkcji aktywacji neuronów w trzeciej warstwie: Leaky ReLU
- Szansa na pominięcie neuronu w trzeciej warstwie: 0,3254
- Liczba neuronów w czwartej warstwie: 24
- Wybór funkcji aktywacji neuronów w trzeciej warstwie: Liniowa
- Szansa na pominięcie neuronu w czwartej warstwie: 0,3746
- Wybór współczynnika uczenia dla optymalizatorów: 0,0058
- Wielkość paczek danych(*batch size*): 46
- Wybór czy paczki danych powinny być przetasowane przed uczeniem modelu: Tak

Wytrenowany model o takich hiperparametrach osiągnął o około 1% gorsze wyniki w średniej dla metryk zbioru testowego przy trzech podejściach uczenia, nie przyniósł więc oczekiwanej poprawy rezultatów.

Tabela 15. Podsumowanie metryk dla zbioru testowego modeli (średnich i najlepszych podejść).

Średnia metryk zbioru testowego dla modelu regresji										
Model/Metryka	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
MSE	0,566	0,465	0,379	0,372	0,416	0,378	0,374	0,422	0,362	<b>0,348</b>
MAE	0,602	0,515	0,461	0,446	0,478	0,456	0,454	0,484	0,448	<b>0,437</b>
RMSE	0,753	0,681	0,616	0,610	0,644	0,615	0,611	0,645	0,602	<b>0,590</b>
Najlepszy wynik w trzech podejściach dla modelu regresji										
Model/Metryka	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
MSE	0,555	0,378	0,359	<b>0,338</b>	0,394	0,360	0,365	0,346	0,349	0,340
MAE	0,559	0,454	0,444	<b>0,424</b>	0,460	0,449	0,449	0,444	0,441	0,429
RMSE	0,745	0,615	0,599	<b>0,582</b>	0,628	0,600	0,605	0,588	0,591	0,583

Średnia metryk zbioru testowego dla modelu klasyfikacji				
Model/Metryka	ModelC 1	ModelC 2	ModelC 3	ModelC 4
SCCE	0,886	0,909	<b>0,875</b>	0,890
Dokładność	0,591	0,575	<b>0,607</b>	0,604
Najlepszy wynik w trzech podejściach klasyfikacja				
Model/Metryka	ModelC 1	ModelC 2	ModelC 3	ModelC 4
SCCE	0,882	0,907	<b>0,859</b>	0,896
Dokładność	0,594	0,585	<b>0,623</b>	0,614

Źródło: Opracowanie własne.

Tabela 15. przedstawia pełne wyniki dla średnich metryk wyliczonych dla zbiorów testowych oraz najlepszych z trzech prób uczenia dla wszystkich modeli. W przypadku modelu regresji nie miała miejsce sytuacja, w której różne próby uczenia osiągałyby lepsze wyniki dla innych metryk. W przypadku modelu klasyfikacji zdarzało się, że jedno z podejść uczenia danego

modelu osiągało lepszy wynik dla metryki SCCE, a inne podejście osiągało lepszy wynik dla metryki dokładność. W takim wypadku za lepszy z modeli wybierano ten, który osiągał wyższą dokładność, jako że to metryka łatwiejsza w bezpośredniej interpretacji. W metrykach MSE, MAE, RMSE oraz SCCE niższy wynik oznacza lepsze dopasowanie modelu do zbioru testowego, w przypadku dokładności – wyższy wynik oznacza lepsze dopasowanie. Z wytrenowanych modeli wybrano 4 najlepsze (2 dla zadania regresji i 2 dla zadania klasyfikacji), żeby przeprowadzić symulację skuteczności inwestycji w przypadku kierowania się sygnałami wyznaczonymi przez wyniki sieci.

### 3.7. Weryfikacja sygnałów inwestycyjnych modeli

Do weryfikacji wybrano najlepsze podejście uczenia modeli: Model 4., Model 10., ModelC 3., ModelC 4. Wszystkie weryfikacje rentowności inwestycji oparto na założeniach, że: można kupić część kontraktu (mniej niż jedną jednostkę) oraz że inwestor kupuje i sprzedaje kontrakt po uśrednionej cenie w ciągu danego dnia (średnia z kursu otwarcia i kursu zamknięcia). Przed sprawdzeniem rentowności inwestycji w kontrakt posiłkując się sygnałami modeli, sprawdzono trzy punkty odniesienia:

- Ref 1 - inwestor dysponuje kwotą 1000 dolarów, kupuje za tę kwotę część kontraktu (1000\$ podzielone na obecną cenę kontraktu) i każdego kolejnego dnia sesyjnego rozliczane są jego zyski i straty, ale kontrakt jest w jego portfelu od początku do końca okresu testowego (marże utrzymania i prowizje są pomijane). Postępując w ten sposób inwestor w całym okresie zarobiłby 4977,56 dolarów (nie wliczając zainwestowanego 1000), ponieważ w testowanym okresie kurs istotnie wzrastał.
- Ref 2 - ile inwestor zarobiłby, gdyby zawsze kupował (utrzymywał) kontrakt, jeśli następnego dnia ma istotnie wzrosnąć jego wartość, a sprzedawałby, jeśli następnego dnia jego wartość miałaby istotnie spaść. Zakładając, że inwestor dysponuje kwotą 1000\$ i nie reinwestuje zarabianych w tym czasie środków, jego zysk na koniec okresu testowego (nie wliczając zainwestowanego 1000) wyniósłby 13885,61\$.
- Ref 3 – ile zarobiłby inwestor opierając się na losowo wygenerowanych sygnałach (losowanie liczb całkowitych z przedziału (0;2), gdzie 0 oznaczało Kupuj, 1 Sprzedaj a 2 Trzymaj). Kierowanie się losowymi sygnałami dało wynik ujemny – inwestor straciłby 341,57 dolara.

Można więc wnioskować, że jeśli sygnały wygenerowane przez sieć pozwoliłyby osiągnąć inwestorowi zysk większy niż 4977,56 dolara, to byłaby ona dobrym narzędziem wspomagającym lokowanie kapitału.

W związku z tym, że założono, iż inwestor dysponuje kwotą 1000 dolarów, liczba jednostek kontraktu została policzona jako  $1000 / \text{cenę kontraktu na dzień, w którym sieć generuje sygnał kupna}$ . Rozliczenia w każdym kolejnym dniu dokonuje się poprzez obliczenie różnicy pomiędzy średnią wartością indeksu z dnia  $t$  i dnia  $t-1$  i przemnożenie jej przez współczynnik kontraktu wynoszący 20 oraz przemnożenie przez posiadaną liczbę jednostek kontraktu (zawsze mniej niż 1, ponieważ cena kontraktu w badanym okresie nie spadła poniżej 1000 dolarów), co można przedstawić wzorem:

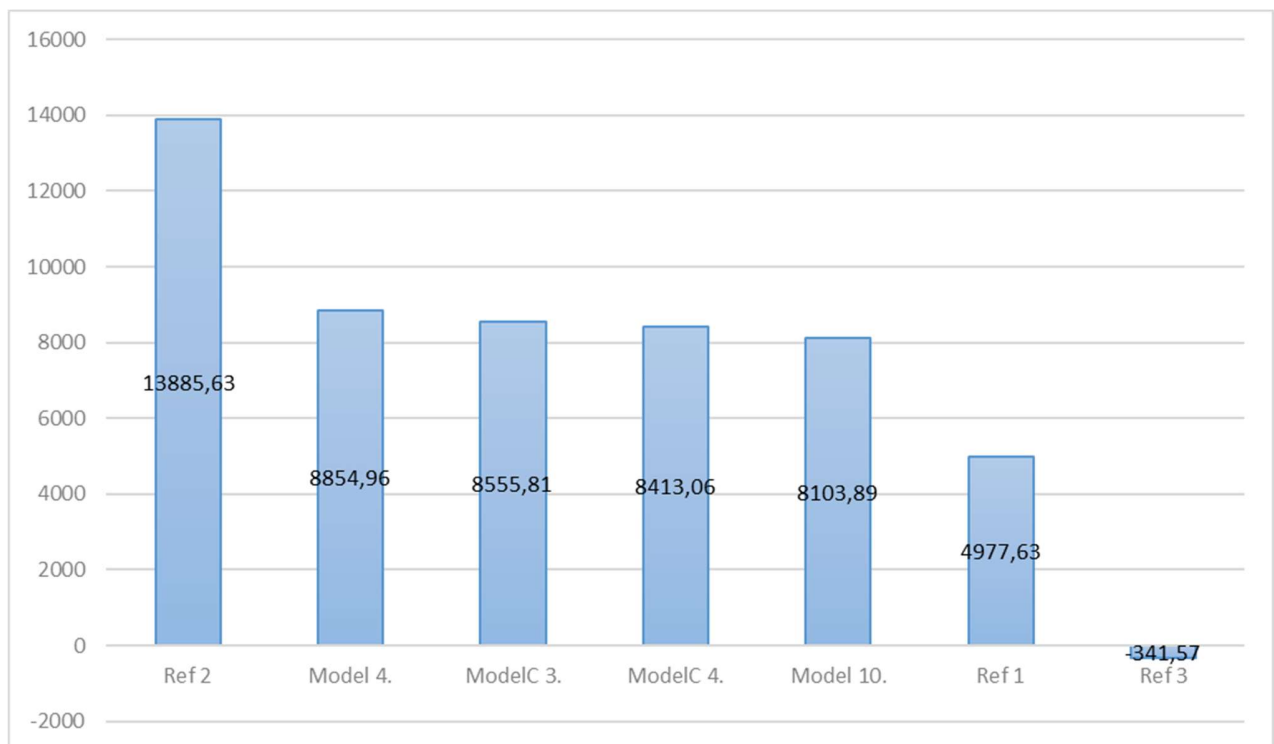
$$Z = (WI_t - WI_{t-1}) * 20 * 1000 / WI_{tk} \quad (3.1)$$

gdzie  $Z$  to zysk(strata) w danym dniu sesyjnym,  $WI_t$  to wartość indeksu w danym dniu sesyjnym,  $WI_{t-1}$  to wartość indeksu w poprzednim dniu sesyjnym, a  $WI_{tk}$  to wartość indeksu w dniu zakupu kontraktu. Założono, że jeśli w przypadku strat inwestor będzie miał mniej niż początkowe 1000\$, ulokuje dodatkowy kapitał tak, by znów inwestować pełną kwotę początkową.

Skrypt napisany w Pythonie automatycznie dokonywał powyższych obliczeń i sumował rozliczenia dzienne (dla dni w których indeks był w portfelu inwestora), by na końcu wypisać sumę zarobionych (straconych) pieniędzy. Wszystkie opłaty i prowizje zostały pominięte, a od zarobionych pieniędzy inwestor w Polsce musiałby zapłacić 19% podatku.

W przypadku modeli regresji skrypt najpierw nadawał etykiety sygnałowe – Kupuj, jeśli sieć przewidywała, że następnego dnia sesyjnego kurs wzrośnie o więcej niż 0,25% (w porównaniu z dniem bieżącym), Sprzedaj, jeśli sieć przewidywała, że następnego dnia sesyjnego kurs spadnie o więcej niż 0,25% i Trzymaj dla wartości pomiędzy z przedziału  $(-0,25\%; 0,25\%)$ . Zdecydowano się na takie przedziały, by uwzględnić konieczność pokrywania prowizji oraz uodpornić skrypt na nieznaczne wahania. Model klasyfikacyjny działa podobnie – etykiety utworzone na etapie obróbki danych (przed trenowaniem modeli) bazowały na tych samych przedziałach zmian procentowych. W związku z tym, jeśli model z zadaniem klasyfikacji przewidzi, że inwestor powinien sprzedać kurs, znaczy to, że model spodziewa się spadku wartości kursu o więcej niż 0,25% (i analogicznie w przypadku Kupuj oraz Trzymaj).

Rys. 25. Zysk osiągnięty przez modele o najlepszych wynikach metryk.



Źródło: Opracowanie własne.

Zgodnie z rysunkiem 25., najlepszy wynik osiągnął Model 4. (zadanie regresji) z wynikiem 8854,99\$, co sprawia, że inwestor posługujący się sygnałami wykreowanymi przez ten model, osiągnąłby o 78% większy zysk niż będąc w posiadaniu kontraktu przez cały okres testowy. Dodatkowa weryfikacja wykazała, że wśród wszystkich wytrenowanych modeli najlepszy rezultat finansowy osiągnąłby ModelC 1. (pierwsze podejście treningowe), dzięki którego sygnałom można było osiągnąć zysk na poziomie 9241,99\$ (mimo tego iż jego metryki były gorsze niż ModelC 3. i ModelC 4.).

Poza całkowitym zyskiem wygenerowanym przez strategie inwestycyjne (w przypadku inwestowania 1000\$), policzono także średnią dzienną stopę zwrotu ważoną ryzykiem. Za miarę ryzyka obrano odchylenie standardowe dziennych stóp zwrotu, a finalna miara powstała poprzez podzielenie średniego dziennego zwrotu przez odchylenie standardowe dziennych stóp zwrotu. Dzięki temu uzyskano dodatkową miarę efektywności modeli, dodatkowo premiującą strategie o niższym ryzyku. Należy jednak zwrócić uwagę na to, że w związku z tym, że strategia inwestycyjna układana jest w oparciu o jeden walor, w przypadku gdy generowany jest sygnał sprzedaj, mogą

pojawiać się okresy, gdy kontraktu nie ma w portfelu, a więc stopa zwrotu w kolejnych dniach wynosi 0.

Tabela 16. Średnie dzienne stopy zwrotu ważone ryzykiem.

Strategia	Średnia dzienna stopa zwrotu (%)	Odchylenie standardowe dziennej stopy zwrotu	Średnia dzienna stopa zwrotu ważona ryzykiem (%)
<b>Ref 1</b>	1,02	0,055	18,55
<b>Ref 2</b>	1,36	0,028	48,64
<b>Model 4.</b>	1,16	0,031	36,83
<b>Model 10.</b>	1,14	0,035	32,19
<b>ModelC 3.</b>	1,15	0,033	34,94
<b>ModelC 4.</b>	1,15	0,033	34,54

Zródło: Opracowanie własne.

Tabela 16. Prezentuje wyniki porównania średnich dziennych stóp zwrotu ważonych ryzykiem dla badanych strategii. Strategia kupuj i trzymaj osiągnęła najgorszy wynik, ponieważ walor trzymany był również w okresie spadków, więc odchylenie w tej próbie było znacznie wyższe, niż w pozostałych przypadkach. Najwyżej premiovana była strategia referencyjna, w której inwestor podejmuje optymalne decyzje za każdym razem (zakładając, że sygnał kupna generowany jest tylko w przypadku bezwzględnej zmiany wartości waloru bazowego większej niż 0,25%). W jej przypadku odchylenie było najniższe, a średnia dzienna stopa zwrotu najwyższa. Wśród strategii opartych o sygnały generowany przez model wyniki były zbliżone, ale najlepszy rezultat osiągnął Model 4., który miał najwyższą stopę zwrotu przy najniższej stopie ryzyka.

Kolejną miarą skuteczności porównywanych strategii może być liczba trafnych i błędnych sygnałów przez nie wygenerowanych oraz liczba trafnych i błędnych sygnałów występujących po sobie. Takie statystyki dają możliwość zobrazowania, w ilu przypadkach strategia będzie działała prawidłowo oraz jaka jest szansa na dłużej trwający cykl straty. W przypadku strategii referencyjnych przyjęto następujące założenia: Ref 1 – pierwszego dnia strategia przyjęła sygnał Kupuj, ostatniego dnia sygnał Sprzedaj, a we wszystkie pozostałe dni sygnał Trzymaj; Ref 2 – strategia zawsze przyjmuje właściwy sygnał.

Tabela 17. Liczba trafnych i błędnych sygnałów wygenerowanych przez strategię.

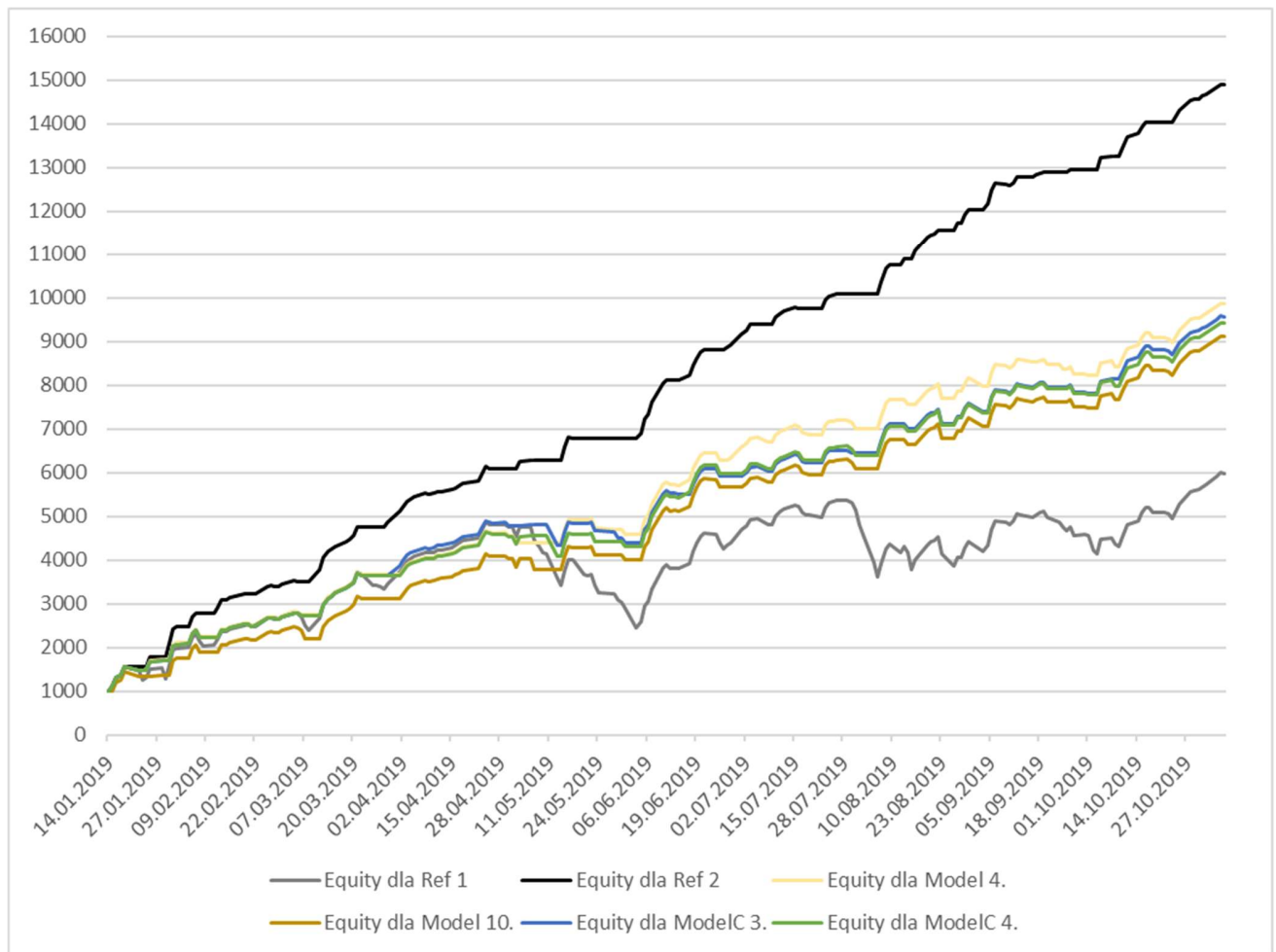
Strategia	Liczba trafnych sygnałów	Liczba błędnych sygnałów	Liczba trafnych sygnałów kolejnych po sobie	Liczba błędnych wskazań kolejnych po sobie
<b>Ref 1</b>	66	141	8	16
<b>Ref 2</b>	207	0	207	0
<b>Model 4.</b>	123	84	8	6
<b>Model 10.</b>	120	87	8	4
<b>ModelC 3.</b>	129	78	10	6
<b>ModelC 4.</b>	127	80	8	4

Źródło: Opracowanie własne.

Wyniki widoczne na Tabeli 17. pokazują, że najwięcej trafnych sygnałów wygenerował ModelC 3. To również on ma najwięcej trafnych sygnałów występujących po sobie – 10 (wobec 8 we wszystkich pozostałych modelach). W przypadku każdego z modeli liczba trafnych sygnałów znacznie przeważa liczbę błędnych sygnałów. Istotnym czynnikiem, o którym należy wspomnieć, jest to że strategię miały za zadanie zidentyfikować trzy klasy – Kupuj, Sprzedaj i Trzymaj. W niektórych przypadkach błędna klasyfikacja niekoniecznie będzie oznaczała stratę (na przykład w danym momencie w strategii kontrakt nie znajduje się w portfelu, a model wygeneruje sygnał Trzymaj zamiast Sprzedaj, to nie będzie to miało wpływu na kapitał inwestora).

Efektywność prezentowanych wyżej strategii inwestycyjnych można zobrazować także za pomocą wykresu linii equity, która pokazuje zmiany w kapitale na przestrzeni okresu stosowania danej strategii. Na rysunku 26. znajduje się wykres linii equity na poziomie dziennym dla wszystkich modeli i strategii kontrolnych poza Ref 3.

Rys. 26. Linie equity dla strategii inwestycyjnych opartych o wyniki modeli i strategii kontrolnych.



Źródło: Opracowanie własne.

Rysunek 28. pokazuje większą efektywność strategii inwestycyjnych modeli względem strategii kup i trzymaj. Fragmenty w których linie wykresów są równoległe do osi X reprezentują okresy, w których waloru nie ma w portfelu (więc nie następuje ich codniowe rozliczenie) . Sygnały inwestycyjne wygenerowane przez modele pozwalały unikać lub częściowo unikać okresów dłuższego spadku wartości kursu w portfelu (jak na przykład pomiędzy 11.05.2019 a 06.06.2019).

## PODSUMOWANIE

Inwestorzy posilkują się wiedzą matematyczną i statystyczną w podejmowaniu decyzji inwestycyjnych. Wraz z rozwojem metod statystycznych oraz stopniowym zwiększaniem łatwo dostępnej mocy obliczeniowej, do ich dyspozycji trafiły kolejne narzędzia – uczenie maszynowe i sieci neuronowe. Dziedziny te wciąż dynamicznie się rozwijają i nowe, usprawnione architektury mogą przynosić coraz większą skuteczność w dostarczaniu sygnałów inwestycyjnych. Istnieje więc prawdopodobieństwo, że sieci neuronowe będą dalej usprawniać handel algorytmiczny i zmieniać oblicze inwestowania.

Wyniki przeprowadzonych eksperymentów zdają się potwierdzać hipotezę badawczą - sieci neuronowe zbudowane w oparciu o architekturę LSTM mogą być skutecznym narzędziem w rękach inwestora. Dowód na to stanowią zarówno wyniki metryk dla zbioru testowego (dokładność powyżej 62%, RMSE na poziomie 0,582), jak i wyniki przeprowadzonych symulacji inwestycyjnych, które pokazują, że inwestor kierujący się sygnałami generowanymi przez najlepszy z modeli mógłby osiągnąć 885% zwrotu z inwestycji. Rzeczywisty wynik mógłby być znacznie wyższy, jeśli inwestor zdecydowałby się na reinwestowanie kapitału zyskanego po każdej sprzedaży waloru.

W testowym okresie, czyli pomiędzy 14.01.2019 a 06.11.2019, kurs instrumentu bazowego wzrósł o 24,89% (8207,63 wobec 6572), czyli przypadał na okres wyraźnej hossy. Przy czym skuteczność przewidywania modelu pozwoliła na osiągnięcie znacznie lepszych zwrotów z inwestycji, niż posługując się strategią kup i trzymaj.

Niemalże cały zbiór treningowy pochodzi z okresu hossy (02.01.2003 – 21.04.2017), z okresem spadkowym tylko w czasie kryzysu finansowego w latach 2007-2009. Nie zbadano, jak model radziłby sobie z przewidywaniem przyszłych wartości w okresie bessy. Ze względu na to, że sieci neuronowe lepiej generalizują dla danych o podobnym charakterze do tych, na których są uczone, system transakcyjny najprawdopodobniej osiągałby niższą skuteczność w okresie spadkowym.

Modele wyjaśniają tylko pewną część zmienności zmiennej objaśnianej. Można wnioskować, że dane historyczne nie są jedynymi czynnikami kształtującymi wartość kursu waloru i wiele zależy od czynników behawioralnych oraz psychologicznych. Pracę nad modelami



generującymi sygnały inwestycyjne można kontynuować poprzez badanie, czy modele autoregresyjne (zawierające tylko dane dotyczące wartości kursu), osiągają niższą skuteczność niż rozbudowane modele analizujące wiele zmiennych (kursy walut, ceny surowców, ceny innych walorów, dane makroekonomiczne).

W związku z tym, że inwestorzy reagują również na prognozowane wartości rozmaitych czynników (na przykład danych makroekonomicznych), zmieniają kurs walorów jeszcze przed oficjalnym ich ogłoszeniem. Prognozy są z kolei ogłaszane nieregularnie i z tego powodu trudno je uwzględnić w modelu. Rozwiązaniem tej kwestii mogłoby być stworzenie modelu sieci neuronowej, której zadaniem byłaby analiza nastroju/wydzźwięku (*sentiment analysis*) dla nagłówków lub treści artykułów z prasy ekonomicznej i uwzględnianie wyników takiej analizy jako zmiennych objaśniających w modelu głównym.

Kolejnym możliwym krokiem byłoby zbudowanie modeli nakładających się na siebie i analizujących różne fragmenty danych. Jeden model mógłby analizować ogólny wpływ ogłaszanych danych makroekonomicznych na kurs waloru, kolejny przeprowadzałby analizę wydzźwięku dla artykułów prasowych, a jeszcze inny identyfikowałby formacje analizy technicznej (na przykład przy użyciu uczenia nienadzorowanego). Wyniki z poszczególnych sieci trafiałyby jako dane wejściowe do kolejnego modelu, który analizowałby ostateczny wpływ ich zmienności na zmienność danego waloru.

## BIBLIOGRAFIA

- Adya, M., Collopy, F., 1998, How effective are neural networks at forecasting and prediction? A review and evaluation, *Journal of Forecasting*, 17(5-6), s. 481–495
- Borowski, K., 2014, *Analiza fundamentalna, metody wyceny przedsiębiorstwa*, Wyd.: Difin SA, Warszawa
- Dębski, W., 2014, *Rynek finansowy i jego mechanizmy*, Wyd. Naukowe PWN, Warszawa
- Flannery, M. J., Protopapadakis, A. A. , 2002, Macroeconomic Factors Do Influence Aggregate Stock Return, *The Review of Financial Studies*, Volume 15, Issue 3, April, s. 751 - 782
- Gamboa, J., 2017, Deep Learning for Time-Series Analysis, *ArXiv abs/1701.01887*
- Geske, R., Roll, R., 1983, The Fiscal and Monetary Linkage Between Stock Returns and Inflation, *The Journal of Finance*, Vol. 38, No. 1 (Mar., 1983), s. 1-33
- Géron, A., 2017, *Hands-On Machine Learning with Scikit-Learn and Tensorflow*, O'Reilly Media, Sebastopol
- Goodfellow, I. J., Bengio, Y., Courville, A., 2016, *Deep Learning*, MIT Press, Cambridge
- Hansson, M., 2017, On stock return prediction with LSTM networks, *LUP Student Papers*, Lund
- Heaton, J. B., Polson, N.G., Witte, J.H., 2018, Deep Learning in Finance, *ArXiv abs/1602.06561*
- Jorion, P., 2006, *Value at Risk: The New Benchmark for Managing Financial Risk*, 3rd Edition, Wyd.: The McGraw-Hill Companies Inc., Nowy Jork
- Karpoff, J., 1987, The Relation between Price Changes and Trading Volume: A Survey, *Journal of Financial and Quantitative Analysis*, 22, issue 1, s. 109-126
- Kim, O., Verrecchia, R. E., 1991, Trading Volume and Price Reactions to Public Announcements, *Journal of Accounting Research* 29, no. 2, s. 302-21
- Lamont, O., 2001, Economic Tracking Portfolios, *Journal of Econometrics*, 105, s. 161-184

- Li, Y., Ma, W., 2010, Applications of Artificial Neural Networks in Financial Economics: A Survey, IEEE 2010 3rd International Symposium on Computational Intelligence and Design (ISCID) - Hangzhou, China (2010.10.29-2010.10.31), s. 211–214
- Lu, W., Li, J., Li, Y., Sun, A., Wang, J., 2020, A CNN-LSTM-Based Model to Forecast Stock Prices, *Complexity*, vol. 2020. Article ID 6622927
- Markowitz, H., 1952, Portfolio Selection, *The Journal of Finance*, Vol. 7, No. 1. (Mar., 1952), s. 77 – 91
- McQueen, G., Roley, V., 1993, Stock Prices, News, and Business Conditions, *Review of Financial Studies*, 6, s. 683-707
- Minami, S., 2018, Predicting Equity Price with Corporate Action Events Using LSTM-RNN, *Journal of Mathematical Finance*, 8, s. 58-63
- Mitchell, T., 1997, *Machine Learning*, McGraw-Hill, Nowy Jork
- Murphy, J. J., 1999, *Technical Analysis of the Financial Markets. A Comprehensive Guide to Trading Methods and Applications*, New York Institute of Finance, Nowy Jork
- Pring, M. J., 2014, *Technical Analysis Explained (5th Edition)*, McGraw Hill, Singapur
- Saad, E.W., Prokhorov, D.V., Wunsch, D.C., 1998, Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks, *IEEE Transactions on Neural Networks*, 9(6), s. 1456–1470
- Schwert, G. W., 1989, Why Does Stock Market Volatility Change Over Time?, *Journal of Finance*, 44, s. 1115-1145
- Siami-Namini, S., Siami-Namini A., 2018, Forecasting Economics and Financial Time Series: ARIMA vs. LSTM, *ArXiv abs/1803.06386*
- Steinbach, M., C., 2001, Markowitz Revisited: Mean-Variance Models in Financial Portfolio Analysis, *SIAM Rev.*, 43(1), s. 31–85
- Tadeusiewicz, R., 1993, *Sieci Neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa

Tadeusiewicz, R., 2007, *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#*, Polska Akademia Umiejętności, Kraków

Tarczyński, W., 2001, *Rynki Kapitałowe: Metody Ilościowe – Vol. I*, wyd. 2, Wyd.: Agencja Wydawnicza „PLACET”, Warszawa

Zhu, X., Wang, H., Xu, L., Li, H., 2008, Predicting stock index increments by neural networks: The role of trading volume under different horizons, *Systems with Applications*, 34(4), s. 3043–3054

Strony internetowe:

<https://www.lynxbroker.pl/edukacja/handel-na-futures-przewodnik/> (dostęp na 20.08.2020)

<https://www.cmegroup.com/markets.html> (dostęp na 20.08.2020)

[https://www.cmegroup.com/trading/equity-index/us-index/e-mini-nasdaq-100\\_contract\\_specifications.html](https://www.cmegroup.com/trading/equity-index/us-index/e-mini-nasdaq-100_contract_specifications.html) (dostęp na 20.08.2020)

<https://capital.com/nasdaq-100-vs-sp-500> (dostęp na 20.08.2020)

[https://www.cmegroup.com/media-room/press-releases/2019/2/04/cme\\_group\\_averaged178millioncontractsperdayinjanuary2019.html](https://www.cmegroup.com/media-room/press-releases/2019/2/04/cme_group_averaged178millioncontractsperdayinjanuary2019.html) (dostęp na 20.08.2020)

<https://tickertape.tdameritrade.com/trading/micro-e-mini-index-futures-earnings-season-17619> (dostęp na 20.08.2020)

[https://www.cmegroup.com/trading/equity-index/us-index/e-mini-nasdaq-100\\_contract\\_specifications.html](https://www.cmegroup.com/trading/equity-index/us-index/e-mini-nasdaq-100_contract_specifications.html) (dostęp na 20.08.2020)

<https://stooq.pl/q/?s=nq.f> (dostęp na 20.08.2020)

<https://www.lynxbroker.pl/media/dokumenty/LYNX-Oplaty.pdf> (dostęp: 20.08.2020)

<https://www.nasdaq.com/market-activity/quotes/nasdaq-ndx-index> (dostęp na 21.08.2020)

[https://indexes.nasdaq.com/docs/Methodology\\_NDX.pdf](https://indexes.nasdaq.com/docs/Methodology_NDX.pdf) (dostęp na 21.08.2020)

[https://indexes.nasdaqomx.com/docs/methodologynew\\_NDX.pdf](https://indexes.nasdaqomx.com/docs/methodologynew_NDX.pdf) (dostęp na 21.08.2020)

<https://www.youtube.com/watch?v=0VH1Lim8gL8&t=4282s> Deep Learning State of the Art (2020) | MIT Deep Learning Series (dostęp na 19.11.2020)

<https://www.ibm.com/pl-pl/analytics/machine-learning> (dostęp na 20.11.2020)

<https://azure.microsoft.com/pl-pl/overview/machine-learning-algorithms/#uses> (dostęp na 07.12.2020)

<http://www.wildml.com/2016/08/rnns-in-tensorflow-a-practical-guide-and-undocumented-features/> (dostęp na 04.01.2021)

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (dostęp na 06.01.2021)

<https://finance.yahoo.com/quote/%5EVIX?p=%5EVIX>, (dostęp na 21.01.2021)

<https://www.macrotrends.net/2015/fed-funds-rate-historical-chart>, (dostęp na 20.01.2021)

<https://ihsmarkit.com/products/us-monthly-gdp-index.html>, dostęp na (20.01.2021)

<https://fred.stlouisfed.org/series/M1>, (dostęp na 23.01.2021)

<https://fred.stlouisfed.org/series/HOUST>, (dostęp na 23.01.2021)

<https://fred.stlouisfed.org/series/BOPGSTB>, (dostęp na 23.01.2021)

<https://fred.stlouisfed.org/series/INDPRO>, (dostęp na 23.01.2021)

<https://fred.stlouisfed.org/series/PCE>, (dostęp na 23.01.2021)

<https://pl.investing.com/commodities/gold-historical-data>, (dostęp na 23.01.2021)

<https://fred.stlouisfed.org/series/PPIACO>, (dostęp na 23.01.2021)

<https://keras.io/>, (dostęp na 25.01.2021)

<https://www.tensorflow.org/about>, (dostęp na 25.01.2021)

<https://machinelearningmastery.com/stateful-stateless-lstm-time-series-forecasting-python/>, (dostęp na 09.02.2021)

<https://www.blog.bitpeak.pl/post/inteligentne-uczenie-sztucznej-inteligencji-czyli-optimalizacja-bayesowska>, (dostęp na 11.02.2021)

# ZESTAWIENIE SPISÓW

## Spis tabel

Tabela 1. Początkowo wybrane zmienne. ....	87
Tabela 2. Zmienne dodane do modelu. ....	89
Tabela 3. Fragment danych po wstępnej obróbce. ....	90
Tabela 4. Korelacja pomiędzy zmiennymi objaśniającymi a zmienną objaśnianą. ....	92
Tabela 5. Wyniki modelu 1. ....	97
Tabela 6. Wyniki z pierwszych 12 dni sesyjnych dla najlepszego podejścia modelu 1. ....	98
Tabela 7. Wyniki modelu 2. z okienkiem czasowym z 7 sesji. ....	99
Tabela 8. Wyniki z pierwszych 10 dni sesyjnych dla najlepszego podejścia modelu 2. ....	99
Tabela 9. Wyniki modelu 3. z okienkiem czasowym z 7 sesji z tasowaniem paczki danych. ....	100
Tabela 10. Wyniki z pierwszych 10 dni sesyjnych dla najlepszego podejścia modelu 3. ....	100
Tabela 11. Wyniki modelu 4. z okienkiem czasowym z 8 sesji (ceteris paribus). ....	101
Tabela 12. Wyniki z pierwszych 9 dni sesyjnych dla najlepszego podejścia modelu 4. ....	102
Tabela 13. Wyniki modelu klasyfikacji (ModelC 1.) zbudowanego z takimi hiperparametrami jak Model 10. ....	109
Tabela 14. Wyniki z pierwszych 9 dni sesyjnych dla najlepszego podejścia ModelC 1. ....	109
Tabela 15. Podsumowanie metryk dla zbioru testowego modeli (średnich i najlepszych podejść). ....	112
Tabela 16. Średnie dzienne stopy zwrotu ważone ryzykiem. ....	116
Tabela 17. Liczba trafnych i błędnych sygnałów wygenerowanych przez strategię. ....	117

## Spis rysunków

Rys. 1. Portfele efektywne (optymalne połączenia zmienności i wartości oczekiwanej zwrotów). ....	10
Rys. 2. Linie trendu dla ceny akcji PKN ORLEN SA w okresie 01.06.2020 – 20.12.2020. ....	21
Rys. 3. Linie oporu i wsparcia dla ceny akcji banku PEKAO SA w okresie 01.03.2020 – 01.07.2020. ....	22
Rys. 4. Wykres kursu indeksu WIG20 w okresie 01.02.2018 – 01.08.2019 z zaznaczoną formacją głowy z ramionami. ....	24
Rys. 5. Formacja podwójnego szczytu dla waloru ORANGEPL w okresie 01.07.2020 – 05.08.2020. ....	25
Rys. 6. Przykładowy schemat działania głębokiej sieci neuronowej rozpoznającej obraz. ....	45
Rys. 7. Symulacja działania metody gradientu prostego dla funkcji o jednym parametrze theta (lub wektorze parametrów theta). ....	51
Rys. 8. Możliwy problem z odnalezieniem minimum funkcji przy zastosowaniu zbyt dużego współczynnika uczenia. ....	52
Rys. 9. Potencjalne trudności w odnalezieniu globalnego minimum. ....	53
Rys. 10. Istota procesu uczenia polega na posuwaniu się w kierunku spadku powierzchni błęd. ....	54
Rys. 11. Rodzaje dopasowania modelu do danych. ....	55
Rys. 12. Błąd w relacji do elastyczności modelu. ....	56

Rys. 13. Porównanie schematów algorytmów rozwiązujących problemy na bazie danych.....	58
Rys. 14. Schemat budowy prostej, dwuwarstwowej sieci neuronowej.....	59
Rys. 15. Struktura neuronu nieliniowego jako rozszerzenie neuronu liniowego o nieliniową funkcję przejścia.....	61
Rys. 16. Wykres funkcji LeakyReLU dla dziedziny $(-100, 100)$ i parametru $\alpha=0,1$ .....	62
Rys. 17. Zmniejszanie się błędu w trakcie procesu uczenia dla różnych wartości początkowych wag w badanym modelu.....	66
Rys. 18. Schemat działania jednoneuronowej sieci rekurencyjnej rozłożonej w czasie.....	68
Rys. 19. Schemat działania sieci rekurencyjnej z 5 krokami czasowymi dla danych wejściowych i 3 krokami w sekwencji wyjściowej.....	70
Rys. 20. Schemat neuronu LSTM.....	72
Rys. 21. Porównanie zmienności trzech największych indeksów giełdowych Stanów Zjednoczonych.....	84
Rys. 22. Pojedyncza sekwencja danych wejściowych w postaci okienka czasowego dla $SEQ\_LEN = 8$ wraz z jej etykietą.....	94
Rys. 23. Wykres metryk zbioru testowego dla średnich z trzech podejść uczenia.....	102
Rys. 24. Porównanie dwóch najlepszych modeli – wartości przewidziane przez model dla zbioru testowego (niebieska linia) i wartości rzeczywiste (pomarańczowa linia).....	107
Rys. 25. Zysk osiągnięty przez modele o najlepszych wynikach metryk.....	115
Rys. 26. Linie equity dla strategii inwestycyjnych opartych o wyniki modeli i strategii kontrolnych.....	118

## Spis wzorów

Wzór 1.1.....	9
Wzór 1.2.....	9
Wzór 1.3.....	9
Wzór 1.4.....	17
Wzór 1.5.....	27
Wzór 1.6.....	27
Wzór 1.7.....	28
Wzór 1.8.....	28
Wzór 1.9.....	29
Wzór 1.10.....	30
Wzór 1.11.....	31
Wzór 2.1.....	48
Wzór 2.2.....	49
Wzór 2.3.....	49

Wzór 2.4.....	50
Wzór 2.5.....	53
Wzór 2.6.....	53
Wzór 2.7.....	53
Wzór 2.8.....	60
Wzór 2.9.....	60
Wzór 2.10.....	61
Wzór 2.11.....	61
Wzór 2.12.....	62
Wzór 2.13.....	62
Wzór 2.14.....	62
Wzór 2.15.....	64
Wzór 2.16.....	64
Wzór 2.17.....	68
Wzór 2.18.....	68
Wzór 2.19.....	73
Wzór 2.20.....	73
Wzór 2.21.....	73
Wzór 2.22.....	74
Wzór 2.23.....	74
Wzór 3.1.....	114



## ZAŁĄCZNIKI

### Załącznik A. Glosariusz nazw zmiennych i pojęć wraz z opisem hiperparametrów i metryk modeli.

Tabela 1. Glosariusz nazw zmiennych i pojęć.

Skrót/pojęcie	Definicja
future	Zmienna objaśniana w modelach regresji – zmiana procentowa wartości kursu z dnia bieżącego na dzień kolejny
trainX	Dane wejściowe modelu – zmienne objaśniające zbioru treningowego
trainY	Dane wyjściowe modelu – zmienna objaśniana zbioru treningowego
devX	Dane wejściowe modelu – zmienne objaśniające zbioru walidacyjnego
devY	Dane wyjściowe modelu – zmienna objaśniana zbioru walidacyjnego
testX	Dane wejściowe modelu – zmienne objaśniające zbioru testowego
testY	Dane wyjściowe modelu – zmienna objaśniana zbioru testowego
MinMaxScaler	Funkcja biblioteki SciKitlearn standaryzująca wartości tak, by mieściły się w zadanym przedziale (zachowując swój rozkład)
Sequential()	Funkcja biblioteki Keras pozwalająca na projektowanie sieci – w ramach tej funkcji należy wybrać typ neuronów, ich ilość, funkcję aktywacji itd.
Batch	Paczka danych treningowych – liczba sekwencji w jednej paczce treningowej po których model aktualizuje swoje wagi
Early stopping	Wczesniejsze przerywanie uczenia w optymalnym momencie w celu uniknięcia nadmiernego dopasowania do danych treningowych
ReLU	Funkcja aktywacji opisana w rozdziale 2.
Adam	Algorytm optymalizacyjny bazujący na iteracyjnej metodzie gradientowej – opisany w rozdziale 2. Powstał z połączenia AdaGrad i RMSProp
RMSProp	Algorytm optymalizacyjny bazujący na iteracyjnej metodzie gradientowej i wykorzystujący średnią kroczącą gradientów
model.fit	Funkcja biblioteki Keras, która uruchamia trening modelu
Predicted	Wartość zmiennej future przewidziana przez model
Actual	Prawdziwa wartość zmiennej future dla danego dnia
IndexValue	Średnia z Otwarcia i Zamknięcia dla Nasdaq 100 E-mini futures dla danego dnia sesyjnego
Signal	Sygnal inwestycyjny generowany przez sieć – Buy, Sell lub Hold
Grid Search	Jedna z metod optymalizacji hiperparametrów modelu - polega na wyszukiwaniu wyczerpującym w ręcznie zdefiniowanym obszarze hiperparametrów algorytmu uczącego
Random Search	Jedna z metod optymalizacji hiperparametrów modelu – wyszukuje optymalnych hiperparametrów spośród losowych wartości. Często stosowana, jeśli model ma niewiele hiperparametrów istotnie wpływających na jego wyniki
Optymalizacja bayesowska	Jedna z metod optymalizacji hiperparametrów modelu - algorytm, który podejmuje próby oszacowania optymalizowanej funkcji na podstawie poznanych wcześniej wartości. Dziedzina jest zbiór hiperparametrów, a zbiorem wartości funkcja straty głównego modelu
Hyperopt	Biblioteka języka Python do przeprowadzania bayesowskiej optymalizacji hiperparametrów na modelach ułożonych w Keras
Hyperas	Biblioteka języka Python - wrapper umożliwiający łatwiejsze wykorzystanie Hyperopta na modelach ułożonych w Keras

FSignal	Zmienna objaśniana w modelu klasyfikacji, która jednocześnie jest sygnałem inwestycyjnym – jeśli następnego dnia sieć przewiduje istotny wzrost kursu, przyjmuje wartość 0, jeśli istotny spadek – wartość 1, jeśli zmiana będzie nieistotna – wartość 2.
Softmax	Funkcja aktywacji w warstwie wyjściowej modelu klasyfikacyjnego – określa, jaka jest szansa na to, że nastąpi jedno z możliwych zdarzeń (nadając prawdopodobieństwo wystąpienia każdej z klas i wybierając tę o najwyższym prawdopodobieństwie).

Źródło: Opracowanie własne.

Tabela 2. Opis hiperparametrów modeli.

Hiperparametry modeli	Opis
SEQ_LEN	Długość okienka czasowego – liczba dni sesyjnych wchodzących w skład pojedynczej sekwencji danych wejściowych (graficznie przedstawione na Rys. 24.)
BATCH SIZE	Wielkość paczki danych treningowych – liczba sekwencji w jednej paczce treningowej po których model aktualizuje swoje wagi
LEARNING RATE	Współczynnik uczenia – hiperparametr sterujący maksymalną dopuszczalną zmianą wag w modelu po jednej paczce danych wejściowych
SHUFFLE	Tasuj – hiperparametr binarny – jeśli ustawiony jako prawdziwy, tasuje paczki danych treningowych (batch) i nadaje im losową kolejność (przez co nie są ustawione chronologicznie)
EPOCHS	Epoki – Liczba iteracji – tyle razy model zyskuje dostęp do wszystkich danych wejściowych zbioru treningowego (chyba że early stopping zatrzyma uczenie wcześniej)
LAYER	Warstwa sieci neuronowej o numerze x – w tych hiperparametrach określa się ilość neuronów w danej warstwie oraz ich rodzaj (LSTM lub w pełni połączone)
ACTIV	Funkcja aktywacji wykorzystana w modelu – ReLU, LeakyReLU lub liniowa
OPT	Funkcja optymalizacyjna modelu – Adam lub RMSProp. Wartość po nazwie funkcji oznacza zastosowany współczynnik uczenia
DROPOUT	Czy część losowych neuronów ma być pomijana w kolejnych iteracjach uczenia, żeby bardziej uodpornić sieć na błędne/brakujące dane. Procent neuronów, które mają być pominięte ustala się osobno dla każdej warstwy

Źródło: Opracowanie własne.

Tabela 3. Opis metryk modeli.

Metryki modeli	Opis
MSE	Błąd średniokwadratowy – wartość oczekiwana kwadratu błędu (różnicy między estymatorem a wartością estymowaną)
RMSE	Pierwiastkowy błąd średniokwadratowy – pierwiastek z MSE
MAE	Średni błąd absolutny – średnia z błędów w próbie (z różnicy pomiędzy wartością przewidywaną a wartością rzeczywistą)
Sparse Categorical Cross Entropy	Różnica pomiędzy dwoma rozkładami prawdopodobieństwa dla danego zestawu zdarzeń
Accuracy	Dokładność – procent poprawnie sklasyfikowanych wektorów w stosunku do całkowitej ich liczby (procent dobrze nadanych klas)

Źródło: Opracowanie własne.

## Załącznik B. Wartości hiperparametrów dla wszystkich trenowanych modeli.

### ZADANIE REGRESJI:

Tabela 4. Wartości hiperparametrów dla Model 1.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
5	64	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	1 FC	ReLU	Adam; 0,001	False	No

Źródło: Opracowanie własne.

Tabela 5. Wartości hiperparametrów dla Model 2.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
7	32	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	1 FC	ReLU	Adam; 0,001	False	Yes

Źródło: Opracowanie własne.

Tabela 6. Wartości hiperparametrów dla Model 3.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
7	32	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	1 FC	ReLU	Adam; 0,001	True	Yes

Źródło: Opracowanie własne.

Tabela 7. Wartości hiperparametrów dla Model 4.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
8	32	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	1 FC	ReLU	Adam; 0,001	True	Yes

Źródło: Opracowanie własne.

Tabela 8. Wartości hiperparametrów dla Model 5.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
9	32	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	1 FC	ReLU	Adam; 0,001	True	Yes

Źródło: Opracowanie własne.

Tabela 9. Wartości hiperparametrów dla Model 6.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
10	32	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	1 FC	ReLU	Adam; 0,001	True	Yes

Źródło: Opracowanie własne

Tabela 10. Wartości hiperparametrów dla Model 7.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
11	32	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	1 FC	ReLU	Adam; 0,001	True	Yes

Źródło: Opracowanie własne

Tabela 11. Wartości hiperparametrów dla Model 8.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
14	32	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	1 FC	ReLU	Adam; 0,001	True	Yes

Źródło: Opracowanie własne

Tabela 12. Wartości hiperparametrów dla Model 9.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
8	32	100	448	336	8
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8	1	Leaky ReLU; Liniowa	RMSprop; 0,001	Yes	Yes

Źródło: Opracowanie własne

Tabela 13. Wartości hiperparametrów dla Model 10.

SEQ_LEN	BATCH_SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
8	48	100	480	302	32
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
32	1	ReLU; Liniowa	RMSprop; 0,0029	Yes	Yes

Źródło: Opracowanie własne

## ZADANIE KLASYFIKACJI:

Tabela 14. Wartości hiperparametrów dla ModelC 1.

SEQ LEN	BATCH SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
8	48	100	480	302	32
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
32	3	ReLU; Liniowa	RMSprop; 0,0029	Yes	Yes

Źródło: Opracowanie własne

Tabela 15. Wartości hiperparametrów dla ModelC 2.

SEQ LEN	BATCH SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
8	32	100	224 LSTM	448 LSTM	224 FC
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
8 FC	3	ReLU	Adam; 0,001	True	Yes

Źródło: Opracowanie własne

Tabela 16. Wartości hiperparametrów dla ModelC 3.

SEQ LEN	BATCH SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
8	32	100	336	84	56
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
16	3	ReLU	RMSprop; 0,001	Yes	Yes

Źródło: Opracowanie własne

Tabela 17. Wartości hiperparametrów dla ModelC 4.

SEQ LEN	BATCH SIZE	EPOCHS	LAYER1	LAYER2	LAYER3
8	46	100	336	96	70
LAYER4	LAYER5	ACTIV	OPT	SHUFFLE	DROPOUT
24	3	Leaky ReLu, Liniowa	RMSprop; 0,0058	Yes	Yes

Źródło: Opracowanie własne

### Załącznik C. Zestawienie wyników dla trzech podejść uczenia każdego z modeli

#### ZADANIE REGRESJI:

Tabela 18. Metryki dla Model 1.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,358	0,435	0,598
	Walidacja	1,257	0,771	1,121
	Test	0,578	0,684	0,760
	Epoka	6		
<b>II PRÓBA</b>	Trening	0,516	0,512	0,718
	Walidacja	0,711	0,599	0,843
	Test	0,566	0,562	0,752
	Epoki	5		
<b>III PRÓBA</b>	Trening	1,015	0,728	1,007
	Walidacja	0,708	0,597	0,842
	Test	<b>0,555</b>	<b>0,559</b>	<b>0,745</b>
	Epoki	1		

Źródło: Opracowanie własne

Tabela 19. Metryki dla Model 2.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,527	0,520	0,726
	Walidacja	0,583	0,550	0,764
	Test	0,512	0,550	0,715
	Epoka	22		
<b>II PRÓBA</b>	Trening	0,501	0,503	0,708
	Walidacja	0,564	0,512	0,751
	Test	0,506	0,541	0,712
	Epoki	29		
<b>III PRÓBA</b>	Trening	0,502	0,507	0,709
	Walidacja	0,495	0,482	0,703
	Test	<b>0,378</b>	<b>0,454</b>	<b>0,615</b>
	Epoki	25		

Źródło: Opracowanie własne

Tabela 20. Metryki dla Model 3.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,550	0,524	0,741
	Walidacja	0,473	0,468	0,688
	Test	0,393	0,469	0,627
	Epoka	20		
<b>II PRÓBA</b>	Trening	0,512	0,508	0,716
	Walidacja	0,454	0,459	0,674
	Test	<b>0,359</b>	<b>0,444</b>	<b>0,599</b>
	Epoki	34		
<b>III PRÓBA</b>	Trening	0,580	0,531	0,762
	Walidacja	0,465	0,475	0,682
	Test	0,386	0,471	0,622
	Epoki	16		

Źródło: Opracowanie własne

Tabela 21. Metryki dla Model 4.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,595	0,551	0,771
	Walidacja	0,477	0,468	0,690
	Test	<b>0,338</b>	<b>0,424</b>	<b>0,582</b>
	Epoka	22		
<b>II PRÓBA</b>	Trening	0,554	0,525	0,744
	Walidacja	0,489	0,475	0,699
	Test	0,399	0,465	0,632
	Epoki	23		
<b>III PRÓBA</b>	Trening	0,546	0,522	0,739
	Walidacja	0,450	0,458	0,671
	Test	0,378	0,448	0,615
	Epoki	31		

Źródło: Opracowanie własne

Tabela 22. Metryki dla Model 5.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,566	0,527	0,752
	Walidacja	1,000	0,489	0,713
	Test	0,457	0,497	0,676
	Epoka	17		
<b>II PRÓBA</b>	Trening	0,540	0,519	0,735
	Walidacja	0,481	0,480	0,694
	Test	0,396	0,476	0,629
	Epoki	34		
<b>III PRÓBA</b>	Trening	0,513	0,509	0,716
	Walidacja	0,461	0,468	0,679
	Test	<b>0,394</b>	<b>0,460</b>	<b>0,628</b>
	Epoki	32		

Źródło: Opracowanie własne

Tabela 23. Metryki dla Model 6.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,518	0,513	0,720
	Walidacja	0,451	0,459	0,672
	Test	0,385	0,464	0,621
	Epoka	40		
<b>II PRÓBA</b>	Trening	0,529	0,517	0,727
	Walidacja	0,459	0,469	0,678
	Test	<b>0,360</b>	<b>0,449</b>	<b>0,600</b>
	Epoki	27		
<b>III PRÓBA</b>	Trening	0,513	0,510	0,716
	Walidacja	0,494	0,480	0,703
	Test	0,389	0,456	0,624
	Epoki	38		

Źródło: Opracowanie własne



Tabela 24. Metryki dla Model 7.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,604	0,542	0,777
	Walidacja	0,472	0,477	0,687
	Test	0,387	0,465	0,622
	Epoka	19		
<b>II PRÓBA</b>	Trening	0,528	0,515	0,726
	Walidacja	0,458	0,459	0,676
	Test	<b>0,365</b>	<b>0,449</b>	<b>0,605</b>
	Epoki	26		
<b>III PRÓBA</b>	Trening	0,537	0,513	0,733
	Walidacja	0,500	0,476	0,707
	Test	0,368	0,449	0,607
	Epoki	39		

Źródło: Opracowanie własne

Tabela 25. Metryki dla Model 8.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,937	0,687	0,968
	Walidacja	0,716	0,604	0,846
	Test	0,562	0,562	0,750
	Epoka	6		
<b>II PRÓBA</b>	Trening	0,636	0,562	0,798
	Walidacja	0,492	0,488	0,701
	Test	<b>0,346</b>	<b>0,444</b>	<b>0,588</b>
	Epoki	27		
<b>III PRÓBA</b>	Trening	0,525	0,520	0,724
	Walidacja	0,477	0,473	0,691
	Test	0,358	0,447	0,599
	Epoki	29		

Źródło: Opracowanie własne

Tabela 26. Metryki dla Model 9.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,579	0,528	0,761
	Walidacja	0,455	0,467	0,675
	Test	0,355	0,446	0,596
	Epoka	12		
<b>II PRÓBA</b>	Trening	0,512	0,508	0,715
	Walidacja	0,436	0,460	0,660
	Test	0,382	0,456	0,618
	Epoki	29		
<b>III PRÓBA</b>	Trening	0,560	0,525	0,748
	Walidacja	0,438	0,461	0,661
	Test	<b>0,349</b>	<b>0,441</b>	<b>0,591</b>
	Epoki	29		

Źródło: Opracowanie własne

Tabela 27. Metryki dla Model 10.

	<b>Metryka/Zbiór</b>	<b>MSE</b>	<b>MAE</b>	<b>RMSE</b>
<b>I PRÓBA</b>	Trening	0,500	0,504	0,707
	Walidacja	0,423	0,449	0,651
	Test	<b>0,340</b>	<b>0,429</b>	<b>0,583</b>
	Epoka	25		
<b>II PRÓBA</b>	Trening	0,553	0,511	0,744
	Walidacja	0,450	0,462	0,671
	Test	0,343	0,439	0,586
	Epoki	24		
<b>III PRÓBA</b>	Trening	0,537	0,514	0,733
	Walidacja	0,441	0,467	0,664
	Test	0,361	0,442	0,601
	Epoki	19		

Źródło: Opracowanie własne

## ZADANIE KLASYFIKACJI

Tabela 28. Metryki dla ModelC 1.

	<b>Metryka/Zbiór</b>	<b>Sparse Categorical Cross Entropy</b>	<b>Accuracy</b>
<b>I PRÓBA</b>	Trening	0,894	0,593
	Walidacja	0,852	0,601
	Test	0,882	<b>0,594</b>
	Epoka	13	
<b>II PRÓBA</b>	Trening	0,922	0,575
	Walidacja	0,894	0,612
	Test	0,900	0,594
	Epoki	11	
<b>III PRÓBA</b>	Trening	0,911	0,572
	Walidacja	0,852	0,629
	Test	<b>0,877</b>	0,585
	Epoki	9	

Źródło: Opracowanie własne

Tabela 29. Metryki dla ModelC 2.

	<b>Metryka/Zbiór</b>	<b>Sparse Categorical Cross Entropy</b>	<b>Accuracy</b>
<b>I PRÓBA</b>	Trening	0,876	0,603
	Walidacja	0,883	0,605
	Test	0,916	0,560
	Epoka	33	
<b>II PRÓBA</b>	Trening	0,873	0,612
	Walidacja	0,884	0,612
	Test	<b>0,903</b>	0,580
	Epoki	29	
<b>III PRÓBA</b>	Trening	0,881	0,606
	Walidacja	0,866	0,610
	Test	0,907	<b>0,585</b>
	Epoki	26	

Źródło: Opracowanie własne

Tabela 30. Metryki dla ModelC 3.

	<b>Metryka/Zbiór</b>	<b>Sparse Categorical Cross Entropy</b>	<b>Accuracy</b>
<b>I PRÓBA</b>	Trening	0,875	0,610
	Walidacja	0,862	0,636
	Test	<b>0,859</b>	<b>0,623</b>
	Epoka	32	
<b>II PRÓBA</b>	Trening	0,913	0,587
	Walidacja	0,855	0,619
	Test	0,881	0,599
	Epoki	18	
<b>III PRÓBA</b>	Trening	0,910	0,590
	Walidacja	0,843	0,629
	Test	0,885	0,599
	Epoki	22	

Źródło: Opracowanie własne

Tabela 31. Metryki dla ModelC 4.

	<b>Metryka/Zbiór</b>	<b>Sparse Categorical Cross Entropy</b>	<b>Accuracy</b>
<b>I PRÓBA</b>	Trening	0,914	0,571
	Walidacja	0,856	0,641
	Test	<b>0,884</b>	0,604
	Epoka	8	
<b>II PRÓBA</b>	Trening	0,892	0,589
	Walidacja	0,887	0,617
	Test	0,896	<b>0,614</b>
	Epoki	17	
<b>III PRÓBA</b>	Trening	0,867	0,605
	Walidacja	0,874	0,608
	Test	0,889	0,594
	Epoki	22	

Źródło: Opracowanie własne