



FEDERACJA NAUKOWA
WSB-DSW Merito

mgr inż. Jarosław Malczewski

**Modele i algorytmy głębokiego uczenia
w analizie szeregów czasowych cen mleka w proszku**

Rozprawa doktorska

**Promotor:
dr hab. Zbigniew Krysiak, prof. SGH**

Warszawa, grudzień 2023

Spis treści

Wstęp	3
1 Podstawowe narzędzia analizy szeregów czasowych i ich znaczenie w ekonomii	12
1.1 Model szeregu czasowego	14
1.2 Modele statystyczne ciągów czasowych	16
1.2.1 Wstęp	17
1.2.2 Procesy stochastyczne - podstawowe definicje i własności	17
1.2.3 Modele autoregresji	20
1.3 Wybrane metody identyfikacji procesów	30
1.3.1 Metoda najmniejszych kwadratów jako metoda estymacji parametrów modeli parametrycznych	31
1.3.2 Empiryczne podejścia do wyboru modeli o charakterze praktycznym	36
1.3.3 Funkcja autokorelacji reszt modeli	40
1.3.4 Komponenty związane z trendami	42
1.3.5 Podejście Boxa-Jenkinsa do identyfikacji modeli na przykładzie modelu ARIMA	43
1.3.6 Proces wyboru modeli autoregresyjnych szeregów czasowych (TAR)	45
1.3.7 Facebook Prophet	48
1.3.8 Identyfikacja modeli na przykładzie modelu ARIMA (podejście Boxa-Jenkinsa)	49
1.3.9 Model ARX i identyfikacja dla ciągu obserwacji procesu	56
1.3.10 Prognoza przy stosowaniu modelu TAR	59
1.4 Sieci neuronowe jako narzędzie służące do prognozowania	68
1.4.1 Podstawy budowy sieci neuronowych	72
1.4.2 Funkcje aktywacji	76
1.4.3 Metody uczenia sieci neuronowych	79
1.4.3.1 Funkcja kosztu	80
1.4.3.2 Uogólniona reguła delta	80
1.4.3.3 Wsteczna propagacja błędów	81
1.4.3.4 Warianty metody gradientu prostego	81
1.4.3.5 Inicjalizacja wag	83
1.4.3.6 Regularyzacja	83
1.4.3.7 Etapy uczenia głębokiej sieci neuronowej	86
1.4.4 Sieci neuronowe a prognozowanie szeregów czasowych	87

1.4.5	Analiza predykcji z wykorzystaniem sztucznych sieci neuronowych	89
1.4.6	Głębokie sieci neuronowe	93
1.4.7	Architektury głębokich sieci neuronowych	94
1.4.7.1	Autokodery	94
1.4.7.2	Ograniczone Maszyny Boltzmannna	95
1.4.7.3	Splotowe sieci neuronowe	99
1.4.7.4	Rekurencyjne sieci neuronowe	101
1.4.8	Głębokie uczenie	102
1.5	Filtracja szeregów czasowych cen	105
1.5.1	Wstęp do filtracji cyfrowej	106
1.5.2	Próbkowanie oszczędne	112
1.5.3	Warunki zastosowania próbkowania oszczędnego	115
2	Proponowany algorytm prognozowania cen odtłuszczonego mleka w prosz-	
	ku	125
2.1	Wstępne odsumianie: kondycjonowanie danych uczących	128
2.2	Przygotowanie danych wejściowych	131
2.3	Komponent LSTM	132
2.4	Mechanizm uwagi	133
2.5	Komponent CNN	133
2.6	Weryfikacja i testowanie skonstruowanego algorytmu predykcyjnego	141
2.7	Wnioski	144
	Streszczenie pracy	161
	Bibliografia	166
	Spis ilustracji	193
	Spis tabel	195

Wstęp

Modelowanie szeregów czasowych jest powszechnym podejściem w finansach do przedstawiania różnych zjawisk. W niniejszym opracowaniu zgłębiono problematykę prognozowania szeregów czasowych w dziedzinie finansów, ze szczególnym uwzględnieniem zagadnienia prognozowania cen akcji. Badanie wykorzystuje konwencjonalne metodologie statystyczne, a mianowicie MA, ARIMA i ARMA-GARCH, w połączeniu ze współczesnymi podejściami opartymi na głębokim uczeniu się, takimi jak RNN, LSTM, GRU, TCN oraz dwukierunkowe LSTM i GRU, do prognozowania cen akcji. Przeprowadzono analizę porównawczą tych technik. Przeprowadzono dokładne badanie i przedstawiono szczegółową analizę wyników eksperymentów w różnych zestawach danych. Wyniki naszego badania wskazują, że ARIMA i GRU wykazują lepsze wyniki w kontekście jednoetapowego przewidywania cen akcji w porównaniu z alternatywnymi architekturami głębokiego uczenia. Nie zaobserwowano, aby włączenie wskaźników rynkowych i ekonomicznych zwiększało skuteczność modeli głębokiego uczenia się. Jeśli chodzi o prognozowanie wieloetapowe, zaobserwowano, że ARIMA wykazuje lepszą wydajność w porównaniu z wieloetapowymi GRU/TCN i Seq2Seq GRU/TCN. Ponadto wykazano, że wykorzystanie uczenia transferowego zwiększa skuteczność modeli głębokiego uczenia się. Akcje stanowią kluczową kategorię aktywów dla każdej osoby pragnącej inwestować. Istnieje wiele korzyści związanych z inwestowaniem w akcje. Z historycznego punktu widzenia akcje wykazały zdolność oferowania zarówno płynności, jak i korzystnych zwrotów w dłuższych okresach czasu. Według dostępnych danych, Nasdaq Composite przyniósł zwrot w wysokości 171,64 proc. w okresie 5 lat. Ponadto korporacje notowane na giełdzie są prawnie zobowiązane do ujawniania swojej kondycji finansowej, dostarczając w ten sposób mnóstwo danych związanych z inwestycjami. Regulacja rynków akcji służy ograniczeniu różnego rodzaju ryzyka transakcji finansowych. Biorąc pod uwagę zalety związane z akcjami jako istotną klasą aktywów, konieczne jest wszechstronne zrozumienie optymalnego momentu kupna i sprzedaży akcji w celu maksymalizacji rentowności. Aby wykonać to zadanie, konieczne jest precyzyjne przewidywanie kursu akcji. Dwie podstawowe kategorie prognozowania cen akcji to analiza fundamentalna i analiza techniczna [203]. W dziedzinie finansów powszechnie przyjmuje się, że rzeczywista wartość akcji jest równa zdyskontowanej sumie jej przewidywanych przyszłych przepływów pieniężnych. Podstawowa zasada leżąca u podstaw popytu i podaży akcji polega na tym, że opierają się one na przewidywanych przyszłych korzyściach płynących z akcji. Kiedy aktualna cena rynkowa akcji spada poniżej jej wartości wewnętrznej, inwestorzy postrzegają to jako korzystny zakup. W przypadku, gdy dominująca cena rynkowa przekracza wartość wewnętrzną, akcjonariusz byłby skłonny zbyć swoje akcje. Procedura analityczna stosowana do określenia wartości wewnętrznej jest

powszechnie określana jako analiza fundamentalna [203]. Dodatkowa forma analizy jest powszechnie określana jako analiza techniczna. Zdaniem analityków technicznych, obecna cena akcji obejmuje wszystkie dostępne informacje. Analiza techniczna to metodologia, która ma na celu wyeliminowanie ludzkiej subiektywności i emocji z analizy rynków finansowych. Podejście to opiera się na szeregu narzędzi, w tym wykresach, wykresach i technikach matematycznych, takich jak regresja liniowa, ARIMA i GARCH. Na cenę akcji mają wpływ różne czynniki, takie jak między innymi ceny historyczne, trendy rynkowe, światowa gospodarka i trendy branżowe. Wymienione czynniki mogą potencjalnie służyć jako predyktory ceny akcji. W przypadku polegania wyłącznie na koszcie pojedynczego zdarzenia w celu przewidywania nadchodzących cen, dane zadanie jest klasyfikowane jako jednowymiarowe zagadnienie prognozowania szeregów czasowych. Problem wielowymiarowych szeregów czasowych pojawia się, gdy cena jest wskazywana przez wiele czynników. Niniejsze badanie przeprowadza badanie różnych technik prognozowania danych szeregów czasowych i ocenia ich skuteczność. Niedawny postęp w głębokim uczeniu się spowodował znaczne zainteresowanie wykorzystaniem technik głębokiego uczenia się do celów prognozowania szeregów czasowych. Dlatego prowadzone jest badanie kontrastu między konwencjonalnymi technikami statystycznymi a współczesnymi metodami głębokiego uczenia. Przeprowadzono analizę porównawczą modeli głębokiego uczenia się i tradycyjnych modeli szeregów czasowych, a mianowicie prostej średniej ruchomej, wykładniczej średniej ruchomej, ARIMA i ARMA GARCH, w celu oceny ich skuteczności w przewidywaniu cen akcji. Badanie obejmuje modele głębokiego uczenia, takie jak rekurencyjne sieci neuronowe (RNN) i ich pochodne, a mianowicie pamięć długoterminowa (LSTM), bramkowana jednostka rekurencyjna (GRU), dwukierunkowa GRU (Bi-GRU) i dwukierunkowa LSTM (Bi-LSTM). Z uwagi na niemożność modelowania wartości szeregów czasowych cen przez liniowe modele predykcyjne oraz aktywny, wciąż zmieniający się w swej naturze i dynamice, wpływ czynników zewnętrznych, konieczne jest zastosowanie innych mechanizmów opisujących dynamikę zmian oraz pozwalających na wiarygodne wyznaczenie przyszłych wartości na podstawie przeszłych wartości z uwzględnieniem, dotychczas pomijanych, zewnętrznych czynników wpływu [271]. Szeregi czasowe zmiennych finansowych i ekonomicznych stanowią zwykle wynik interakcji kilku składowych, które różnią się typem oraz charakterem [213]. W klasycznym podejściu do modelowania i prognozowania ekonometrycznego wartości szeregu czasowego doprowadza się dekompozycji na konkretne składowe, biorąc pod uwagę zwykle wahania sezonowe oraz tendencję rozwojową. Zastosowanie w tym procesie metod matematyczno-statystycznych umożliwia projektowanie modeli biorących pod uwagę istnienie konkretnych osobliwości. Pozwala to na budowanie prognoz opartych na deterministycznych fundamentach [319]. W szacowaniu przyszłych wartości szeregów czasowych można już skutecznie zastosować sztuczne

sieci neuronowe, które osiągają umiejętność prognozowania w wyniku procesu uczenia. Sztuczne sieci neuronowe umożliwiają budowę modeli odzwierciedlających złożone relacje zachodzące między danymi wejściowymi i wyjściowymi dla zjawisk, których charakter, zasady działania oraz czynniki przyczynowo-skutkowe nie zostały rozpoznane w stopniu pozwalającym na ich modelowanie przy pomocy znanych metod numerycznych [241]. Sieci są zdolne do ekstrapolowania ze zbiorów danych związków niewykrywalnych nawet przez tradycyjne metody statystyczne [147], [13]. Ponadto, sieci neuronowe wykazują zdolność do aproksymowania dowolnej funkcji, w tym o silnie nieliniowych cechach, a takie właśnie mają miejsce w przypadku szeregów czasowych cen mleka w proszku. Należy dostrzec, że ze względu na dużą złożoność szeregów cen produktów giełdowych takich, jak mleko w proszku i masło, tradycyjne metody deterministyczne oraz stochastyczne nie prowadzą do otrzymywania wiarygodnych wyników prognozowania [122], [111].

Czynnikiem pomijanym w prognozowaniu przy użyciu metod konwencjonalnych, takich jak np. ARIMA, jest brak uwzględnienia szeregu czynników psychologiczno-społecznych, które stanowią istotny czynnik determinujący zmiany cen. Aspekty te mogą zostać uwzględnione przy projektowaniu systemu „inteligentnego”, jednak skuteczność takiego modelu zależy również od „relewantności” danych uczących, co musi zostać uwzględnione przy projektowaniu takiego modelu. Z tych powodów wiele badań analityków koncentruje się na prognozowaniu cen odnotowanych przez giełdy rynku światowego, co uznano za jedno z najważniejszych, ale trudnych zadań w dziedzinie badań prognostycznych. Bazując na wynikach istniejących badań, można zaobserwować, że zastosowano różne bardzo wydajne modele sztucznej inteligencji (ang. Artificial Intelligence, AI) w celu przezwyciężenia trudności w modelowaniu nieliniowości zawartej w przebiegach czasowych oraz danych wejściowych. Ponadto, wymagane w tradycyjnych modelach, założenia co do właściwości statystycznych (np. stacjonarność, regularność i liniowość) nie są wymagane w modelach AI. W związku z tym wiele technik AI, np. sztuczne sieci neuronowe (ang. Artificial Neural Networks, ANN), maszyny wektorów nośnych (ang. Support Vector Machine, SVM) i różne algorytmy optymalizacji (np. algorytm genetyczny (ang. Genetic Algorithm, GA)), były powszechnie stosowane w prognozowaniu ekonomicznym. Ostatnie dwie dekady rodzą wiele całkowicie nowych rozwiązań technologicznych w tej materii, np. głębokie sieci neuronowe i uczenie, które to, jako merytoryczne podłoże, stanowiły inspirację do zaprojektowania własnego algorytmu predykcyjnego, który jest jednym z przedmiotów niniejszej rozprawy doktorskiej [214].

Uzasadnieniem stosowania mechanizmów sztucznej inteligencji i zainteresowania rynkiem mleka są trudności w oszacowaniu przyszłych wartości przy pomocy metod konwencjonalnych, takich jak AR, MA, ARMA, ARIMA, z uwagi na immanentny szum informacyjny, który prowadzi do błędów predykcji [146], [188]. Źródłami szumu są czynniki

prowadzące do pojawienia się w szeregu czasowym wartości, które przyczyniają się do zaistnienia krótkookresowych skoków wartości, co w dalszej kolejności utrudnia interpretację trendu. Do tego rodzaju czynników zaliczyć należy m.in. tendencje spekulacyjne (granie na wyżkę albo spadek ceny), nieprawidłowe wartości w szeregu czasowym cen, incydentalne decyzje kadry zarządzającej sektorem mleczarskim pod wpływem czynników deterministycznych lub psychologicznych, nagłe zmiany pogody, klęski żywiołowe, stany zagrożenia epidemicznego, kryzysy gospodarcze, panika inwestorów giełdowych, uwolnienie wolnego kapitału inwestorskiego i transferowanie go do sektora mleczarskiego, fałszywe informacje medialne (ang. fake news), itd. Aspekty związane z polityką monetarną oraz kursami wymiany walut należy również wziąć pod uwagę, ponieważ wpływają one na przebieg międzynarodowych operacji finansowych.

Z powodu występowania tych czynników wstępna filtracja danych mająca na celu wyeliminowanie szumu z szeregów czasowych cen, tj. odsumianie, jest czynnością konieczną z uwagi na wymagania stawiane przez algorytmy predykcyjne na wektory uczące. Precyzyjne oszacowanie przyszłych wartości miałyby istotny wpływ zarówno na działania operacyjne, jak i inwestorskie, bo prowadziłyby do ograniczenia ryzyka finansowego [192]. Utylitarność podjętej problematyki rozprawy doktorskiej polega na dostarczeniu narzędzi analitycznych dla głównych podmiotów rynkowych, tj. producentów mleka, przetworzonej żywności, rolników, zakładów przetwórstwa mleka i produktów sprzężonych oraz decydentów polityki rolnej. W literaturze przedmiotu brak jest przedstawionych algorytmów predykcji trendów dla rynków rolnych, także dla rynku mleka. W tym wyraża się innowacyjność problematyki podejmowanej w rozprawie. Ponadto, zastosowany, w autorskim algorytmie, model splotowych sieci neuronowych posługujących się metodologią tzw. głębokiego uczenia do tej pory nie był stosowany w obszarze predykcji cen, a stanowił podstawę działania narzędzi klasyfikujących w obszarze cyfrowego przetwarzania sygnałów mowy oraz obrazów [172], [282].

Cel i zakres pracy

Wpływ cen OMP na koszty produkcji w przemyśle spożywczym oraz trudności w ich modelowaniu przez większość algorytmów predykcyjnych skłania do zainteresowania problematyką prognostycznych algorytmów obliczeniowych, które wykazują zdolność do analizy szeregów czasowych cen OMP. Potrzeba ta ujawnia również lukę w dostępnych narzędziach do dokładnego prognozowania cen OMP. Możliwość opracowania precyzyjniejszych narzędzi predykcyjnych umożliwiłaby złagodzenie niepożądanych wpływów zmian cen substratów mleczarskich i w dalszej kolejności na utrzymanie stabilnych cen produktów mlecznych. Stanowi to podstawowy obszar pracy i podjętych w niej rozważań. Zakres pracy obejmuje zarówno wprowadzenie teoretyczne dotyczące charakterystyki produkcji, rynku i przetwórstwa mleka, jak też uwarunkowań ekonomicznych stanowiących podłoże dużej zmienności dynamicznej oraz nieregularności szeregów czasowych cen mleka w proszku, jak i najistotniejszą część tej pracy, tj. matematycznych podstaw i uzasadnienia do zastosowania przedstawionego w pracy autorskiego algorytmu predykcyjnego.

Należy zauważyć, że w ramach badań zrealizowanych w niniejszej pracy ujawniono istotną lukę badawczą oznaczającą brak dostępnych narzędzi do prognozowania cen OMP. Rynki rolne nie cieszą się szczególnym zainteresowaniem środowiska ekonometrycznego, które nie dysponuje ostatnimi zdobyczami nauki w tym zakresie. Prognozowanie szeregów czasowych, co do ogółu, wzbudza jednak duże zainteresowanie społeczności naukowej. Obserwuje się również liczne próby adaptacji, dotychczas stosowanych w innych obszarach nauki (np. w cyfrowym przetwarzaniu sygnałów), technik matematyki stosowanej oraz metod numerycznych w obszarze prognozowania ekonometrycznego. Do tej pory nie znalazły one powszechnego zastosowania w obszarze prognozowania cen OMP.

W swoich pracach Murphy i in. [213] oraz Rasekhschaffe i Jones [232] pokazują wszechstronność wielu dostępnych algorytmów prognozowania oraz opisują wymogi stawiane w metodologii oceny skuteczności działania poszczególnych algorytmów, co dalej służyć ma zrealizowaniu rzetelnych testów porównawczych poszczególnych rozwiązań konkurencyjnych. Zasadniczym celem niniejszej pracy jest zaprojektowanie oraz implementacja autorskiego algorytmu predykcji cen odtłuszczonego mleka w proszku. Skonstruowany algorytm numeryczny ma ponadto zostać poddany rygorystycznym testom ilościowym w celu wykazania jego przewagi nad konkurencyjnymi rozwiązaniami.

W odniesieniu do pytań o charakterze konstrukcyjnym, eksplanacyjnym (wyjaśniającym) oraz deskryptywnych (opisowych) sformułowano następujące hipotezy badawcze:

- (a) Algorytmy sztucznej inteligencji są właściwym narzędziem służącym do konstrukcji i implementacji przewyższających rozwiązania konkurencyjne systemów prognozowania cen odtłuszczonego mleka w proszku.

- (b) Metody rekonstrukcji sygnałów niepełnych są właściwym narzędziem służącym do wyeliminowania immanentnego szumu obciążającego próbki szeregów czasowych cen mleka w proszku.
- (c) Szeregi czasowe cen mleka w proszku są realizacją niestacjonarnego procesu.
- (d) Szeregi czasowe cen mleka w proszku nie można sprowadzić do postaci stacjonarnej na drodze różnicowania.

Formułując problem badawczy postawiono następujące cele szczegółowe:

- (a) identyfikacja czynników wpływu na zmienność cen odtłuszczonego mleka w proszku,
- (b) porównanie algorytmów predykcji szeregów czasowych cen,
- (c) identyfikacja czynników ograniczających dokładność algorytmów predykcji,
- (d) optymalizacja algorytmów sztucznej inteligencji poprzez stosowanie odszumiania danych trenujących,
- (e) optymalizacja algorytmów sztucznej inteligencji poprzez zastosowanie metod głębokiego uczenia,
- (f) opracowanie hybrydowych algorytmów analizy cen odtłuszczonego mleka w proszku.

Z uwagi na postęp techniczny w zakresie metod predykcyjnych oraz rosnące oczekiwania uczestników rynku rolnego w zakresie stosowania efektywnych algorytmów prognostycznych, koniecznością stało się sięgnięcie po splotowe (konwolucyjne) sieci neuronowe, dotychczas stosowane jedynie w obszarze klasyfikacji obrazów i dźwięków [207], [190], [201], [293], [308].

W części analitycznej pracy postawiono również następujące pytania szczegółowe:

- (a) Czy istnieje możliwość predykcji cen produktów mleczarskich przy użyciu optymalnych algorytmów sztucznej inteligencji?
- (b) Jak wyeliminować ograniczenia konwencjonalnych systemów predykcyjnych?
- (c) Jak wyeliminować ograniczenia systemów predykcyjnych opartych na algorytmach sztucznej inteligencji?
- (d) Jak zidentyfikować (kwantyfikować) cechy szczególne rynków rolnych oraz jak je uwzględnić w algorytmach numerycznych?

Nowatorskim elementem, który zastosowano w niniejszej pracy po raz pierwszy stanowi użycie sieci splotowych jako narzędzia predykcyjnego oraz nowe podejście do odszumiania zbiorów danych, tj. stworzenie nowego paradygmatu głębokiego uczenia SSN poprzez wykorzystanie opartych na metodologii próbkowania oszczędnego (ang. compressed sensing, CS) procedur odszumiania [191], [276]. Metoda ta zakłada że dane uczące są rzadkie i niekompletne. Protokół rekonstrukcji sygnałów rzadkich, będący częścią metodologii próbkowania oszczędnego, służy w niniejszej pracy jako efektywne narzędzie odszumiania. [151], [168], [292].

Wadą istniejących rozwiązań zaprezentowanych w literaturze dotyczącej prognozowania cen akcji jest ich niezdolność dokładnego przewidzenia wysoce dynamicznych i szybko zmieniających się wzorców ruchu cen akcji. Obecne rozwiązania mają na celu zaradzić temu niedociągnięciu, wykorzystując potencjał konwolucyjnych sieci neuronowych (ang. Convolutional Neural Networks, CNN) w uczeniu się zachowań zmian cen akcji w przeszłości i na tej podstawie sporządzaniu bardzo dokładnych prognoz przyszłych zachowań giełdowych [162].

W prezentowanym algorytmie dane uczące są odszumiane przy użyciu algorytmu bazującego na metodologii próbkowania oszczędnego. Wstępne filtrowanie danych uczących wykorzystuje zatem rzadkość danych. Odszumianie oparte na strukturze CS zamienia tę słabość (tj. pozornie niedostateczną reprezentację liczbową) w mocną stronę. Algorytmy CS w przeszłości wykorzystane były w algorytmach gromadzenia danych. Koncepcja ta zakłada, że dane w bazie pewnego przekształcenia ortogonalnego są nadmiarowe (w sensie kryterium wymagalności ilości danych potrzebnych do poprawnej rekonstrukcji sygnałów według Nyquista-Shannona). Filtrowanie wektorów uczących ma zatem na celu zmniejszenie poziomu szumu i dzięki temu, w dalszej kolejności, zwiększenie wydajności procesu przewidywania. Z tego powodu istotną część niniejszej pracy poświęcono zasadom rekonstrukcji sygnałów, w tym sygnałów o znacząco zredukowanej reprezentacji informacyjnej.

Prezentowana praca ma charakter teoretyczno-konstrukcyjny, a jej struktura obejmuje trzy rozdziały oraz nienumerowane części jak: wstęp, podsumowanie, wnioski, bibliografię oraz spis ilustracji i tabel. W pierwszej części rozdziału pierwszego przedstawiono problematykę rynku w gospodarce. Druga część tego rozdziału jest próbą opisu i kwantyfikowania charakterystycznych i specyficznych cech rynku mleczarskiego, które stwarzają obiektywne trudności dla konwencjonalnych algorytmów predykcyjnych, wskazując tym samym na konieczność zastosowania metod o charakterze samoorganizującym. Część druga (rozdział 2) pracy traktuje o podstawowych narzędziach analizy szeregów czasowych, by na jej koniec starać się przybliżyć problematykę zarówno sieci neuronowych jak i filtracji cyfrowej, które to znalazły zastosowanie w trzecim rozdziale. Trzecia część niniejszej

pracy ma charakter konstrukcyjno-doświadczalny i jest próbą opisu i scharakteryzowania, proponowanego przez autora niniejszej pracy, własnego algorytmu predykcyjnego. W rozdziale trzecim przedstawiono też badania doświadczalne, prezentując jednocześnie zarówno użyte w pracy miary oceny wydajności algorytmów predykcji, jak i porównano przy ich użyciu proponowany algorytm z czołowymi rozwiązaniami światowymi w zakresie predykcji szeregów czasowych. Pracę wieńczy podsumowanie i wnioski przedstawiające wyniki końcowe i wskazuje możliwą perspektywę dalszych badań.

Metoda i materiał

W części teoretycznej pracy zaprezentowano studium literatury przedmiotu badań, w tym przeanalizowano krajową i zagraniczną literaturę z zakresu analizy rynków, sytuacji produkcyjnej i cenowej na światowych rynkach mleka, współczesnych modeli ekonometrycznych, zwłaszcza algorytmów sztucznej inteligencji, głębokiego uczenia oraz odszumiania szeregów czasowych. Część praktyczno-analityczna natomiast obejmuje informacje pozyskane z międzynarodowych i krajowych baz danych, w tym m.in.: zaprojektowanie autorskich modeli ekonometrycznych, przygotowanie i implementację, przygotowanie i analizę danych uczących dla algorytmów sztucznej inteligencji algorytmów eliminacji szumu z szeregów czasowych, analizę skuteczności działań rozwiązań autorskich oraz porównanie rozwiązań autorskich z algorytmami *state-of-the-art*. W niniejszej pracy wykorzystano wektory danych tygodniowych cen OMP od 2000 do 2020 roku, dostarczone przez Global Dairy Trade. Głównym celem pracy jest zaprojektowanie, wykazanie jego skuteczności przy użyciu miar numerycznych, algorytmu predykcji cen OMP. W tym celu zaimplementowano i porównano m.in. sieć sprzężenia zwrotnego oraz sieć rekurencyjną. Sieć z wyprzedzeniem była uczona przez standardowy algorytm wstecznej propagacji, podczas gdy rekurencyjna była wyzwalana metodą wielostrumieniową opartą na rozszerzonym filtrze Kalmana [214].

Dotychczas stosowane modele prognostyczne posłużyły jako inspiracja do pokonania przeszkody w wyrażaniu wysoce nieliniowej, ukrytej charakterystyki cen akcji OMP. W pracy zaprojektowano, zaimplementowano i przetestowano autorski algorytm predykcyjny, który to w dalszej kolejności poddano restrykcyjnym testom wydajnościowym. W części eksperymentalnej pracy testom poddano nie tylko konkretne algorytmy predykcyjne, ale również submetody wchodzące w skład ich struktur. Osobnej analizie poddano techniki odszumiania wektorów uczących. Czynność ta miała na celu wskazanie zarówno najlepszego algorytmu (w sensie mierzalnych kryteriów wydajności), jak i która metoda odszumiania pozwala na osiągnięcie najlepszych wyników. Z tego powodu uwagę poświęcono sieciom pamięci długoterminowej (*Long Short Term Memory - LSTM*) [183], sztucznym sieciom neuronowym (*Artificial Neural Network-ANN*), modelom opartym na mechanizmie uwagi oraz głębokim sieciom splotowym (*Deep Convolutional Network-DCN*) [232]. W niniejszej pracy zasadniczy mechanizm predykcyjny jest realizowane poprzez wykorzystanie jednej ze struktur modelowania głębokiego uczenia, tzw. „Multiple Input, Multiple Output” (MIMO). Wskaźniki wydajności obliczono przy użyciu ustalonych kryteriów oceny (średniej różnicy, MSE, RMSE, RMSLE oraz D_{stat}) dla każdej procedury w celu oceny wydajności zarówno rozwiązania jak i rozwiązań konkurencyjnych.

Rozdział 1.

Podstawowe narzędzia analizy szeregów czasowych i ich znaczenie w ekonomii

Szereg czasowy to ciąg obserwacji danego zjawiska pokazujący się jego kształtowanie w kolejnych okresach (dniach, miesiącach, kwartałach, latach, itp.). Szeregi czasowe stanowią podstawę oceny zjawisk dynamicznych. Szeregi czasowe charakteryzujące zmienne ekonomiczne stanowią zazwyczaj rezultat złożenia wielu, różniących się charakterem, składowych. W klasycznym podejściu do modelowania ekonometrycznego wartości szeregu czasowego doprowadza się do jego dekompozycji na poszczególne składowe obejmujące zazwyczaj tendencję rozwojową i wahania sezonowe. Zastosowanie w tym obszarze metod matematyczno-statystycznych pozwala na budowę formalnych modeli opisujących zaobserwowane prawidłowości. Szacowanie przyszłych wartości na podstawie szeregów czasowych jest istotną trudnością w naukach inżynierskich oraz ekonomicznych. Tradycyjne metody przewidywania wykorzystują bezpośrednie obliczanie przyszłych wartości w oparciu o liniową kombinację ich przeszłych wartości. Istnieją obszerne badania [212], [80], [174], [239], [140], [141], [101] w tej dziedzinie, które ujawniają liczne zastosowania w takich dziedzinach, jak między innymi ekonomia, ekonometria, finanse i prognozowanie giełdowe. Stock i in. [101] oraz Weng [334] poczynili duże starania w zakresie rozważań nad zastosowaniem metod opartych na numerycznych modelach liniowych. Modele te wykorzystują koncepcje autoregresji (AR) i średnich ruchomych/kroczących (MA). Procedura AR wymaga założenia, że wartość bieżąca szeregu czasowego jest liniową kombinacją jego poprzednich wartości. Istotą modeli średnich ruchomych (MA) jest założenie, że bieżąca wartość może być reprezentowana przez kombinację przypadkowych przejść lub szumów, które wpłynęły na szeregi czasowe. Tego typu podejścia dowiodły swojej przydatności w dziedzinie oceny dynamiki wielu szeregów czasu rzeczywistego.

Niemniej jednak, z powodu potrzeby założenia liniowości, modele te okazały się niewystarczające w naśladowaniu nieliniowego charakteru rzeczywistej dynamiki, a to może wystąpić w praktycznych zastosowaniach [195], [205]. Z tych powodów liczni autorzy dostrzegli konieczność zastosowania adoptowalnych modeli numerycznych opartych na metodologiach sztucznej inteligencji. Zastosowanie sztucznych sieci neuronowych pozwala na dokładniejsze prognozowanie, ponieważ można uwzględnić istnienie nieliniowości w szeregach cen wielu surowców [304], [138], [186]. Alternatywne podejścia natomiast skupiły się na wykorzystaniu jednoetapowych procedur prognozowania [139]. W ostatnich latach szczególną uwagę poświęcono metodom prognozowania opartym na sztucznych sieciach neuronowych (ANN, ang. artificial neural networks), częściowo ze względu na ich

elastyczność i możliwości w zakresie modelowania przyszłych wartości [29], [253]. Najnowsze metody wykorzystują rozszerzone topologie ANN, włączając model AR oparty na nieliniowości, a także element architektury dynamicznej [211]. Uwagę poświęcono również modelom wykorzystującym sieci neuronowe, [183], [225] w których wymusza się włączenie zarówno modelu AR, jak i uwzględnienie funkcji nieliniowych. Ponadto Lyons i in. zaproponowali zastosowanie technik hybrydowych wielowarstwowo - perceptronowych modeli liniowych [205].

Niezależnie od osiągnięć w obszarze SSN i ich faktycznego wpływu na naukę, jest oczywiste, że wiele kwestii pozostaje wciąż dla nich otwartych [312]. Należy podkreślić fakt istnienia kilku czynników obecnych w praktycznych metodach modelowania [90]. Czynniki te należy uważać za abstrakcyjne i związane z przeszłym zachowaniem modelowanego procesu [315]. Z tego powodu aspekty te są problematyczne w zakresie uwzględnienia ich w architekturach algorytmów podejmowania decyzji [183]. Kolejną poważną przeszkodą, z jaką borykają się SSN, jest wykluczenie przez nie modeli MA [50]. Większość proponowanych metod wykorzystuje jedynie nieliniowy model AR. Niestety w tych pracach model MA nie jest w ogóle brany pod uwagę. Ta wada prowadzi do błędnych przewidywań [212]. Z tych powodów, by osiągnąć oczekiwane wyniki prognozowania, należy uwzględnić zarówno AR, jak i komponenty MA. Ich włączenie umożliwia modelom predykcyjnym opartym na sztucznej inteligencji modelowanie obu czynników przy użyciu ich elastycznych oraz komplementarnych właściwości [113]. Konwencjonalne modele stochastyczne i ekonometryczne stosowane w prognozach finansowych szeregów czasowych nie są w stanie skutecznie poradzić sobie z niepewnym zachowaniem szeregów walutowych. Najnowsze badania zrealizowane w tym zakresie pokazują skuteczność SSN w zakresie klasyfikacji i predykcji. Udowodniono, że SSN może przybliżać dowolną funkcję ciągłą [312]. Wykorzystanie SSN w prognozowaniu szeregów danych finansowych było zatem oczywistą koniecznością [111]. Tego rodzaju algorytmy przewidywania szeregów czasowych pokazują swoją wyższość nad metodami konwencjonalnymi, takimi jak Box-Jenkins, AR, ARMA czy ARIMA [13]. Metody oparte na sztucznej inteligencji nie muszą być wyzwalane przez jakiegokolwiek informacje a priori dotyczące serii danych. Należy zauważyć, że jedną z najpoważniejszych przeszkód dla algorytmów prognozowania jest ich interaktywna złożoność, dynamiczny charakter i stale zmieniające się czynniki. Te aspekty powodują wysoki poziom szumu, który wpływa na dane wejściowe i w konsekwencji prowadzi do słabej wydajności algorytmów przewidywania. Z tych powodów filtrowanie szumów może prowadzić do poprawy ich wydajności. Wielu autorów zaproponowało kilka metod odszumiania. Sang i in. w [236] zasugerowali nowe algorytmy odszumiania bazujące na transformacji falkowej, powiązane z entropią, do klasyfikacji szeregów czasowych. Z kolei He w [179] przedstawił schemat regresji wektorów wspierających metodą Slantle-

ta najmniejszych kwadratów (LSSVR) do prognozowania kursu wymiany. Faria w [169] przedstawił wykładniczy, wygładzający model SSN, z uwzględnieniem odszumiania, do prognozowania akcji giełdowych. Yuan [277] zasugerował obiecujący schemat, łącząc koncepcję przełączania Markowa i filtrowanie Hodricka-Prescotta do prognozowania kursów wymiany. Nasser w [214] przedstawił metodę hybrydową wykorzystującą rozszerzony filtr Kalmana i algorytm genetyczny celem oceny przyszłego zapotrzebowania na wodę. Z kolei Chen i in. w [287] zaadoptowali właściwości transformacji Fouriera w schemacie przewidywania rozmytych szeregów czasowych cen akcji. Sang [237] zasugerował zmodyfikowaną strukturę odszumiania opartą na transformacji falkowej.

1.1 Model szeregu czasowego

Studia literaturowe [122] - [299] z zakresu predykcji szeregów czasowych wskazują, że rosnącą podgrupą technik predykcyjnych są modele bazujące na koncepcji sztucznych sieci neuronowych. Przedstawione dotyczą modeli długo, średnio oraz krótkookresowych oraz operacyjnych [21]. Charakteryzują się one różną długością horyzontu czasowego, tj. im jest on dłuższy, tym w większym stopniu dotyczy zjawisk ilościowych i jakościowych. Im jest on jednak, tym w większym stopniu tracimy kontrolę nad ilustracją istotnych cech zjawiska, tj. trendu i zmian jakościowych. Dla najkrótszych horyzontów czasowych, rzędu 1 godziny, istotne jest jedynie zwiększanie lub zmniejszanie wartości zmiennej przewidywanej wzdłuż linii trendu, tj. obrazowania zmian ilościowych. W przypadku prognozy operacyjnej, w której znana jest krzywa trendu, zaleca się zastosowanie metod ekstrapolacji [147]. Prognozy operacyjne są jednak najmniej powszechne w publikacjach krajowych i zagranicznych. Z doświadczenia i dotychczasowych badań należy wnioskować, że model sztucznych sieci neuronowych bardzo dobrze nadaje się do tych celów i zostało to wielokrotnie potwierdzone, np. [202], [146], [192] [188], [243], [306], [244]. Dowiedziono, że modele SSN dają lepsze wyniki prognozowania niż tradycyjne modele ekonometryczne oparte na metodach statystycznych. Należy jednak zaznaczyć, że wszystkie przedstawione, w cytowanej powyżej literaturze przykłady, oparte są na odpowiednio przygotowanych szeregach czasowych. W tej części przeprowadzona zostanie odpowiednia analiza szeregów czasowych [29]. Zakłada się, że obrazowany proces jest zjawiskiem dynamicznym i ciągłym. Tworzy on zbiór mierzonych wartości i umieszcza je chronologicznie w tych samych i kolejnych momentach (interwałach)[187]. Ten zbiór nazywany jest szeregiem czasowym. Oznaczając przez $t(t = 0, 1, \dots, n - 1)$ przedziały czasu, w których obserwowano wartości pewnego procesu, a przez y_t wyniki obserwacji, szereg czasowy zapisujemy, jako zbiór $\{y_t; t = 0, 1, \dots, n - 1\}$. Wartości zmiennej charakteryzującej owy proces (zmiennej zależnej - endogenicznej) uwarunkowane są czasem t (zmienną niezależną - egzogeniczną). Ponadto, szereg o charakterze dynamicznym może służyć jako podstawa wnioskowania

o przebiegu i rozwoju opisywanego zjawiska lub procesu w badanym horyzoncie czasowym, ale tylko wtedy, gdy zawiera zarejestrowaną dostateczną liczbę zmiennych przebiegu procesu w czasie przeszłym. Do formalnego opisu szeregów czasowych należy użyć następujących składowych:

1. Składowa systematyczna - jest skutkiem oddziaływań stałego zestawu czynników:
 - A. tendencja rozwojowa - tzw. trend, odzwierciedla długookresową skłonność do jednokierunkowych zmian wartości analizowanego procesu (wzrost lub spadek ceny), oznaczana jako - trend $T(t)$;
 - B. stały (przeciętny) poziom zmiennej prognozowanej, występuje wówczas, gdy w szeregu czasowym wartości prognozowanej zmiennej oscylują wokół pewnego stałego poziomu i brak jest trendu, oznaczana jako - stały średni poziom;
 - C. składowa okresowa (periodyczna):
 - wahania cykliczne, czyli długookresowe rytmiczne wahania, niekoniecznie regularne, wartości zmiennej wokół stałego (przeciętnego) poziomu lub trendu, oznaczana jako - cykl $C(t)$;
 - wahania sezonowe, czyli powtarzające się w określonym przedziale czasowym w ciągu roku wahania wartości obserwowanego procesu wokół tendencji rozwojowej lub stałego poziomu tej zmiennej. W przypadku cen mleka w proszku może to odzwierciedlać wpływ czynników determinujących cenę, tj. zachowań inwestorów, sezonowego zapotrzebowania na surowiec, itp. oznaczana jako - sezon $S(t)$;
2. Składowa przypadkowa - tzw. szum, składnik losowy lub wahania przypadkowe, oznaczana jako - nieregularność $\varepsilon(t)$.

W skomplikowanych procesach stochastycznych, jakim bez wątpienia jest rozkład ceny mleka w proszku w czasie, występowanie wahań sezonowych i cyklicznych identyfikuje się poprzez analizę funkcji korelacji zmiennej oraz tej samej wartości zmiennej, lecz z opóźnieniem o $1, 2, 3, \dots, k$ jednostek czasu. Dla pewnego przedziału czasowego można sformułować tzw. klasyczny model szeregu czasowego, który zawiera wszystkie składowe szeregu czasowego. W zależności od przyjętych założeń odnośnie powiązania między elementami składowymi szeregu przyjmuje on postać: **addytywną** - zgodnie, z którą wpływ poszczególnych komponentów składowych szeregu czasowego na kształtowanie wartości zmiennej cen Y w czasie t sumuje się. Przyjęcie addytywnego modelu szeregu czasowego wymusza konieczność założenia, że wszystkie jego elementy składowe są niezależne (nie zachodzą relacje między poszczególnymi składowymi) oraz, że wyrażone są w tych samych jednostkach, co zmienna Y . Jednocześnie założyć należy, że addytywny model

szeregu czasowego charakteryzuje się stałymi wahaniami okresowymi, które są wyrażane jako odchylenia od tendencji rozwojowej. Model przyjmuje postać:

$$Y_t = T_t + C_t + S_t + \varepsilon_t \quad (1.1)$$

gdzie: T_t - funkcja czasu, opisująca tendencję rozwojową - trend czasowy, w przypadku stałego (średniego) poziomu prognozowanej zmiennej $T_t = const$,

C_t - funkcja czasu, charakteryzująca wahania sezonowe,

S_t - funkcja czasu, charakteryzująca wahania cykliczne,

$\varepsilon(t)$ - zmienna losowa (składnik losowy) **multiplikatywną** - zgodnie, z którą wahania cykliczne, sezonowe i przypadkowe wyrażone są, jako względne odchylenia wartości zmiennej obserwowanego procesu Y od jej wartości wynikającej z trendu. Przyjęcie multiplikatywnego modelu szeregu czasowego przesądza o tym, w jakich jednostkach wyrażone muszą być jego poszczególne elementy składowe. I tak wartość wynikająca z trendu wyrażona jest najczęściej w takich samych jednostkach jak zmienna Y [16] natomiast wszystkie odchylenia mają charakter względny, tzn. wyrażone są w postaci bezwymiarowych wskaźników (często w %). Multiplikatywny model szeregu czasowego charakteryzuje się proporcjonalnymi (do skali zjawiska) wahaniami okresowymi i ma postać:

$$Y_t = T_t \cdot C_t \cdot S_t \cdot \varepsilon_t \quad (1.2)$$

gdzie: T_t - funkcja czasu, charakteryzująca tendencję rozwojową - trend czasowy, w przypadku stałego (średniego) poziomu prognozowanej zmiennej $T_t = const$,

C_t - funkcja czasu, charakteryzująca wahania sezonowe,

S_t - funkcja czasu, charakteryzująca wahania cykliczne,

$\varepsilon(t)$ - zmienna losowa (składnik losowy)

1.2 Modele statystyczne ciągów czasowych

Modele statystyczne to matematyczne reprezentacje rzeczywistych zjawisk, które służą do analizy i interpretacji danych. Modele te mają na celu przechwytywanie podstawowych wzorców i relacji w danych oraz przewidywanie lub wnioskowanie na podstawie tych wzorców. Są one szeroko stosowane w różnych dziedzinach, w tym między innymi w ekonomii, naukach społecznych, inżynierii i biologii. Modele statystyczne mogą być proste lub złożone, w zależności od złożoności danych i stawianego pytania badawczego. Zazwyczaj są one opracowywane przy użyciu oprogramowania statystycznego i oceniane na podstawie ich zdolności do dopasowania danych, mocy predykcyjnej i zdolności do uogólniania na nowe dane.

Analiza szeregów czasowych to technika statystyczna używana do analizy danych gromadzonych w czasie. Obejmuje badanie wzorców i trendów w danych w celu przewidywania przyszłych wartości. Analiza szeregów czasowych jest powszechnie stosowana

w takich dziedzinach, jak ekonomia, finanse i inżynieria, do prognozowania przyszłych trendów i podejmowania świadomych decyzji na podstawie danych historycznych.

1.2.1 Wstęp

Zastosowanie modeli statystycznych, które zostały opracowane w ramach dyscypliny statystyki matematycznej, odegrało zasadniczą rolę w analizie sygnałów dyskretnych. Modele te znalazły szerokie zastosowanie w różnych dziedzinach nauki. Głównym celem opracowania tych modeli było zaspokojenie wymagań ekonometrii i teorii sterowania. Teoria sekwencji czasowych obejmuje określone uporządkowanie, które ma istotne znaczenie w teorii sterowania. To uporządkowanie zostało po raz pierwszy wprowadzone przez Boxa i Jenkinsa w ich pracy nad liniowymi modelami szeregów czasowych i składa się z czterech odrębnych komponentów, a mianowicie AR, MA, ARMA i ARIMA (Box i Jenkins, 1970).

Badania prowadzone w dziedzinie ekonometrii doprowadziły do powstania różnych modeli statystycznych, np. dotyczących analizy nieliniowych szeregów czasowych. Przykłady takich modeli obejmują modele autoregresji progowej (TAR) i autoregresyjnego gładkiego przejścia (STAR). Stwierdzono, że modele te są szczególnie przydatne do rejestrowania złożonych relacji w danych ekonomicznych.

Modele wykorzystane w analizie obejmują model autoregresji progowej, model autoregresji warunkowej heteroskedastyczności i kilka innych modeli udokumentowanych w literaturze (Brockwell i Davis, 1996), (Chatfield, 1975), (Chen i Tsay, 1993), (Welfe, 2009).

Badania statystyczne opierają się na badaniu sekwencji obserwacji pochodzących z rzeczywistych procesów. Ilustracją takich danych są notowania czołowych indeksów rynkowych, takich jak PMP, NASDAQ, Dow-Jones, S&P, notowania kursów walut, oceny zmienności procesów technologicznych, obserwacje parametrów pogodowych czy obserwacje astronomiczne.

1.2.2 Procesy stochastyczne - podstawowe definicje i własności

Artykuł ten ma na celu przedstawienie przeglądu podstawowych pojęć i charakterystyk procesów stochastycznych.

Zbiór dowolnych zmiennych ułożonych chronologicznie nazywa się procesem stochastycznym, natomiast manifestacja tych zmiennych w kolejnych przedziałach czasowych lub instancjach określana jest jako szereg czasowy. Według Fiszera (1967) próbkę N kolejnych obserwacji, oznaczoną jako $x' = (x_1, x_2, \dots, x_N)$, można uznać za ciąg czasowy losowany z nieskończonej puli możliwych próbek, które być wytwarzane w tym samym

procesie. Proces stochastyczny X_t jest zbiorem zmiennych losowych, które są ułożone w sekwencji czasowej na podstawie zmiennej t reprezentującej czas.

$$\{X(t, \omega); X \in \chi, t \in T, \omega \in \Omega\}, \quad (1.3)$$

Równanie (1.3) reprezentuje zbiór wartości X , gdzie X należy do zbioru χ , t należy do zbioru T , a ω należy do zbioru Ω .

Powyższy zapis oznacza, że zbiór wartości procesowych jest reprezentowany przez χ , natomiast zbiór wartości parametrów t , typowo związanych z czasem, jest reprezentowany przez T . Przestrzeń zdarzeń elementarnych jest oznaczana przez Ω , jak ustalono wcześniej w pracach Dooba (1953), Fisz (1967) i Papoulis (1972).

$X(t)$ jest zmienną stochastyczną o rozkładzie prawdopodobieństwa, gdzie t jest stałą.

Funkcja $F_t(x)$ reprezentuje prawdopodobieństwo, że zmienna losowa $X(t)$ jest mniejsza od x .

Proces stochastyczny $X(t)$ charakteryzuje się wspólnym n -wymiarowym rozkładem prawdopodobieństwa, który jest określony przez dystrybuantę dla dowolnego zbioru wartości parametrów $\{t_1, t_2, \dots, t_n\}$.

w zapis F_{t_1, t_2, \dots, t_n} . Wyrażenie $(x_1, x_2, \dots, x_n) = P\{X(t_1) < x_1, \dots, X(t_n) < x_n\}$ oznacza prawdopodobieństwo, że zmienne losowe X w chwilach t_1 do t_n są mniejsze niż odpowiednie wartości od x_1 do x_n .

Pełną charakterystykę procesu stochastycznego uzyskuje się poprzez zebranie wszystkich funkcji dystrybucyjnych o skończonych wymiarach, gdzie n przyjmuje wartości 1, 2 i tak dalej. (Fisz, 1967).

Funkcja $X(t) = X(t, \omega')$ reprezentuje realizację procesu, gdzie ω' jest zdarzeniem elementarnym w Ω , które jest stałe.

W dziedzinie teorii szeregów czasowych podstawowym pojęciem jest pojęcie stacjonarności. Stacjonarność sensu stricto procesu stochastycznego $X(t)$ charakteryzuje się tym, że jego n -wymiarowe rozkłady pozostają stałe w czasie. Według Fisz (1967) następująca zależność zachodzi dla każdego n , każdego podzbioru $\{t_1, t_2, \dots, t_n\}$ należącego do T będącego podzbiorem R i dla każdego τ .

Formuła $F_{t_1, t_2, \dots, t_n}(x_1, x_2, \dots, x_n) = F_{t_1 + \tau, t_2 + \tau, \dots, t_n + \tau}(x_1, x_2, \dots, x_n)$, gdzie τ reprezentuje przesunięcie czasowe zastosowane do zmiennych wejściowych od x_1 do x_n .

w której

Zbiór $\{t_1 + \tau, t_2 + \tau, \dots, t_n + \tau\}$ jest podzbiorem T .

Jeżeli proces stochastyczny $X(t)$ wykazuje ściśle sensowną stacjonarność, można wywnioskować, że odpowiednie charakterystyki są należycie spełnione.

Przedstawiono równanie $F_t(x) = F_0(x)$ i równanie $F_{t_1, t_2}(x_1, x_2) = F_{0, t_2 - t_1}(x_1, x_2)$.

Dane dotyczące równości sugerują, że proces stacjonarny w najściślejszym tego słowa znaczeniu wykazuje jednowymiarowe funkcje rozkładu, które pozostają spójne dla wszystkich wartości parametru t . Dodatkowo dwuwymiarowe dystrybuanty takiego procesu zależą wyłącznie od różnicy między wartościami $t_2 - t_1$ parametru t (czas).

W kontekście procesów stochastycznych proces $X(t)$ jest uważany za stacjonarny w bardziej ogólnym sensie, jeśli jego wartość oczekiwana i wariancja pozostają stałe w czasie, a jego funkcja kowariancji zależy wyłącznie od różnicy czasu między momentami t_2 i t_1 , co oznaczamy jako τ .

$$\begin{aligned}m_x(t) &= m_x = \text{const}, \\ \sigma_X^2 &= \sigma_X^2 = \text{const}, \\ K_X(t_2 - t_1) &= K_X(\tau).\end{aligned}$$

Równanie $m_x(t) = m_x$ reprezentuje stałą wartość, podczas gdy $\sigma_X^2 = \sigma_X^2$ pozostaje stałe. Dodatkowo $K_X(t_2 - t_1) = K_X(\tau)$ oznacza zależność między K_X a przedziałem czasu ($t_2 - t_1$).

Funkcja kowariancji procesu stacjonarnego spełnia w szerszym znaczeniu następujące warunki:

W chwili $\tau = 0$ początkowa wartość funkcji kowariancji procesu $K_X(0)$ jest równoważna stałej σ_X^2 . Oznacza to, że funkcja kowariancji procesu pozostaje skończona i nie zależy od czasu.

Funkcja $K_X(\tau)$ wykazuje symetrię względem τ , o czym świadczy równość $K_X(\tau) = K_X(-\tau)$.

nierówność $|K_X(\tau)| \leq K_X(0)$, przy czym wartość bezwzględna $K_X(\tau)$ maleje wraz ze wzrostem τ .

Znormalizowana funkcja kowariancji jest często stosowana w praktycznych zastosowaniach dla procesów stacjonarnych.

Równanie $k_X(\tau) = \frac{K_X(\tau)}{K_X(0)}$ jest prawdziwe, gdy wartość bezwzględna $k_X(\tau)$ jest mniejsza lub równa 1.

Omawiane wartości dotyczą współczynników korelacji między zmiennymi losowymi $X(t)$ i $X(t + \tau)$. Zgodnie z trzecim warunkiem funkcji autokowariancji można wnioskować, że korelacja między wyrazami procesu stacjonarnego wykazuje tendencję spadkową wraz ze wzrostem odstępów czasu między tymi wyrazami.

Ponieważ wartość bezwzględna τ dąży do nieskończoności, funkcja $k_X(\tau)$ dąży do zera.

O spełnieniu warunków stacjonarności w szerokim sensie można wnioskować, jeśli proces stochastyczny jest stacjonarny w sensie wąskim i jeśli istnieją momenty drugiego rzędu. Twierdzenie to zostało sformułowane przez Dooba (1953), Fisz (1967) i Papoulisa

(1972).

Uważa się, że proces ma czas dyskretny, jeśli jego przejścia z jednego stanu do drugiego są możliwe wyłącznie w ściśle określonych momentach t_1, t_2, \dots, t_n . W przerwach między tymi momentami proces pozostaje w poprzedniej pozycji. Proces, który może przejść z jednego stanu do drugiego w dowolnym nieznanym czasie, nazywany jest procesem ciągłym.

Proces deterministyczny charakteryzuje się tym, że jest jednoznacznie zdeterminowany w każdym momencie. Cechą charakterystyczną tego zjawiska jest możliwość prognozowania przyszłych wartości procesowych na podstawie aktualnej próby. Trend liniowy jest przykładem procesu deterministycznego.

Według Mańczaka (1970) procesy ergodyczne odnoszą się do tych procesów, których cechy probabilistyczne post factum mogą być reprezentowane przez cechy utrwalone w dłuższym czasie.

Aby umożliwić praktyczne zastosowanie teorii procesów stochastycznych, konieczne jest ustalenie probabilistycznych właściwości procesów opartych na pojedynczej, odpowiednio długiej realizacji procesu. Warunek ergodyczności w procesie uznano za konieczny na podstawie badań przeprowadzonych przez Mańczaka w 1970 roku.

1.2.3 Modele autoregresji

Modele autoregresji to modele statystyczne, które wykorzystują wcześniejsze obserwacje szeregu czasowego do przewidywania przyszłych wartości tego samego szeregu. Modele te są powszechnie stosowane w różnych dziedzinach, w tym w ekonomii, finansach i inżynierii, do analizowania i prognozowania danych zależnych od czasu. Modele autoregresji opierają się na założeniu, że przyszłe wartości szeregów czasowych są liniowo powiązane z ich przeszłymi wartościami i można je formułować przy użyciu różnych technik matematycznych, takich jak regresja liniowa, oszacowanie największej wiarygodności i wnioskowanie bayesowskie. Dokładność modeli autoregresyjnych zależy od jakości i ilości dostępnych danych, a także trafności założeń i parametrów modelu.

Autoregresja to technika statystyczna polegająca na regresji przyszłej wartości sekwencji na podstawie jej przeszłych wartości, przy czym te ostatnie służą jako zmienne objaśniające.

Modele autoregresji, zgodnie z definicją Boxa i Jenkinsa (1983), Niederlińskiego (1979) i Welfe (2009), odnoszą się do modeli, w których obecne wartości szeregu są reprezentowane jako liniowe funkcje wcześniejszych obserwacji.

Jeśli obecna wartość ciągu x_n jest zależna wyłącznie od jednej poprzedzającej wartości w sposób liniowy.

Jeśli model oparty jest na poprzedniej wartości zmiennej, oznaczanej jako x_{n-1}

, można go sklasyfikować jako model autoregresji pierwszego rzędu i reprezentować w literaturze naukowej jako AR(1).

Model autoregresji rzędu p , AR(p), charakteryzuje się liniową zależnością bieżącej wartości ciągu x_n od poprzedzających go wartości px_{n-1}, \dots, x_{n-p} .

Proces autoregresji skończonego rzędu, gdzie p jest większe od 1, można wyrazić za pomocą następującego równania:

$$x_n = a_1x_{n-1} + a_2x_{n-2} + \dots + a_px_{n-p} + \varepsilon_n, \quad (1.4)$$

Równanie przedstawione przez użytkownika można przedstawić naukowo w następujący sposób: x_n jest równe sumie $a_1x_{n-1} + a_2x_{n-2}$ i tak dalej, aż do a_px_{n-p} , z dodatkiem składnika błędu reprezentowanego przez ε_n .

Gdzie:

Model procesu charakteryzuje się parametrami p , oznaczonymi jako od 1 do p . Dodatkowo proces obejmuje proces absolutnie losowy, określany jako szum biały, oznaczany jako ε_n . Biały szum ma określone właściwości.

$$\mu[\varepsilon_n] = 0,$$

$$re_2[\varepsilon_n] = \sigma\varepsilon \leq \infty$$

Kowariancja między ε_n i ε_{ns} jest równa zero, gdzie s nie jest równe zero.

Artykuł ten ma na celu przedstawienie akademickiej dyskusji na temat stochastycznego procesu autoregresji pierwszego rzędu AR(1) i związanych z nim właściwości.

Proces AR(1) jest rodzajem modelu procesu stochastycznego w dyskretnej przestrzeni czasowej, opisanego przez Welfe (2009) i Walaszek Babiszewska (2015).

Równanie 1.5 przedstawia rekurencyjną zależność między x_n i jego poprzednią wartością, x_{n-1} , pomnożoną przez stałą a i dodaną do składnika błędu ε_n . Zależność ta zachodzi dla wartości n w zakresie od 2 do N .

Zmienna „ a ” reprezentuje parametr procesu, który nie został jeszcze zidentyfikowany, podczas gdy „ ε_n ” oznacza absolutnie losowy proces, który charakteryzuje się białym szumem i pozostaje nieznanym.

Aby obliczyć przewidywaną wartość procesu $E[x_n]$, należy wykonać przekształcenie wzoru (1.5) poprzez podstawienie x_{n-1}, x_{n-2} , i tak dalej. Strukturę powyższej procedury oznaczono jako (1.5).

$$x_n = ax_{n-1} + \varepsilon_n, n = 2, \dots, N, \quad (1.5)$$

Równanie $x_n = ax_{n-1} + \varepsilon_n$ można rekurencyjnie wyrazić jako $x_n = a(ax_{n-2} + \varepsilon_{n-1}) + \varepsilon_n$ i tak dalej. Równanie można wyrazić w następujący sposób:

$$x_n = ax_{n-1} + \varepsilon_n = a(ax_{n-2} + \varepsilon_{n-1}) + \varepsilon_n = \dots = a^m x_{n-m} + a^{m-1} \varepsilon_{n-(m-1)} + \dots + \varepsilon_n \quad (1.6)$$

Zakładając, że $|a| < 1$ $\lim_{m \rightarrow \infty} a^m = 0$. Ponadto wyrażenie 1.6 implikuje, że $E[x_n] = 0$.

Dowody empiryczne sugerują, że wariancję procesu AR(1) można wyrazić następująco: $|a| < 1$.

Równanie oznaczone jako

$$D^2[X_n] = \sigma_X^2 = \sigma_\varepsilon^2 \frac{1}{1 - a^2} \quad (1.7)$$

przedstawiono w kontekście akademickim.

Wariancja procesu AR(1) jest zależna zarówno od wariancji szumu, jak i parametru a .

Według Welfe (2009), gdy wartość parametru kowariancji procesu AR(1), $cov[x_n, x_{n-m}]$, wynosi $|a| < 1$, maleje z potęgą m .

Podane równanie pokazuje kowariancję między x_n i x_{n-m} ,

$$cov[x_n, x_{n-m}] = \sigma_\varepsilon^2 \frac{a^m}{1 - a^2} = \sigma_X^2 a^m, \quad m = 0, 1, 2, \dots \quad (1.8)$$

Znormalizowaną funkcję autokorelacji, oznaczoną jako $k_x(m) = k_x(x_n, x_{n-m})$, można przedstawić w postaci macierzowej.

$$k_x(m) = \begin{bmatrix} 1 & a & a^2 & \dots & a^{m-1} \\ a & 1 & a & \dots & a^{m-2} \\ a^2 & a & 1 & \dots & a^{m-3} \\ \dots & \dots & \dots & \dots & \dots \\ a^{m-1} & a^{m-2} & a^{m-3} & \dots & 1 \end{bmatrix} \quad (1.9)$$

Tematyka dotyczy stochastycznego procesu autoregresji drugiego rzędu, a konkretnie procesu AR(2).

Model AR(2) jest rodzajem modelu procesu stochastycznego w dyskretnej przestrzeni czasu, który należy do klasy procesów autoregresyjnych drugiego rzędu. Można to wyrazić w następującej postaci:

$$x_n = ax_{n-1} + bx_{n-2} + \varepsilon_n, \quad n = 3, \dots, N \quad (1.10)$$

Równanie (1.47) przedstawia matematyczną zależność między x_n i poprzedzającymi go wyrazami, x_{n-1} i x_{n-2} , jak również składnik błędu, ε_n . Współczynniki a i b są również zawarte w tym równaniu. To równanie jest ważne dla wartości n w zakresie od 3 do N .

Parametry modelu, oznaczone przez a i b , podlegają wpływowi stochastycznego procesu białego szumu, reprezentowanego przez ε_n .

Modele autoregresyjne są wykorzystywane do charakteryzowania procesów stacjonarnych. Cechą definiującą sekwencje czasowe, które są stacjonarne, jest to, że ich funkcje probabilistyczne pozostają stałe nawet po przesunięciu w czasie.

Według Sveshnikova (1965) i Sobczyka (1996) funkcja korelacji jest zawsze obecna w procesie stochastycznym drugiego rzędu.

Modele średniej ruchomej (MA) to rodzaj modelu statystycznego powszechnie stosowanego w analizie szeregów czasowych. Polegają one na obliczeniu średniej serii punktów danych w określonym przedziale czasu, przy czym wynikowa wartość reprezentuje „średnią ruchomą” dla tego okresu. Modele MA są przydatne do identyfikowania trendów i wzorców w danych i mogą być używane do prognozowania przyszłych wartości na podstawie wyników z przeszłości.

Proces średniej ruchomej można scharakteryzować jako model regresji liniowej, który wiąże obecną wartość szeregu czasowego z przeszłymi składnikami błędów, które są związane z białym szumem lub przypadkowymi zakłóceniami. Zwykle przyjmuje się, że identyczne rozkłady, zazwyczaj rozkład normalny ze stałą wariancją i wartością oczekiwaną równą zero, są źródłem losowych zaburzeń w każdym punkcie. Model ten wykazuje charakterystyczną cechę, zgodnie z którą perturbacje stochastyczne rozchodzą się na kolejne wartości szeregów czasowych, jak zauważyli Niederliński, Kasprzyk i Figwer (1997).

Według Welfe (2009) model MA(q), który obejmuje średnią ruchomą rzędu q , można wyrazić w następującej postaci.

Powyższe równanie przedstawia wartość x w chwili n , gdzie jest funkcją składowej błędu w chwili $n(\varepsilon_n)$ i składowej błędu w dwóch poprzednich okresach czasu (ε_{n-1} i ε_{n-2}), każdy pomnożony przez odpowiedni współczynnik

$$x_n = \varepsilon_n + b_1\varepsilon_{n-1} + b_2\varepsilon_{n-2} + \dots + b_q\varepsilon_{n-q}, \quad (1.11)$$

Powyższe równanie odnosi się do modelu z parametrami q oznaczonymi jako b_1 do b_q , gdzie ε_n reprezentuje proces białego szumu w czasie dyskretnym.

Niniejszy dyskurs będzie dotyczył procesu średniej ruchomej i odpowiadających mu właściwości, ze szczególnym uwzględnieniem fundamentalnej formuły modelu MA(1).

$$x_n = \tau_n + b\tau_{n-1}, \quad (1.12)$$

Położenie τ_n jest procesem białego szumu charakteryzującym się właściwościami, że jego wartość oczekiwana wynosi zero, oznaczana jako $E[\tau_n] = 0$.

$$re^2[\tau_n] = \theta_\tau^2,$$

$$cov(\tau_n, \tau_{n-s}) = 0, \quad s \neq 0$$

Wyznaczenie wartości oczekiwanej, wariancji i funkcji kowariancji procesu MA(1) można osiągnąć poprzez wykorzystanie odpowiednich definicji, jak nakreślono w Welfe

(2009) i Walaszek-Babiszewskiej (2015).

Równanie $E(x_n) = E(\tau_n) + bE(\tau_{n-1}) = 0$ jest prawdziwe.

Równanie $D^2(x_n) = D^2(\tau_n) + b^2D^2(\tau_{n-1}) = 0$ Można zaobserwować $\theta_\tau^2(1 + b^2)$.

Kowariancję między x_n i x_{n-1} można wyrazić jako wartość oczekiwaną iloczynu różnicy między x_n a jego wartością oczekiwaną oraz różnicy między x_{n-1} a jego wartością oczekiwaną. Można to dalej uprościć do oczekiwanej wartości iloczynu x_n i x_{n-1} .

Obliczane jest wyrażenie $E[(\tau_n + b\tau_{n-1})(\tau_{n-1} + b\tau_{n-2})] =$.

Wyrażenie matematyczne $E[(\tau_n\tau_{n-1} + b\tau_{n-1}^2 + b\tau_n\tau_{n-2} + b^2\tau_{n-1}b\tau_{n-2})]$ można uprościć do $E[b\tau_{n-1}^2]$. Uproszczenie to opiera się na założeniu, że kowariancja między x_n i x_{n-s} jest równa zero dla s większych od jeden, oraz że b i θ są stałymi.

Proces średniej ruchomej rzędu początkowego wykazuje zerową średnią i skończoną wariancję, która jest zależna zarówno od wariancji zaburzeń, jak i parametru modelu b . Proces MA(1) można scharakteryzować jako nieskorelowany proces stochastyczny, podobny do białego szumu. Model średniej ruchomej ma zalety i wady:

Jedną z zalet tego podejścia jest wygoda i wydajność, jaką oferuje w wykonywaniu obliczeń matematycznych.

Wady obejmują niewystarczającą precyzję prognozy, utratę danych podczas procesu oraz niewystarczające uwzględnienie trendów i wpływów sezonowych.

Model autoregresji średniej ruchomej (ARMA) to model statystyczny stosowany w analizie szeregów czasowych. Łączy koncepcje modeli autoregresji (AR) i modeli średniej ruchomej (MA), aby stworzyć bardziej wszechstronny model do analizy danych szeregów czasowych. Model ARMA jest szeroko stosowany w różnych dziedzinach, w tym w finansach, ekonomii i inżynierii, do prognozowania przyszłych wartości szeregu czasowego na podstawie jego zachowania w przeszłości.

Model ARMA, który oznacza autoregresywną średnią ruchomą, jest strukturą matematyczną wykorzystywaną w analizie statystycznej w celu analizy i prognozowania stacjonarnych szeregów czasowych (Zhang i Moore, 2015; Chen, 2019).

Model ARMA składa się z dwóch podstawowych modeli szeregów czasowych, a mianowicie modelu autoregresji.

Modele AR i MA są powszechnie stosowane w analizie szeregów czasowych. Proces autoregresji skończonego rzędu s , gdzie s jest większe od 1, można przedstawić matematycznie za pomocą następującego wzoru:

$$x_n = a_1x_{n-1} + a_2x_{n-2} + \dots + a_sx_{n-s} + \varepsilon_n, \quad (1.13)$$

Podane równanie reprezentuje formułę rekurencyjną, w której x_n jest równe sumie iloczynu każdego współczynnika a_i i odpowiedniego składnika x_{n-i} , gdzie i mieści się w zakresie od 1 do liczby wyrazów w równaniu. Równanie (1.58) składa się ze zmiennych a, s, n i ε , gdzie ε oznacza składnik błędu.

Gdzie:

Parametry modelu są oznaczone jako od 1 do s , podczas gdy składnik losowy jest reprezentowany przez ε_n . Ten ostatni jest procesem stochastycznym charakteryzującym się białym szumem.

Proces MA(q) charakteryzuje się średnią ruchomą rzędu q i można go wyrazić w następujący sposób:

$$x_n = \varepsilon_n + b_1\varepsilon_{n-1} + b_2\varepsilon_{n-2} + \dots + b_q\varepsilon_{n-q} \quad (3.11a)$$

Gdzie:

Parametry modelu są oznaczone jako b_1 do b_q , podczas gdy losowy składnik procesów białego szumu jest reprezentowany przez ε_n .

Proces autoregresyjny leżący u podstaw średniej ruchomej rzędu q , oznaczany jako $ARMA(s, q)$, jest złożony z dwóch procesów (Welfe, 2009). W szczególności proces można przedstawić jako

$$x_n = a_1x_{n-1} + a_2x_{n-2} + \dots + a_sx_{n-s} + \varepsilon_n + b_1\varepsilon_{n-1} + b_2\varepsilon_{n-2} + \dots + b_q\varepsilon_{n-q} \quad (1.14)$$

Modele ARMA są wykorzystywane do przedstawiania sekwencji czasowych, które są stacjonarne. Niektóre formy niestacjonarności można przekształcić w procesy stacjonarne, na przykład poprzez implementację procesów różnicowych i późniejsze wykorzystanie modeli ARMA. Ponadto w poniższej sekcji przedstawiono kategorię modeli autoregresyjnej zintegrowanej średniej ruchomej (ARIMA).

Skuteczność modeli ARMA przypisuje się ich oszczędnej reprezentacji. Funkcja autokorelacji jest zwykle używana do opisu cech modeli ARMA, podobnie jak modele AR i MA, jak zauważyli Chan i Ngai (2011) oraz Zhang i Moore (2015).

Model ARIMA jest metodą statystyczną powszechnie stosowaną w analizie szeregów czasowych.

Model ARIMA, który oznacza autoregresyjną zintegrowaną średnią ruchomą, jest rodzajem procesu stochastycznego, który łączy modele autoregresyjne i średniej ruchomej z integracją. Zintegrowane modele są powszechnie stosowane do przedstawiania procesów stochastycznych, które są niestacjonarne (Stellwagen i Tashman, 2013).

Model Autoregressive Integrated Moving Average (ARIMA) to narzędzie statystyczne, które wykorzystuje dane szeregów czasowych do prognozowania przyszłych wzorców.

Zintegrowany model autoregresji średniej ruchomej jest techniką statystyczną wykorzystywaną w analizie regresji do oceny wielkości zmiennej zależnej w stosunku do innych zmiennych. Powyższy model jest szeroko wykorzystywany do prognozowania przyszłych wartości danych. Osiąga się to poprzez analizę rozbieżności między wartościami obecnymi w ciągach znaków, a nie samych wartości oryginalnych.

Model ARIMA składa się z trzech podstawowych komponentów, a mianowicie autoregresji (AR), integracji (I) i średniej ruchomej (MA).

Autoregresja odnosi się do modelu statystycznego, który przedstawia regresję zmiennej na podstawie jej przeszłych wartości.

Według Jamesa Chena (2019) integracja odnosi się do procedury obliczania rozbieżności między nieprzetworzonymi obserwacjami w celu wygenerowania szeregów czasowych, które składają się z rozbieżności między punktami danych w sąsiadujących instancjach. W kontekście modeli ARIMA zwyczajowo oznacza się model jako ARIMA z parametrami p, d, q , gdzie p reprezentuje liczbę opóźnień zmiennej zależnej w składniku autoregresyjnym, d reprezentuje stopień zróżnicowania, a q reprezentuje liczbę opóźnień składnika błędu w składniku średniej ruchomej. Zmienna „ d ” reprezentuje stopień różnicy. Zapytanie dotyczy zakresu średniej ruchomej.

Model ARIMA można skonstruować za pomocą różnych metodologii, aby działał jako model ARMA lub alternatywnie jako proste modele AR, I lub MA. Według Jamesa Chena (2019) modele ARIMA są wykorzystywane do charakteryzowania hałasu w zakłóceniach.

Modele ARIMA są konstruowane w oparciu o procesy różnicowe, które są wyrowadzane z obserwacji procesu podstawowego $\{x_n, x_{n-1}, x_{n-2}, \dots, x_{n-m}, \dots\}$. Te procesy różnicowe służą jako podstawa do budowy modeli ARIMA.

Początkowa rozbieżność między tymi dwoma zjawiskami:

Wzór na pierwszą różnicę ciągu wyraża się następująco: $\Delta^1 X_n = x_n - x_{n-1}$, natomiast pierwsza różnica poprzedniego wyrazu jest dana wzorem $\Delta^1 X_{n-1} = x_{n-1} - x_{n-2}$.

Równanie przedstawione jako $\Delta^1 X_{n-m} = x_{n-m} - x_{n-m}$ jest wyrażeniem akademickim.

Proces drugich różnic polega na obliczeniu różnic między kolejnymi wyrazami ciągu, a następnie obliczeniu różnic między tymi różnicami. Można to wykorzystać do analizy wzorców i trendów w sekwencji.

Równanie $\Delta^2 X_n = \Delta^1 X_n - \Delta^1 X_{n-1}$ można wyrazić jako $x_n - x_{n-1} - (x_{n-1} - x_{n-2}) = x_n - 2x_{n-1} + x_{n-2}$.

Trzecia rozbieżność dotyczy procesu.

Równanie $\Delta^3 X_n = \Delta^2 X_n - \Delta^2 X_{n-1}$ można wyrazić jako $x_n - 2x_{n-1} + x_{n-2} - (\Delta^1 X_{n-1} - \Delta^1 X_{n-2})$. Dalsze uproszczenie daje $x_n - 2x_{n-1} + x_{n-2} - [x_{n-1} - x_{n-2} - (x_{n-2} - x_{n-3})]$. Można to zapisać jako $x_n - 3x_{n-1} + 3x_{n-2} - x_{n-3}$.

W przypadku, gdy proces różnic rzędu l , $\Delta^l X_n$ jest modelowany za pomocą schematu AR(p), kolejnym krokiem jest oszacowanie parametrów a_1, \dots, a_p , a następnie rozwiązanie wspomnianych układów równania w celu określenia podstawowego procesu.

ARX

Model autoregresyjny z wejściem zewnętrznym, potocznie nazywany ARX, należy do dostępnych dynamicznych modeli obiektowych. Model wejścia-wyjścia ARX ma zastosowanie do systemów charakteryzujących się obiektami stochastycznymi. W chwili n wyjście obiektu jest określane przez liniową kombinację jego przeszłych wartości i wejścia, które zostało opóźnione. Według Niederlińskiego (1979) model ARX wyraża się w określonej formie.

Wyrażenie „ $x_n + a_1x_{n-1} + \dots$ ” można przedstawić w języku akademickim jako równanie wielomianowe ze zmienną x podniesioną do różnych potęg i pomnożoną przez współczynniki. Przedstawione równanie ma charakter matematyczny, a konkretnie równanie wielomianowe. Obejmuje zmienne reprezentowane przez litery i współczynniki reprezentowane przez liczby. Równaniem można manipulować za pomocą operacji algebraicznych w celu rozwiązania nieznanymi zmiennymi. Modele ARX są powszechnie stosowane do charakteryzowania systemów, w których zmienna wyjściowa systemu jest oznaczona przez x , zmienna wejściowa systemu jest oznaczona przez u , a e reprezentuje proces całkowicie losowy. Określenie kolejności modelu ARX jest uzależnione od uwzględnienia ilości kroków wejściowych i wyjściowych, a także opóźnienia s .

Model autoregresji z wejściami egzogenicznymi (ARX) to podstawowe ramy charakteryzowania układów dynamicznych podlegających zaburzeniom stochastycznym. Wspomniane modele przedstawiają wyjście badanego procesu jako sumę regresji wcześniejszych obserwacji funkcji wejście-wyjście oraz białego szumu, który charakteryzuje błąd w równaniu (Soderstrom & Stoica, 1997). Kontekst stochastyczny, podobnie jak inne modele błędów równań, powstrzymuje się od formułowania wyraźnych założeń dotyczących źródła zgodności między wynikiem procesu a obserwacjami (Diversi i in., 2010).

TAR

Dyscyplina ekonometrii przyniosła dodatkowe kategorie modeli statystycznych, takie jak te odnoszące się do reprezentacji nieliniowych sekwencji czasowych. Przykładami takich modeli są progowe modele autoregresyjne (TAR) i modele autoregresyjne płynnego przejścia (STAR), jak zauważył Welfe (2009).

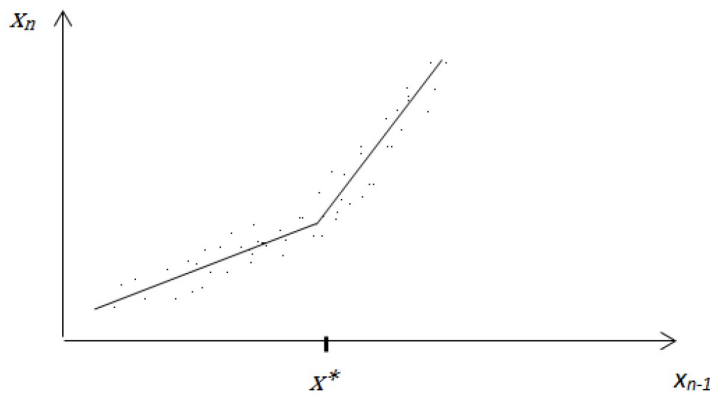
W dziedzinie prognozowania rzadko zdarza się, aby warunek stacjonarności procesu był spełniony, zwłaszcza jeśli chodzi o stałą wartość średnią. W konsekwencji modele autoregresji liniowej, które są ustalone dla pełnego zestawu obserwacji, nie są już odpowiednim obrazem.

Wiarygodne rozwiązanie obejmuje wdrożenie fragmentarycznego modelu liniowego, takiego jak progowy model autoregresji (TAR). W przypadkach, gdy istnieje powód, by sądzić, że określony obszar przestrzeni stanów procesu, oznaczony jako χ , charakteryzuje się modelami autoregresji wielokrotnej, zastosowanie modelu autoregresji progowej (TAR)

umożliwia integrację tych modeli lokalnych w ujednoczony model. Takie podejście umożliwia kompleksowy opis badanego procesu. Proces polega na podziale przestrzeni χ na wzajemnie wykluczające się przedziały na podstawie wybranej zmiennej, takiej jak zmienna niezależna x_{n-1} . W każdym przedziale tworzone są modele autoregresji $x_n^r = f(x_{n-s})$ dla r -tego przedziału. Proces przechodzenia z jednej linii regresji do drugiej nazywany jest funkcją przejścia, oznaczaną jako λ_r . Ta funkcja przyjmuje wartość 1 dla r -tej przedział i wartość 0 dla wszystkich pozostałych przedziałów. Model globalny można wyrazić w formie matematycznej w następujący sposób:

$$x_n = \sum_r \lambda_r x_n^r, \quad r = 1, 2, \dots, R \quad (1.15)$$

Model autoregresji AR(1) lub AR(2) wyznaczany jest dla wybranej zmiennej w r -tym przedziale, gdzie x i n są odpowiednimi parametrami. Progi modelu, oznaczone jako x^* , reprezentują granice przedziałów.



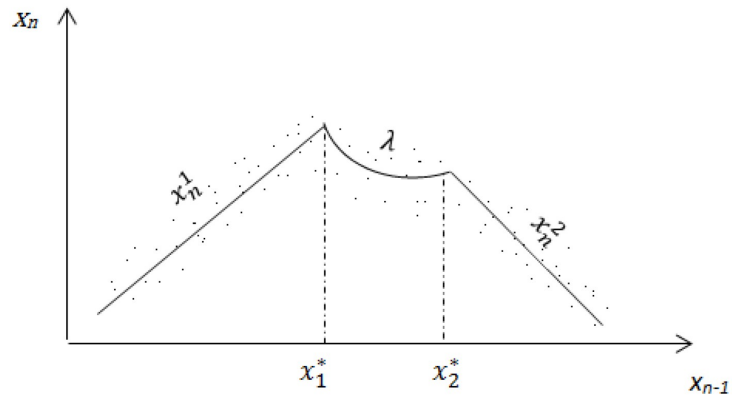
Rysunek 1.1: Przykład modelu TAR na tle zbioru obserwacji w układzie współrzędnych (x_n, x_{n-1})

Ilustracja przedstawiona na rysunku 1.1 przedstawia zastosowanie modelu autoregresyjnego szeregów czasowych (TAR) w korelacji ze zbiorem obserwacji w wyznaczonym układzie współrzędnych.

Model TAR jest złożeniem modeli autoregresyjnych, które są wykorzystywane do rozwiązywania nieliniowości w obliczeniach, zgodnie z sugestią Aznarte i Benitez (2013).

STAR

Model autoregresyjny, a konkretnie model Smooth Transition Autoregressive (STAR), zawiera ciągłe funkcje przejścia oznaczone jako $\lambda_r(b^r, x_{n-1}^r)$, które określają położenie i konfigurację przejścia wśród lokalnych modeli autoregresyjnych x_n^r . Funkcja $\lambda_r(b^r, x_{n-1}^r)$ podlega odwzorowaniu na przedział jednostkowy $[0, 1]$ i typowo wykazuje nieliniowość względem swoich zmiennych niezależnych (Walaszek-Babiszewska, Rydel i Kashpruk, 2018).



Rysunek 1.2: Przykład modelu STAR na tle zbioru obserwacji w układzie współrzędnych (x_n, x_{n-1})

Na rysunku 1.2 zilustrowano przykład modelu STAR w odniesieniu do zbioru obserwacji w układzie współrzędnych.

W zaawansowanej analizie finansowej modele STAR są wykorzystywane ze względu na ich płynne przełączanie reżimów w przeciwieństwie do dyskretnych przełączania reżimów. Modele te są wykorzystywane do analizy dynamiki ciągów czasowych w sytuacjach, w których decyzje inwestorów nie są podejmowane jednocześnie, jak zauważają Aznarte i Benitez (2013).

Tabela 1.1: Tabela porównawcza modeli parametrycznych

	Modele				Ilość zmiennych		Liczba parametrów		Modele procesów	
	Linijowe	Nieliniowe	Sygnatów	Obiektów	jedna	wiele	jeden	wiele	stacjonarne	niestacjonarne
AR	+		+		+		+		+	
MA	+		+		+		+		+	
ARMA	+		+		+			+	+	
ARIMA	+		+		+			+		+
ARX	+			+		+		+	+	
TAR		+	+		+			+	+	
STAR		+	+		+			+	+	

W tabeli 1.1 przedstawiono porównanie modeli parametrycznych procesów stochastycznych, które zostały omówione w tym rozdziale.

1.3 Wybrane metody identyfikacji procesów

Identyfikacja procesu lub systemu w kontekście automatyzacji obejmuje wyrowadzenie modelu matematycznego poprzez badania empiryczne. Identyfikacja służy ustaleniu zależności między zmiennymi obiektu w stanie równowagi lub w czasie trwania procesów przejściowych. Przedmiotem rozważań jest rozeznanie atrybutu stałego lub zmiennego (prototypu). Zazwyczaj zakłada się określoną kategorię modeli teoretycznych. Włączenie kryterium identyfikacji wynika z procesu porządkowania w ramach klasy wzorcowej, co zauważają Mańczak (1971), Bubnicki (1974) oraz Mańczak i Nahorski (1983).

Niniejszy rozdział ma na celu zbadanie procesu identyfikacji modeli statystycznych. W artykule zostanie przedstawiona metoda najmniejszych sum kwadratów błędów modelu. Metoda ta jest powszechnie stosowana jako kryterium wyboru modelu parametrycznego, obok metody największej wiarygodności. Wybór modelu odbywa się poprzez

identyfikację parametrów, które pochodzą z określonego zbioru danych.

W pracy wyprowadzono wzory do estymacji modeli AR(1) i AR(2) oraz przyjęto miary oceny dopasowania modeli do danych eksperymentalnych. Miary te obejmowały funkcję autokorelacji reszt modelu. Dodatkowo w badaniu wyprowadzono wzory na składowe trendu modelu. Proces identyfikacji modeli AR(1) i AR(2) zilustrowano za pomocą diagramu, który następnie posłużył jako podstawa algorytmu obliczeniowego.

W prezentowanym rozdziale przedstawiono przykłady obliczeń identyfikacyjnych dotyczących modeli AR(1) i AR(2) w różnych empirycznych szeregach czasowych. Zbiory danych to instancje procesów stochastycznych, które mają dyskretny czas i wykazują różne cechy.

Ponadto opracowano procedurę obliczeniową w celu ustalenia parametrów modelu stochastycznego ARX dla systemu. W niniejszym badaniu jako ilustrację identyfikacji wykorzystano zestawy obserwacji dwóch zmiennych odnoszących się do pojedynczego procesu technologicznego.

1.3.1 Metoda najmniejszych kwadratów jako metoda estymacji parametrów modeli parametrycznych

Metoda najmniejszych kwadratów błędów modelu jest kryterium wyboru modelu z danej klasy modeli parametrycznych. Ta metoda ułatwia identyfikację odpowiednich parametrów modelu, które są zgodne z danymi eksperymentalnymi.

Powyższa technika została scharakteryzowana jako składnik konwencjonalnego podejścia do analizy regresji, w którym uwzględnia się założenia Gaussa-Markowa dotyczące stochastycznych charakterystyk zmiennych modelowych. Powyższe założenia ułatwiają wyprowadzanie wnioskowania statystycznego dotyczącego estymatorów modelowych, jak postulują Mańczak (1971) i Welfe (2009). Metoda najmniejszych kwadratów nie wymaga takich założeń i ułatwia wyznaczenie parametrów modelu liniowego.

Wykorzystanie analizy regresji, w połączeniu z ustalonymi rozkładami warunków Gaussa-Markowa, ułatwia ocenę istotności równania regresji oraz istotności i współczynników \hat{a} i \hat{b} (Mańczak, 1971; Welfe, 2009).

W niniejszym badaniu zastosowano wyłącznie metodę najmniejszych kwadratów ze względu na rzadkie występowanie rozkładów normalnych zmiennych w badaniach eksperymentalnych, co jest warunkiem wstępnym analizy regresji.

Zmienna y jest zdefiniowana jako wartość oczekiwana y przy danym x i jest modelowana przy użyciu postaci liniowej, a konkretnie linii regresji pierwszego rzędu.

$$\hat{y} = E[y|x], \quad (1.16)$$

Oczekiwana wartość y przy danym x jest reprezentowana przez $E\{y|x\}$ i może być wyrażona jako liniowa kombinacja x ze współczynnikami a i b .

$$E\{y|x\} = \hat{a}x + \hat{b}. \quad (1.17)$$

Rozważmy dany zbiór obserwacji $\{(x_i, y_i)\}$ $i = 1, 2, \dots, N$, gdzie x oznacza zmienną niezależną, a y zależną zmienną losową. Celem jest wyznaczenie warunkowej wartości oczekiwanej, która spełnia określony warunek.

Celem jest zminimalizowanie kwadratu różnicy między wartościami przewidywanymi ($ax_i + b$) a wartościami rzeczywistymi (y_i).

Parametry modelu \hat{a} i \hat{b} są estymowane w procesie minimalizacji sumy kwadratów błędów w modelu.

$$E[e_i]^2 \cong \frac{1}{N} \sum [y_i - (\hat{a}x_i + \hat{b})]^2 \xrightarrow{a,b} \min \quad (1.18)$$

Równanie (1.47) ma na celu zminimalizowanie sumy kwadratów reszt poprzez oszacowanie parametrów a i b w modelu regresji liniowej, gdzie $E[e_i]^2$ jest w przybliżeniu równe sumie kwadratów różnic między obserwowanymi wartościami y_i a przewidywanymi wartościami ($ax_i + b$).

Warunek, który jest wymagany dla obecności ekstremum, daje w wyniku zestaw równań algebraicznych.

$$E[y_i - (\hat{a}x_i + \hat{b})]^2 \xrightarrow{a,b} \min.$$

Powyższe równanie przedstawia sumę kwadratów reszt w modelu regresji liniowej, gdzie N to liczba obserwacji, x_i i y_i to odpowiednio zmienne predykcyjne i odpowiedzi, a a i b to współczynniki nachylenia i wyrazu wolnego. Celem analizy regresji jest zminimalizowanie tej sumy kwadratów reszt w celu uzyskania najlepiej dopasowanej linii, która wyjaśnia związek między predyktorem a zmiennymi odpowiedzi.

$$\begin{aligned} \frac{d}{d\hat{a}} \left\{ \frac{1}{N} \sum_{i=1}^N [y_i - (\hat{a}x_i + \hat{b})]^2 \right\} &= 0 \\ \frac{d}{d\hat{b}} \left\{ \frac{1}{N} \sum_{i=1}^N [y_i - (\hat{a}x_i + \hat{b})]^2 \right\} &= 0 \end{aligned}$$

Powyższe równanie przedstawia wyrażenie matematyczne, w którym suma kwadratów różnic między wartościami obserwowanymi a wartościami przewidywanymi jest minimalizowana. Zmienne użyte w tym równaniu obejmują N , które reprezentują całkowitą liczbę obserwacji, x_i i y_i , które reprezentują odpowiednio zmienne predykcyjne i odpowiedzi, oraz a i b , które reprezentują współczynniki modelu regresji liniowej. Celem tego równania jest znalezienie wartości a i b , które najlepiej pasują do obserwowanych danych.

Po wykonaniu określonych procedur wynik jest następujący:

$$\begin{cases} \hat{a} \sum_{i=1}^N x_i^2 + \hat{b} \sum_{i=1}^N x_i = \sum_{i=1}^N x_i y_i \\ \hat{a} \sum_{i=1}^N x_i + \hat{b} N = \sum_{i=1}^N y_i \end{cases}$$

Równanie przedstawione przez użytkownika jest wyrażeniem matematycznym polegającym na zsumowaniu iloczynu dwóch zmiennych, x_i i y_i , gdzie i wynosi od 1 do N . Zmienne a i b to współczynniki mnożone przez sumę x_i do kwadratu i odpowiednio sumowanie x_i .

Powyższe równanie przedstawia model regresji liniowej, gdzie P to liczba punktów danych, N to liczba zmiennych niezależnych, x_i to i -ta zmienna niezależna, y_i to i -ta zmienna zależna, a i b to współczynniki modelu.

W celu uproszczenia zapisu stosujemy podstawienia literowe A, B, C i D w układzie równań algebraicznych.

$$\begin{cases} \hat{a}C + \hat{b}A = D \\ \hat{a}A + \hat{b}N = B \end{cases}$$

Przedstawione równanie jest układem liniowym dwóch zmiennych, w którym współczynniki zmiennych są reprezentowane przez stałe a i b . Prawa strona równania jest oznaczona stałą D.

Przedstawione równanie jest wyrażeniem matematycznym w postaci kombinacji liniowej, gdzie współczynniki zmiennych „ a ” i „ b ” są reprezentowane odpowiednio przez symbole „+” i „ N ”. Wynikowa wartość tego wyrażenia jest oznaczona symbolem „ B ”.

Układ równań algebraicznych należy rozwiązywać za pomocą wyznaczników.

$$W = \begin{vmatrix} C & A \\ A & N \end{vmatrix} = CN - A^2, \quad W \neq 0.$$

$$W_{\hat{a}} = \begin{vmatrix} D & A \\ B & N \end{vmatrix} = DN - BA,$$

$$W_{\hat{b}} = \begin{vmatrix} C & D \\ A & B \end{vmatrix} = DB - AD.$$

Ostatecznie poszukiwane estymatory \hat{a} , \hat{b} wynoszą:

$$\hat{a} = \frac{W_{\hat{a}}}{W}, \quad \hat{b} = \frac{W_{\hat{b}}}{W}.$$

Równanie regresji oznaczone jako (1.46) jest zdefiniowane przez przypisane estymatory.

Zastosowanie zasady najmniejszych kwadratów rozciąga się na identyfikację modeli dynamicznych. Analogicznie do modelu statycznego opisanego w sekcji 2.2, modele dynamiczne obejmują również obliczanie parametrów poprzez odpowiednie obliczenia. AR(1), AR(2).

Obliczenia do oszacowania modelu AR (1).

Model AR(1) jest specyficznym przypuszczeniem, które jest brane pod uwagę podczas analizy rzeczywistego procesu stochastycznego. Biorąc pod uwagę założenia o stacjonarności i ergodyczności dla badanego procesu, możliwe jest wyprowadzenie estymatora dla modelu AR(1) z obserwowanego zbioru danych $\{x_n\}$. Estymator \hat{a} w modelu AR(1) można obliczyć metodą najmniejszych kwadratów. Polega to na poszukiwaniu minimalnej wartości funkcji kryterium określonej przez:

$$\hat{y} = \hat{a}x + \hat{b} \quad (1.19)$$

Celem jest zminimalizowanie „ a ” w równaniu $Q = \sum_{n=2}^n (x_n - \hat{a}x_{n-1})^2$. Warunkiem istnienia ekstremum $\frac{dQ}{d\hat{a}} = 0$ jest:

$$Q = \sum_{n=2}^n (x_n - \hat{a}x_{n-1})^2 \rightarrow \min . \quad (1.20)$$

Po zastosowaniu różnych przekształceń wynikowy wynik to:

$$\frac{dQ}{d\hat{a}} = \sum_{n=2}^N 2(x_n - \hat{a}x_{n-1})(-x_{n-1}) = 0. \quad (1.21)$$

Wyrażenie $\sum_{n=2}^N (x_n - \hat{a}x_{n-1})x_{n-1} = 0$ oznacza równanie matematyczne.

Równanie $\sum_{n=2}^N x_n x_{n-1}$ można wyrazić jako $\hat{a} \sum_{n=2}^N x_{n-1}^2$. Parametr α w metodzie najmniejszych kwadratów jest szacowany przez estymator \hat{a} , który otrzymujemy z danych empirycznych $\{x_n\}$ za pomocą następującego wzoru:

$$\hat{a} = \frac{\sum_{n=2}^N x_n x_{n-1}}{\sum_{n=2}^N x_{n-1}^2} \quad (1.22)$$

Ponadto, aby lepiej zrozumieć jakość modelu, ustalane są kolejne metryki ocenianej procedury:

W danym zbiorze danych \bar{x} reprezentuje wartość średnią, podczas gdy D^2x reprezentuje wariancję. Dodatkowo e^2 oznacza sumę kwadratów reszt modelu, która jest równoważna sumie kwadratów błędów modelu.

$$e^2 = \sum_n (x_n - \hat{a}x_{n-1})^2, \quad (1.23)$$

Równanie (1.23) przedstawia sumę kwadratów różnic między każdym punktem danych (x_n) a oszacowaną wartością ($a \cdot \hat{x}x_{n-1}$), gdzie a jest stałą.

Średnia wartość danej zmiennej.

$$\bar{E} = \frac{1}{N-1} e^2. \quad (1.24)$$

Niniejsza praca dotyczy obliczeń estymatorów dla parametrów a i b modelu autoregresyjnego drugiego rzędu (AR(2)).

W sposób zbliżony do modelu AR(1), pod warunkiem, że badany rzeczywisty proces jest zarówno stacjonarny, jak i ergodyczny, możliwe jest wyprowadzenie estymatorów dla domniemanego modelu AR(2) poprzez wykorzystanie dostępnych danych z obserwacji procesu. Proces wyznaczania estymatorów parametrów a i b dla modelu AR(2) metodą najmniejszych kwadratów jest podobny do procesu wyznaczania modelu AR(1).

$$Q = \sum_{n=3}^N [x_n - (ax_{n-1} + bx_{n-2})]^2 \xrightarrow{a,b} \min \quad (1.25)$$

Celem podanego równania (1.25) jest zminimalizowanie sumowania kwadratów różnic między wartościami x przy indeksie n i liniową kombinacją x przy indeksach $n-1$ i $n-2$, pomnożonych przez współczynniki a i b , odpowiednio. Sumowanie odbywa się w zakresie n od 3 do N .

Wychodząc z założenia, że funkcja (1.25) musi mieć ekstremum, zostanie wyprowadzony układ dwóch równań algebraicznych ze względu na a i b .

Wyrażenie matematyczne podane przez użytkownika można przeformułować w sposób akademicki w następujący sposób:

$$\begin{cases} a \sum_{n=3}^N x_{n-1}^2 + b \sum_{n=3}^N x_{n-1}x_{n-2} = \sum_{n=3}^N x_n x_{n-1} \\ a \sum_{n=3}^N x_{n-1}x_{n-2} + b \sum_{n=3}^N x_{n-2}^2 = \sum_{n=3}^N x_n x_{n-2} \end{cases} \quad (1.26)$$

W celu uproszczenia zapisu proponuje się reprezentację wartości sum w układzie równań (1.26) jako A, B, C, D i E.

$$\begin{cases} aA + bB = C \\ aB + bE = D \end{cases} .$$

Przedstawione równanie jest równaniem chemicznym, gdzie A i B to reagenty, które łączą się, tworząc produkt C. Współczynniki a i b reprezentują odpowiednio stosunki stechiometryczne reagentów i produktu.

Przedstawione równanie jest wyrażeniem matematycznym w postaci kombinacji liniowej, gdzie zmienne a i b są współczynnikami mnożącymi odpowiednio zmienne „B” i „E”. Otrzymana suma tych iloczynów jest równa stałej „D”.

Powyższy układ równań algebraicznych zostanie rozwiązany z wykorzystaniem wyznaczników.

$$W = AE - B^2, W \neq 0$$

$$W_a = \begin{vmatrix} C & B \\ D & E \end{vmatrix} = CE - DB,$$

$$W_b = \begin{vmatrix} A & C \\ B & D \end{vmatrix} = AD - BC,$$

$$\hat{a} = \frac{W_a}{W},$$

$$\hat{b} = \frac{W_b}{W}.$$

Podane równanie to $C = CE - DB$.

Model AR(2) można wyrazić w następujący sposób:

$$x_n = \hat{a}x_{n-1} + \hat{b}x_{n-2}. \quad (1.27)$$

1.3.2 Empiryczne podejścia do wyboru modeli o charakterze praktycznym

Istnieją różne metody szacowania współczynników, w tym:

Zwykła procedura najmniejszych kwadratów jest metodą statystyczną powszechnie stosowaną w analizie regresji.

Równania Yule-Walkera, znane również jako metoda momentów, to technika stosowana w analizie statystycznej.

Metoda Burga.

Algorytm innowacji to proces generowania nowych pomysłów i rozwiązań problemów.

Hannana- Rissanena jest metodą obliczeniową stosowaną w modelowaniu statystycznym i analizie szeregów czasowych.

W kontekście analizy szeregów czasowych zwyczajowo stosuje się metodę Yule - Walkera lub Burga w celu uzyskania wstępnej estymacji modeli AR(p). I odwrotnie, algorytm innowacji lub algorytm Hannana - Rissanena są zwykle wykorzystywane w procesach MA(q). Po zakończeniu wstępnego procesu dopasowywania modelu, estymatory największej wiarygodności Gaussa są uzyskiwane poprzez wykorzystanie algorytmu innowacji. Jak stwierdził Sahu (2005) , możliwe jest łatwe uzyskanie przedziałów ufności dla oszacowanych współczynników .

Równania Yule-Walkera reprezentują podejście oparte na momentach, które obejmuje porównanie wartości kowariancji odnoszących się do analizowanych danych. Po podstawieniu oszacowania próbki dla funkcji autokorelacji i późniejszym rozwiązaniu równania, otrzymane oszacowanie parametrów jest oszacowaniem Yule-Walkera. Wzór na oszacowanie funkcji autokorelacji jest zilustrowany w następujący sposób:

$$\gamma_m = E[X_{t-m}X_t] = \sum_{j=1}^p \varphi_j E[X_{t-m}X_{t-j}] + E[X_{t-m}Z_t] = \sum_{j=1}^p \varphi_j \gamma_{|m-j|} + \sigma^2 \delta_m \quad (1.28)$$

Biorąc pod uwagę $m > 0$, rozważ następujący układ równań:

$$\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \dots \\ \gamma_p \end{bmatrix} = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \dots \\ \gamma_1 & \gamma_0 & \gamma_1 & \dots \\ \gamma_2 & \gamma_1 & \gamma_0 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \varphi_j \\ \varphi_j \\ \dots \\ \varphi_j \end{bmatrix}$$

Układ odwraca się, aby otrzymać wynik φ_j dla $j > 1$, dzięki estymacji γ_i . Odwracalność macierzy można przypisać jej dodatniości. Następnie równanie jest wykorzystywane do podstawienia w celu określenia σ^2 . Wspomniana metodologia jest powszechnie określana w literaturze akademickiej jako równania Yule-Walkera. Powyższe podejście pokazuje wykonalność zidentyfikowania modelu autoregresyjnego (AR), który odpowiada dowolnej ilości momentów w funkcji autokorelacji (ACF), jak opisano w równaniach Yule-Walkera (źródło: https://en.wikipedia.org/wiki/Autoregressive_model#Yule\T1\textendashWalker_equations).

Algorytm Burga jest dodatkową techniką estymacji parametrów. Niniejszy algorytm ma na celu zminimalizowanie błędów w predykcji liniowej poprzez wykorzystanie średniej harmonicznej współczynników korelacji cząstkowej. Niezmiennność odwrócenia czasu jest cechą charakterystyczną procesów liniowych. Podstawową zaletą tego algorytmu jest konsekwentne dostarczanie stabilnych modeli. Algorytm VOS FastBurg wykazał wyższą wydajność w porównaniu z metodą Yule-Walker w przypadkach, gdy rekordy danych są ograniczone. Ponadto skutecznie eliminuje potrzebę dokonywania kompromisów między wykorzystaniem obciążonych lub nieobciążonych oszacowań autokorelacji podczas dzielenia wag. Więcej informacji na temat tego algorytmu można znaleźć na stronie https://opus-codec.org/docs/vos_fastburg.

Hannana- Rissanena dzieli estymację parametrów na odrębne etapy.

W grę wchodzi dwa odrębne etapy.

Do danych dopasowano model autoregresji wyższego rzędu (AR) z m parametrami, gdzie m jest większe niż maksimum p i q . Ten konkretny model jest wykorzystywany

do określenia panujących warunków akustycznych.

$$\epsilon_n = x_n - \sum_{j=1}^m \hat{a}_j x_{n-j} \quad (1.29)$$

Estymację parametrów (p, q) modelu ARMA w drugim kroku przeprowadza się metodą regresji liniowej metodą najmniejszych kwadratów $(x_{n-1}, \dots, x_{n-j}, \epsilon_{n-1}, \dots, \epsilon_{n-j})$ (Eckehard Olbricha, 2008).

Metoda maksymalnego oszacowania wiarygodności (MLE) jest stosowana dla modeli nieliniowych. Czynnikiem odróżniającym metodę oszacowania największej wiarygodności (MLE) od metody najmniejszych kwadratów jest to, że ta pierwsza uwzględnia rozkład prawdopodobieństwa początkowych składników p .

Estymacja parametru θ metodą największej wiarygodności polega na wyznaczeniu wartości parametru maksymalizującego funkcję wiarygodności. Technika szacowania największej wiarygodności (MLE) może wykazywać zmniejszoną wydajność obliczeniową, gdy jest stosowana do zbiorów danych o znacznej wielkości. Ogólnie rzecz biorąc, powszechną praktyką jest stosowanie jednej z wyżej wymienionych technik w celu uzyskania odpowiedniego oszacowania wstępnego, a następnie optymalizacja numeryczna w celu wyznaczenia oszacowania największej wiarygodności (MLE) (Kochanek, 2016).

Technika oszacowania maksymalnego prawdopodobieństwa (MLE) opiera się na konstrukcji funkcji wiarygodności, która jest wyprowadzana z wartości funkcji gęstości prawdopodobieństwa elementów składowych próby losowej. Proces estymacji parametrów polega na wyborze określonego zestawu parametrów rozkładu, co skutkuje maksymalizacją funkcji wiarygodności, jak zauważył Mańczak Dzendzelyuk, Kostiv i Rabyk (2013) oraz Brockwell i Davis (2016).

Kryteria stosowane do oceny.

W istniejącej literaturze udokumentowano różne metryki do oceny dokładności modeli, szczególnie w prognozowaniu. Metryki te są zwykle oparte na resztach lub błędach modeli. Dwie powszechnie stosowane miary w analizie statystycznej to błąd średniokwadratowy (MSE) i błąd średniokwadratowy (RMSE).

Dwie rozważane miary statystyczne to MAD, co oznacza średnie odchylenie bezwzględne, oraz RMSPE, co oznacza pierwiastek średniokwadratowego błędu procentowego.

Metryka znana jako MAPE, co oznacza średni bezwzględny błąd procentowy, została wprowadzona przez Niederlińskiego w 1979 roku.

Zmienna \hat{x}_t reprezentuje prognozowaną wartość ciągu czasowego w zadanym czasie t , natomiast x_t oznacza rzeczywistą wartość zjawiska w tym samym czasie t . Poniżej przedstawiono wyrażenia matematyczne dla każdego rodzaju błędu:

$$MSE = \frac{1}{N} \sum_t (x_t - \hat{x}_t)^2 \quad (1.30)$$

Wyrażenie odnosi się do pierwiastka kwadratowego ze średniej sumy kwadratów błędów prognozy dla N obserwacji.

Pierwiastek błędu średniokwadratowego (RMSE) można wyrazić jako

$$RMSE = \sqrt{\frac{1}{N} \sum_t (x_t - \hat{x}_t)^2} = \sqrt{MSE} \quad (1.31)$$

Średni błąd bezwzględny (MAE) jest miarą statystyczną stosowaną do oceny dokładności modelu predykcyjnego opartego na N obserwacjach. Jest obliczana jako średnia różnica bezwzględna między wartościami przewidywanymi a wartościami rzeczywistymi. Średnie odchylenie bezwzględne (MAD) jest reprezentowane przez wzór

$$MAD = \frac{1}{N} \sum_t |x_t - \hat{x}_t| \quad (1.32)$$

Pierwiastek kwadratowy z sumy kwadratów błędów względnych, wyrażony w procentach.

$$RMSPPE = \sqrt{\sum_t \left(\frac{x_t - \hat{x}_t}{x_t} \right)^2} \cdot 100\% \quad (1.33)$$

Średnia arytmetyczna wartości bezwzględnych błędów względnych dla N przypadków, wyrażona w procentach.

$$MAPE = \frac{1}{N} \sum_t \left| \frac{x_t - \hat{x}_t}{x_t} \right| \cdot 100\% \quad (1.34)$$

Początkowe trio kryteriów odnosi się do niedokładności jednostek miary procesu, a ich wielkość jest uzależniona od poziomu szczegółowości danych czasowych. Ostatnie dwa kryteria są oceniane przy użyciu jednostek względnych, co umożliwi porównanie modeli w różnych procesach. Podejście lingwistyczne jest czasami stosowane do określenia precyzji prognoz błędu względnego, przy czym prognoza jest wyrażona słownie. a) Jeśli wskaźnik dokładności jest poniżej 10%, można go uznać za wysoki.

Prognozę uważa się za korzystną, gdy wzrasta z 10% do 20%.

Średnio odsetek zadowolających efektów mieści się w przedziale od 20% do 40%.

Każdy poziom dokładności przekraczający 40% można uznać za poniżej normy i niezadowolający.

Przypisane wartości wykazują zmienność i są uzależnione od konkretnego celu analizy predykcyjnej.

Wybór granic miar dokładności opiera się na spełnieniu z góry określonych oczekiwań.

Niniejsze omówienie dotyczy kryteriów oceny dopasowania modelu, a mianowicie Akaike Information Criterion (AIC) i Bayesowskiego Kryterium Informacyjnego (BIC).

Kryteria informacyjne, AIC i BIC, są wykorzystywane do określenia optymalnego modelu regresji do dokładnego opisu odpowiednich szeregów czasowych.

Kryterium informacyjne Akaike (AIC) jest kryterium statystycznym wykorzystywanym do selekcji modeli o różnej liczbie predyktorów. Powyższy wskaźnik służy do oceny poziomu jakości modeli statystycznych zastosowanych do analizowanego zbioru danych. Zazwyczaj model, który zawiera większą liczbę predyktorów, zwykle daje bardziej precyzyjne prognozy, ale jest również bardziej podatny na przeuczenie. Przedstawiony poniżej wzór został wprowadzony przez Akaike w 1973 roku i jest powszechnie określany jako AIC.

$$AIC = \frac{2 * k}{n} + \ln \left(\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \right), \quad (1.35)$$

Przedstawiony wzór jest oznaczony jako AIC i jest wyrażony jako (1.35), gdzie n jest wielkością próby.

Zmienne x_i i \hat{x}_i oznaczają odpowiednio wartość prawdziwą ciągu i wartość wyjściową modelu. Długość ciągu czasowego jest oznaczona przez n , natomiast liczba parametrów przez k .

Bayesowskie Kryterium Informacyjne (BIC), zaproponowane przez Schwarzera w 1978 r., jest obliczane przy użyciu sumy kwadratów reszt zgodnie z następującym wzorem.

$$BIC = \frac{k * \ln n}{n} + \ln \left(\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \right), \quad (1.36)$$

Skrót „BIC” oznacza się jako (1.36) w kontekście danego badania lub analizy, gdzie „ n ” oznacza wielkość próby.

Zmiennymi, o których mowa, są x_i , reprezentujące obserwowaną wartość sekwencji, oraz \hat{x}_i , reprezentujące wartość wyjściową modelu. Dodatkowo n oznacza długość sekwencji czasowej, podczas gdy k reprezentuje liczbę zaangażowanych parametrów.

Wyżej wymienione kryteria informacyjne są powszechnie określane jako techniki porównywania modeli dla zmiennej odpowiedzi i są wykorzystywane do określenia optymalnego modelu. Zgodnie z przyjętymi konwencjami optymalny model wyznaczają najniższe wartości kryteriów AIC i BIC.

Kryterium informacyjne Bayesa (BIC) i kryterium informacyjne Schwarzera to kryteria statystyczne wykorzystywane do wyboru modelu z ograniczonego zestawu modeli. Za najkorzystniejszy uznawany jest model, który wykazuje najniższy BIC. Stwierdzenie zakłada, że omawiana koncepcja jest częściowo oparta na funkcji prawdopodobieństwa i wykazuje silną korelację z kryterium informacyjnym Akaike (AIC), jak zauważyli Chen i Chen (2008).

1.3.3 Funkcja autokorelacji reszt modeli

W trakcie modelowania i identyfikowania układów dynamicznych konieczne jest należyte uwzględnienie kontroli reszt modelu. W przypadku obiektów stacjonarnych odpowiednie są kryteria sumy błędów kwadratowych lub błędu średniokwadratowego. Niemniej

jednak konieczne jest przetestowanie resztkowych funkcji autokorelacji modelu sekwencji czasowej i porównanie ich z funkcjami autokorelacji standardowych procesów stochastycznych, jak sugerują Mańczak i Nahorski (1983) oraz Kashpruk (2016b).

Autokorelacja to zjawisko polegające na tym, że wpływ fluktuacji stochastycznych utrzymuje się poza określonym przedziałem czasu t i przenosi się na kolejne przedziały. Autokorelacja w modelu może być:

W przypadkach, w których występują sąsiednie przypadkowe składowe o tej samej polaryzacji, można zaobserwować pozytywny wynik.

Negatywność pojawia się w sytuacjach, w których istnieją przypadkowe komponenty, które wykazują naprzemienne znaki w bliskiej odległości od siebie.

Znormalizowana funkcja autokorelacji (Mańczak i Nahorski, 1983) zostanie wykorzystana do obliczenia funkcji autokorelacji reszt modelu.

$$k_e(s) = \frac{\sum_{n=m+1}^{N-s} (e_n - \bar{e})(e_{n+s} - \bar{e})}{\sqrt{\sum_{n=m+1}^{N-s} (e_n - \bar{e})^2} \sqrt{\sum_{n=m+1}^{N-s} (e_{n+s} - \bar{e})^2}} \quad (1.37)$$

gdzie: W kontekście modelowania zmienna e_n reprezentuje składową rezydualną modelu, podczas gdy m oznacza rząd modelu. Na przykład, gdy $m = 2$, modelem jest AR(2), gdzie e_n jest wyrażone jako różnica między $(ax_{n-1} + bx_{n-2})$. Dodatkowo s jest argumentem funkcji autokorelacji, gdzie s może przyjmować wartości od 0 do S , przy czym S jest mniejsze lub równe jednej trzeciej całkowitej liczby obserwacji, oznaczonej jako N , w sekwencji x_n .

Funkcja autokorelacji częściowej

Funkcja częściowej autokorelacji jest narzędziem statystycznym służącym do identyfikowania korelacji między zmienną a jej opóźnionymi wartościami przy jednoczesnym kontrolowaniu wpływu innych zmiennych.

Wcześniej zbadano funkcję autokorelacji, która zapewnia korelacje między bieżącymi obserwacjami czasowymi a obserwacjami we wszystkich poprzedzających punktach czasowych. Funkcja częściowej autokorelacji (PACF) jest składnikiem funkcji autokorelacji (ACF), jak sama nazwa wskazuje. Wykorzystanie funkcji autokorelacji (ACF) i częściowej funkcji autokorelacji (PACF) służy do określenia odpowiedniej kolejności modeli, jak stwierdził Cory Maklin w 2019 r. Funkcja częściowej autokorelacji (PACF) opisuje korelację między daną wartością x_n i jego poprzedzające wartości, pomijając wpływ opóźnień. Jego matematyczne wyrażenie jest następujące:

$$x_n = \mu + \rho_1 x_{n-1} + \dots + \rho_s x_{n-s} + e_n, \quad (1.38)$$

Funkcja częściowej autokorelacji (PACF) określa ilościowo stopień powiązania między dwiema obserwacjami czasowymi, biorąc pod uwagę wpływ innych punktów da-

nych. Obecna funkcja jest formułowana za pomocą obliczeń statystycznych korelacji częściowej, przy należywym uwzględnieniu wpływu opóźnień.

1.3.4 Komponenty związane z trendami

Autor często konceptualizuje każdą instancję szeregu czasowego jako zbiór odrębnych składników, a mianowicie trendu, składnika sezonowego i szumu. Wykorzystanie silnika analizy regresji i średniej ruchomej ułatwia identyfikację i badanie trendów. Analiza efektów sezonowych jest powszechnie prowadzona poprzez implementację modeli wygładzania sezonowego i autoregresji sezonowej, jak opisano w pracach Boxa i Jenkinsa (1970) oraz Niederlińskiego, Kasprzyka i Figwera (1979).

Analiza spektralna jest wykorzystywana do wykrywania zmian trendów, podczas gdy modele autoregresji lub modele harmoniczne są wykorzystywane do przedstawiania i przewidywania takich zjawisk. Przy prognozowaniu ciągów czasowych kluczowe jest przyjęcie założenia, że zaobserwowany w przeszłości trend jest wyraźny i w miarę spójny.

Proces wyboru linii trendu jest często złożonym i wieloaspektowym przedsięwzięciem. Ponadto możliwe jest, że ciąg testowy zachowuje elementy sezonowe i cykliczne. Elementy cykliczne różnią się od sezonowych zwiększoną trwałością i nieciągłością amplitudy. Według Welfe (2009) okresy składnika sezonowego są mierzone ilościowo w dniach, tygodniach lub miesiącach, podczas gdy okresy składnika cyklicznego są mierzone w latach, a nawet dziesięcioleciach, takich jak cykle koniunkturalne. Podstawowe techniki wykorzystywane do badania skomplikowanych szeregów czasowych obejmują średnią ruchomą, wygładzanie wykładnicze i modelowanie trendów.

Dowolną sekwencję czasową można przedstawić jako kombinację dwóch odrębnych składowych, a mianowicie deterministycznej i losowej.

$$x_n = f(u) + e_n, \quad (1.39)$$

Przedstawione równanie jest oznaczone jako (1.39) i wyraża związek między x_n , $f(u)$ i e_n .

Funkcję $f(u)$ można przedstawić jako linię trendu, podczas gdy zmienna e_n oznacza składnik losowy.

Możliwe jest rozbicie dowolnej obserwacji sekwencji czasowej na trzy odrębne czynniki:

Zmienna oznaczona jako T reprezentuje wzór lub tendencję, którą można zaobserwować w czasie.

Odwołuje się do pojęcia sezonowości (S).

Element stochastyczny, w szczególności składowa białego szumu.

Proces konstruowania tych modeli ostatecznie obejmuje identyfikację elementów składowych T , S i e dla każdego szeregu. W kontekście modeli addytywnych składowe T

wykazują właściwość addytywną, dzięki której się kumulują lub sumują.

$$X = T + S + e. \quad (1.40)$$

Multiplikatywny model sekwencji czasowych można wyrazić następującym równaniem:

$$X = T \cdot S \cdot e. \quad (1.41)$$

Model multiplikatywny zakłada, że elementy systemu są mnożone razem, aby uzyskać obserwowane wartości zmiennej docelowej.

Według Welfe (2009) techniki wykorzystywane do wyodrębniania trendów, wahań okresowych i wahań losowych z szeregów czasowych są powszechnie znane jako metody eliminacji szeregów czasowych lub metody dekompozycji szeregów czasowych.

1.3.5 Podejście Boxa-Jenkinsa do identyfikacji modeli na przykładzie modelu ARIMA

Metodologia Boxa-Jenkinsa jest szeroko stosowanym podejściem do identyfikacji modeli i w tym przypadku zostanie zastosowana do modelu Autoregressive Integrated Moving Average (ARIMA).

Istnieje wiele metodologii i podejść do prognozowania wielkości zmiennej zależnej od czasu. Wśród modeli, które zostały wzięte pod uwagę, jest ARIMA (Autoregressive Integrated Moving Average). Analizując takie modele, można podzielić szeregi czasowe na trzy odrębne komponenty: trend, który dotyczy długoterminowego wzrostu lub spadku danych; zmienność sezonowa, która odnosi się do wahań występujących w określonych okresach lub momentach; oraz szum, który oznacza przypadkowe zakłócenia pojawiające się w nieregularnych odstępach czasu.

Typowa procedura konstruowania modelu szeregów czasowych obejmuje szereg kroków, jak nakreślili Box i Jenkins (1983):

Proces oceny stacjonarności i sezonowości danego zbioru danych.

Identyfikacja potencjalnych propozycji modeli.

Jednym z powszechnych podejść w modelowaniu statystycznym jest stosowanie modeli dopasowywania, a następnie udoskonalanie procesu selekcji za pomocą AIC/BIC. Dodatkowo często przeprowadzane są testy diagnostyczne w celu oceny skuteczności modeli.

Generowanie prognoz.

Analiza Boxa-Jenkinsa to rygorystyczne podejście do identyfikacji, estymacji i oceny zintegrowanych modeli szeregów czasowych, które zawierają komponenty autoregresji i średniej ruchomej, powszechnie znane jako modele ARIMA. Podejście to ma za-

stosowanie do rozszerzonych i pośrednich szeregów czasowych, wymagających minimum 50 obserwacji.

Konstrukcję empirycznych modeli szeregów czasowych można przeprowadzić za pomocą sprawnej metodologii Boxa-Jenkinsa, która obejmuje trzyetapowy proces.

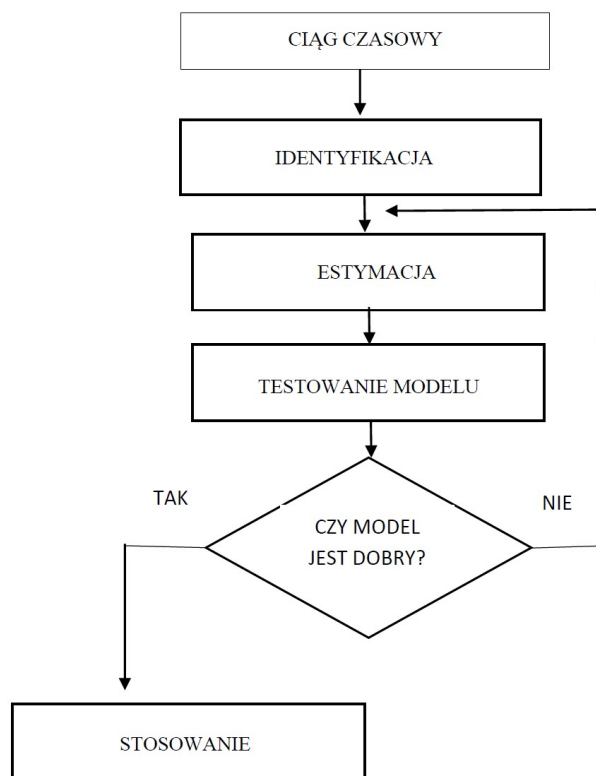
Proces specyfikacji lub identyfikacji modelu obejmuje wykorzystanie różnych narzędzi, takich jak wykresy danych, autokorelacje, autokorelacje częściowe i inne istotne informacje. W procesie tym wybiera się zestaw nieskomplikowanych modeli ARIMA i przeprowadza estymację odpowiednich wartości parametrów p , d i q .

Szacowanie modelu.

Proces diagnozowania lub testowania modeli obejmuje ocenę dopasowanego modelu pod kątem potencjalnych wad, z uwzględnieniem autokorelacji reszt modelu.

Wyżej wymienione procedury są wykonywane wielokrotnie, aż trzeci krok nie przyniesie żadnego ulepszenia modelu, jak stwierdzili Jerry Dr. i Hintze L. w 2007 roku.

Schemat Boxa-Jenkinsa służący do identyfikacji modelu przedstawiono na rysunku 1.3.



Rysunek 1.3: Schemat Boxa-Jenkinsa do identyfikacji modeli

Wstępnie ocenia się stacjonarność danych za pomocą reprezentacji graficznej i testu Dickeya-Fullera. W przypadku, gdy dane wykazują niestacjonarność, konieczne jest obliczenie procentowych zmian kursu walutowego, a następnie ponowne przeprowadzenie testu.

Wykorzystując funkcję autokorelacji (ACF) i funkcję częściowej autokorelacji (PACF), dążymy do zidentyfikowania minimalnych wartości p i q , które wykazują najmniejsze stopnie modeli autoregresji (AR) i średniej ruchomej (MA).

Kolejny etap polega na przybliżeniu wielkości p i q .

Następnie przystępujemy do kompleksowego badania pozostałych modeli i oceniamy poziom przezroczystości odnoszący się do białego szumu. W przypadku braku jakichkolwiek oznak białego szumu uzasadnione jest włączenie opóźnień p lub q lub obu.

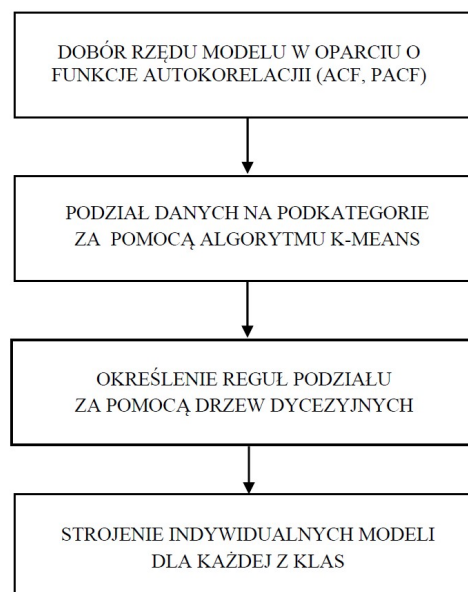
Wykorzystanie kryteriów informacyjnych, takich jak AIC i BIC, zaproponowane przez Hannana i Quinna w 1979 r., może ułatwić proces selekcji.

1.3.6 Proces wyboru modeli autoregresyjnych szeregów czasowych (TAR)

Proces konstruowania modeli statystycznych dla ciągu jednowymiarowego $\{x_n\}$, gdzie $n = 1, 2, N$, oraz $x \in \chi$, polega na wybraniu klasy modeli, takiej jak... Modele autoregresji, oznaczane jako $AR(k, A_k)$, są modelami liniowymi charakteryzującymi się wektorem parametrów $A_k = [a_0, a_1, \dots, a_k]$, gdzie k reprezentuje rząd modelu. Estymatory parametrów modelu \hat{A}_k wyprowadzane są z obliczeń statystycznych z wykorzystaniem zbioru danych $\{x_n\}$ i techniki najmniejszych kwadratów.

O precyzji modeli decyduje arbitralnie wybrana klasa modeli oraz zbiór danych wejściowych $\{x_n\}$. Ciąg danych białego szumu nie może być modelowany ze względu na brak korelacji między poszczególnymi wyrazami ciągu.

Diagram przedstawiony na rysunku 1.4 przedstawia proces wyboru modelu dla modelu progowej autoregresji (TAR).



Rysunek 1.4: Schemat doboru modeli TAR

Dokładność modeli progowych TAR, które obejmują obliczanie modeli AR(k) dla różnych obszarów przestrzeni χ , zależy od kilku czynników. Należą do nich jakość danych wejściowych, precyzja podziału przestrzeni χ na podzbiory do obliczeń poszczególnych modeli oraz wybór klas modeli.

Wiedza ekspercka może odgrywać kluczową rolę zarówno w modelach AR, jak i TAR, ponieważ wpływa na wybór kategorii AR(k) oraz podział przestrzenny w modelach TAR.

Proces dopasowywania modeli do danych eksperymentalnych, a następnie organizowania ich w klasy modeli pociąga za sobą ustalenie kryteriów identyfikacji reszt.

Algorytm grupowania k-średnich jest powszechnie stosowaną techniką w analizie danych i uczeniu maszynowym. Polega na podziale danego zbioru danych na K odrębnych klastrów na podstawie podobieństwa punktów danych. Algorytm iteracyjnie przypisuje każdy punkt danych do najbliższego środka ciężkości klastra i ponownie oblicza środek ciężkości każdego klastra, aż do osiągnięcia zbieżności. Grupowanie k-średnich zostało zastosowane w różnych dziedzinach, w tym w segmentacji obrazu, segmentacji klientów i wykrywaniu anomalii.

Algorytm grupowania k-średnich jest wykorzystywany do kategoryzowania obiektów w grupy. Algorytm dokonuje grupowania w oparciu o kryterium minimalizacji sumy kwadratów odległości między danymi a odpowiadającymi im centrami klastrów.

Algorytm k-średnich można wyrazić w następujący sposób:

Przyjmuje się, że liczba grup jest oznaczona przez k i wybiera się k początkowych centrów tych grup, które można wybrać losowo ze wszystkich punktów poszczególnych skupień.

Proces przypisywania obiektów (punktów) do określonych grup na podstawie wybranej miary podobieństwa, przy czym każdy punkt jest przypisywany do najbliższego skupienia.

Proces identyfikowania centrów nowych grup obejmuje obliczenie średniej lokalizacji punktów danych, które są powiązane z określoną grupą.

Weryfikacja warunku zakończenia algorytmu. W przypadku niespełnienia kryterium podobieństwa konieczne staje się powtórzenie drugiego etapu.

Kryterium konwergencji powszechnie stosowane w algorytmach grupowania polega na określeniu minimalnej różnicy, która powstaje po ponownym przypisaniu punktów danych do odpowiednich wyznaczonych centrów klastrów.

Algorytm k-średnich jest korzystny ze względu na swoją wydajność obliczeniową i prostotę, dzięki czemu nadaje się do przetwarzania obszernych zbiorów danych. Wynik algorytmu grupowania k-średnich jest taki, że każdy pojedynczy punkt danych jest przypisany wyłącznie do jednego klastra. Podejście k-średnich rozmytych klastrów, podobne do

logiki rozmytej, umożliwia przypisanie każdego punktu danych do wielu klastrów jednocześnie, z odpowiednim stopniem przynależności do każdego zbioru (Rutkowski, L., 2020; Talar, J., 2007) .

Drzewa decyzyjne są popularnym narzędziem w uczeniu maszynowym i eksploatacji danych, które służą do modelowania decyzji i ich możliwych konsekwencji. Są to graficzne reprezentacje procesów decyzyjnych obejmujących szereg wyborów i ich wyników. Drzewa decyzyjne są powszechnie stosowane w różnych dziedzinach, w tym w biznesie, finansach, opiece zdrowotnej i inżynierii, aby pomóc decydentom analizować złożone problemy i podejmować świadome decyzje na podstawie dostępnych danych.

Wykorzystanie drzew decyzyjnych jest istotną techniką klasyfikacji danych. Struktura struktury składa się z połączonych ze sobą gałęzi i węzłów, przy czym można wyróżnić węzeł główny i węzły końcowe. Drzewa decyzyjne to graficzne reprezentacje, które są skierowane i acykliczne. Wierzchołki grafu odpowiadają węzłom, podczas gdy gałęzie między węzłami reprezentują powiązania. W kontekście drzew decyzyjnych można zauważyć, że węzły są reprezentatywne dla testów przeprowadzanych na wartościach atrybutów warunkowych. Z drugiej strony krawędzie wskazują na wyniki tych testów, podczas gdy liście są reprezentatywne dla etykiet kategorii. W teorii grafów węzły pozbawione potomstwa są powszechnie określane jako liście, podczas gdy węzły pozbawione poprzednika są znane jako korzenie. Aby sklasyfikować atrybut w strukturze drzewa, jest on przesuwany w dół od węzła głównego w kierunku węzłów liścia, podążając za węzłami wewnętrznymi, które odpowiadają wynikom testu, aż do osiągnięcia końcowego węzła, znanego jako węzeł liścia. Obecnie etykieta klasy przypisana atrybutowi jest powiązana z węzłem liścia, jak podaje Probiez Barbara (2017).

Metodologia drzew decyzyjnych zaliczana jest do metod klasyfikacji i ułatwia reprezentację wiedzy w zrozumiały i prosty sposób. Wykorzystanie drzewa decyzyjnego ma na celu zidentyfikowanie zbioru spójnych zasad. Podstawowym standardem jest optymalizacja pozyskiwania informacji, czyli minimalizacja heterogeniczności w obrębie węzła. Do oceny wyszukiwania informacji w różnych algorytmach drzew decyzyjnych stosuje się różne kryteria.

Metodologia zastosowana do zautomatyzowanego generowania drzew decyzyjnych opiera się na technice „dziel i zwyciężaj”, która obejmuje hierarchiczny proces krok po kroku w celu określenia optymalnego atrybutu do przypisania jako etykiety dla wewnętrznych węzłów drzewa. Początkowy etap obejmuje wybór atrybutu, który służy jako reprezentacja korzenia drzewa. Po wybraniu atrybutu początkowego generowana jest odrębna gałąź dla każdej możliwej wartości atrybutu. Partycjonowanie zestawu danych jest oparte na wybranej wartości atrybutu. Następnie proces selekcji jest rekurencyjnie realizowany na każdym kanale przez wykorzystanie węzła, który odpowiada stosownemu podzbiorowi

atrybutów wykazujących porównywalne wartości. Według Probiez Barbary (2017) węzły liściowe powstają, gdy obiekty podzbioru mają tę samą etykietę klasy lub gdy spełnione jest z góry określone kryterium stopu, i są one wykorzystywane do reprezentowania przewidywanej etykiety klasy.

Algorytm CART (Classification and Regression Trees) jest szeroko stosowaną metodą konstruowania drzew decyzyjnych, która jest cennym narzędziem reprezentacji wiedzy. Technika CART to podejście do analizy danych, które dzieli dany zbiór danych na dwa podzbiory w celu zminimalizowania heterogeniczności. Każdy podział jest oparty na jednej zmiennej. Zdaniem Nowak- Brzezińskiej (2012) niektóre zmienne mają potencjał do wielokrotnego wykorzystania, inne mogą nie zostać wykorzystane w ogóle.

1.3.7 Facebook Prophet

Tematem zainteresowania jest Facebook Prophet.

Prognozowanie oparte na sekwencji czasowej jest szeroko przyjętą praktyką, przy czym dominującym podejściem są modele autoregresji. Uzyskanie wiarygodnych prognoz może okazać się trudnym zadaniem, ponieważ wymaga specjalistycznej wiedzy i dostępu do odpowiedniego oprogramowania.

Obecny dyskurs dotyczy artykułu ekspertów z dziedziny Facebooka, w którym przedstawiają pakiet do prognozowania szeregów czasowych znany jako Prophet. Oprogramowanie to umożliwia tworzenie precyzyjnych modeli predykcyjnych w sposób półautomatyczny. Taylor i Letham (2017) dostarczają wyczerpującego opisu pakietu Proroka i jego zastosowania. Podstawowym podejściem zastosowanym w tym pakiecie jest procedura dopasowywania uogólnionych modeli addytywnych (GAM):

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t, \quad (1.42)$$

Gdzie:

Funkcja $g(t)$ reprezentuje ogólny trend szeregu, podczas gdy $s(t)$ oddaje wahania sezonowe, takie jak te występujące w ujęciu rocznym lub tygodniowym. Dodatkowo $h(t)$ jest funkcją, która uwzględnia wpływ świąt i innych wydarzeń na szereg.

Błąd t -modelu ε obejmuje szereg informacji i zakłóceń, które nie są uwzględnione w modelu.

Oszacowane parametry modelu są uzyskiwane poprzez zastosowanie zasad statystyki bayesowskiej, w szczególności techniki wyznaczania maksimum a posteriori (MAP).

Wykorzystanie biblioteki Pythona, a mianowicie Facebook Prophet (<https://facebook.github.io/prophet/>), służy zamierzonemu celowi.

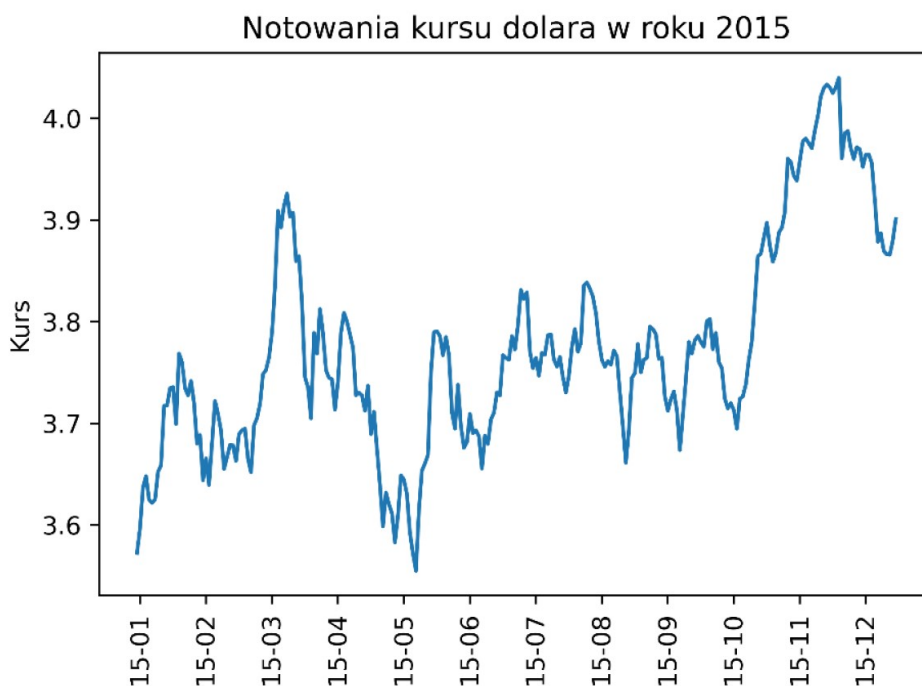
Model Proroka dla szeregów czasowych został specjalnie opracowany w celu przewidywania szeregów czasowych związanych z biznesem. Według Taylora i Lethama (2018)

model posiada parametry, które są intuicyjne, umożliwiając w ten sposób dostosowanie bez konieczności znajomości zawłości leżącego u podstaw modelu.

1.3.8 Identyfikacja modeli na przykładzie modelu ARIMA (podejście Boxa-Jenkinsa)

Niniejsze badanie koncentruje się na identyfikacji modeli, w szczególności na modelu ARIMA, z wykorzystaniem podejścia Boxa-Jenkinsa.

Ilustracją będzie analiza dziennych notowań walut USD/PLN. Zbiór danych obejmujący 250 jednostek dotyczy roku 2015, zaczerpnięty z <https://money.pl/czeniadze/nbparch/srednie/?symbol=USD>.



Rysunek 1.5: Przebieg notowań kursu dolara w 2015 roku

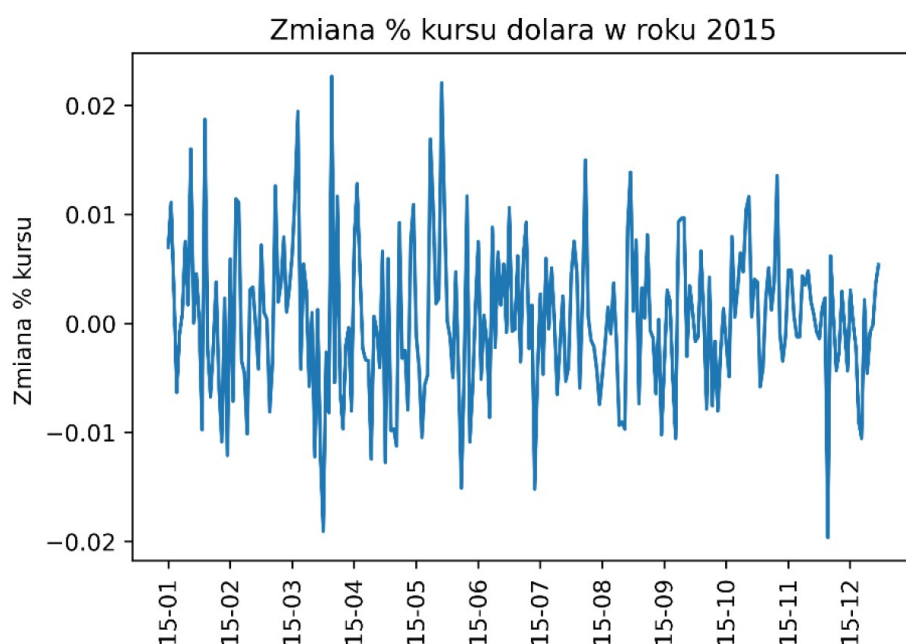
Do analizy stacjonarności sekwencji wykorzystuje się rozszerzony test Dickeya Fullera (ADF). Warto wspomnieć, że hipoteza zerowa testu ADF sugeruje obecność niestacjonarności podczas analizy. Kiedy obliczona wartość p jest niższa niż określony poziom α , hipoteza zerowa zostaje obalona i można stwierdzić, że sekwencja jest stacjonarna. Jeśli dane nie wykazują stacjonarności, uważa się je za niestacjonarne.

W tabeli 1.2 przedstawiono wyniki testu Augmented Dickey-Fuller (ADF) przeprowadzonego na ciągu danych dotyczącym zmiany kursu USD.

Tabela 1.2: Wyniki testu ADF dla ciągu danych notowań zmiany kursu USD

ADF Statistic:	-2.357800
p-value:	0.154003

W konsekwencji można wnioskować, że model wykazuje cechy białego szumu i wykazuje stacjonarność. Następnie, przy użyciu funkcji `pct_change()`, obliczana jest zmiana procentowa między bieżącym a poprzedzającym elementem kursu USD. Rysunek przedstawia uzyskany ciąg danych.



Rysunek 1.6: Zmiana procentowa notowań kursu USD w 2015 roku.

Jeśli chodzi o sekwencję kursów USD, test ADF jest obliczany dla danych o procentowej zmianie USD. Tabela 2.2 przedstawia wyniki testu procentowej zmiany kursu dolara amerykańskiego.

Wyniki testu ADF dla procentowej zmiany USD przedstawiono w tabeli 1.3.

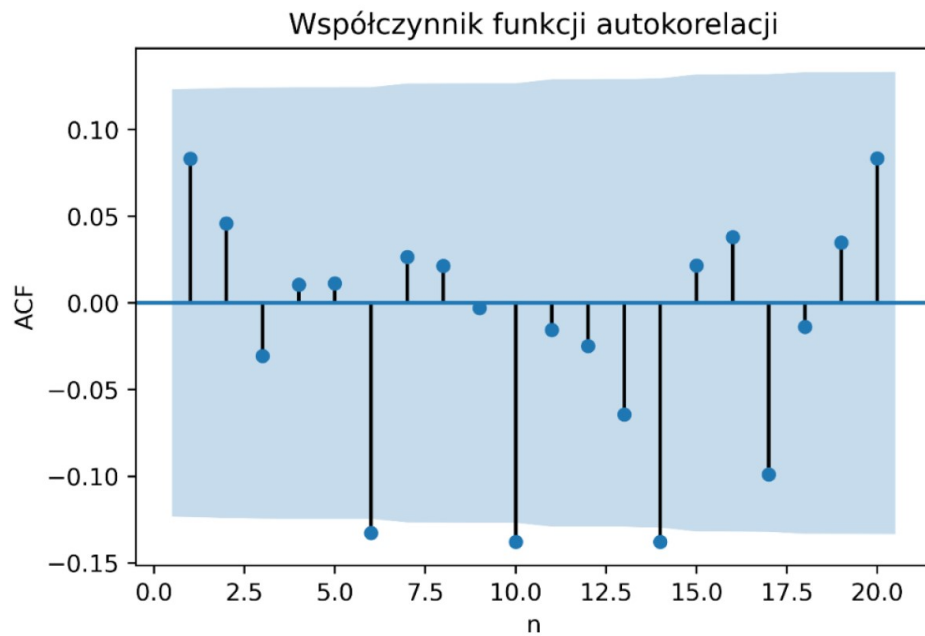
Tabela 1.3: Wyniki testu ADF dla ciągu % zmiany kursu USD

ADF Statistic:	-14.555140
p-value:	0.000000

Niestacjonarność naszego modelu wynika z wyników testu ADF. Aby kontynuować nasze badanie, wskazane jest obliczenie funkcji autokorelacji i częściowej autokore-

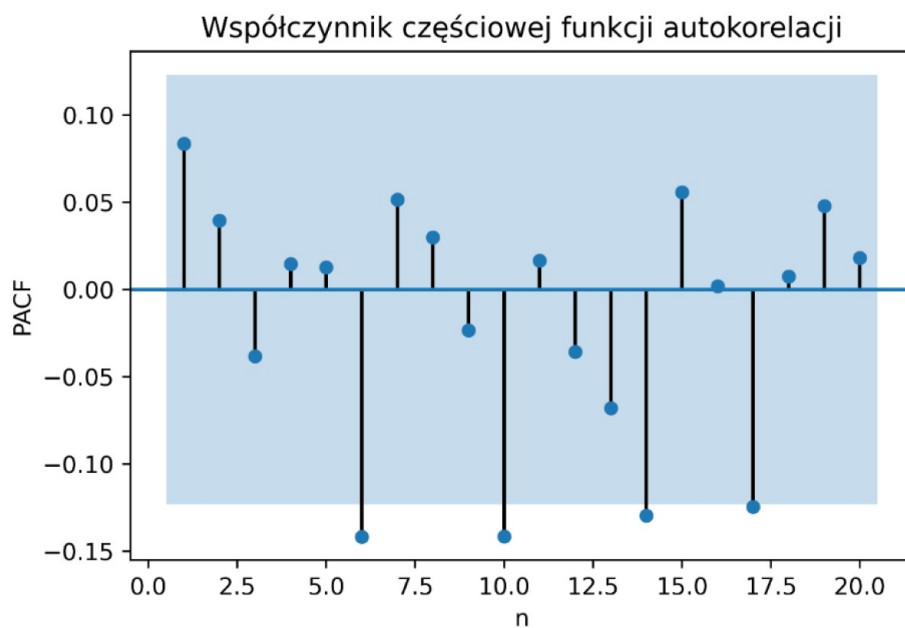
lacji.

Funkcja autokorelacji dla danych o zmianie % USD jest przedstawiona na rysunku 1.7.



Rysunek 1.7: Wykres funkcji autokorelacji dla danych % zmiany kursu USD

Funkcja autokorelacji (ACF) zapewnia korelacje między obserwacjami w chwili obecnej i we wszystkich poprzednich punktach czasowych, jak pokazano na rysunku 1.7.



Rysunek 1.8: Wykres częściowej funkcji autokorelacji dla danych % zmiany kursu USD

Przedstawiona reprezentacja wizualna, oznaczona na rysunku 1.8, przedstawia wykres funkcji cząstkowej autokorelacji dla danych dotyczących procentowej zmiany dolara amerykańskiego.

Funkcja częściowej autokorelacji (PACF) to narzędzie statystyczne, które mierzy korelację między dwiema obserwacjami czasowymi, biorąc pod uwagę wpływ innych punktów danych. Obliczenie tej funkcji obejmuje obliczenie korelacji cząstkowej przy jednoczesnym wykonaniu poprawki statystycznej na wpływ opóźnień pośrednich. Rysunek 2.6 przedstawia częściową funkcję autokorelacji j odnoszącą się do procentowej zmiany kursu dolara amerykańskiego.

Optymalne parametry modelu do prognozowania można określić za pomocą funkcji `auto.arima`. Kryteria informacyjne AIC, BIC i HQIC zostały obliczone i użyte do wybrania optymalnego modelu, którym okazał się ARIMA(3,1,1). Parametry modelu przedstawiono w tabeli 1.4.

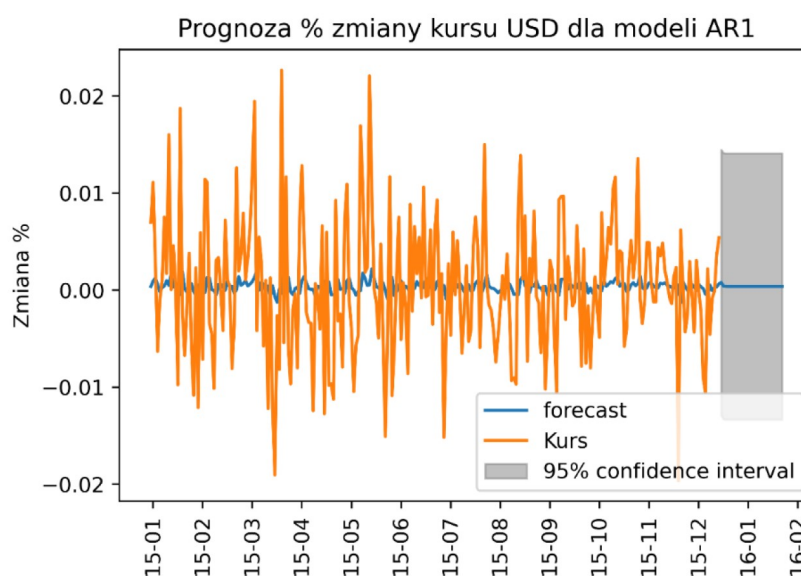
Tabela 1.4: Parametry modelu ARIMA(3,1,1)

Model:	ARIMA(3,1,1)
Log Likelihood	883.735
AIC	-1755.470
BIC	-1734.293
HQIC	-1746.949
intercept	$-8.522e - 06 \pm 4.33e - 05$
AR.L1	0.1356 ± 0.072
AR.L2	-0.0995 ± 0.068
AR.L3	-0.0622 ± 0.069
MA.L1	-0.9147 ± 0.042
sigma2	0.0025 ± 0.000

Analiza funkcji autokorelacji sugeruje, że obserwowane zjawisko można potencjalnie scharakteryzować jako proces stochastyczny o zależności dalekiego zasięgu. W szerszym znaczeniu proces ten uważa się za stacjonarny, o czym świadczy zmniejszanie się funkcji autokorelacji do zera (patrz rysunek 1.7). Współczynniki ACF i PACF sugerują

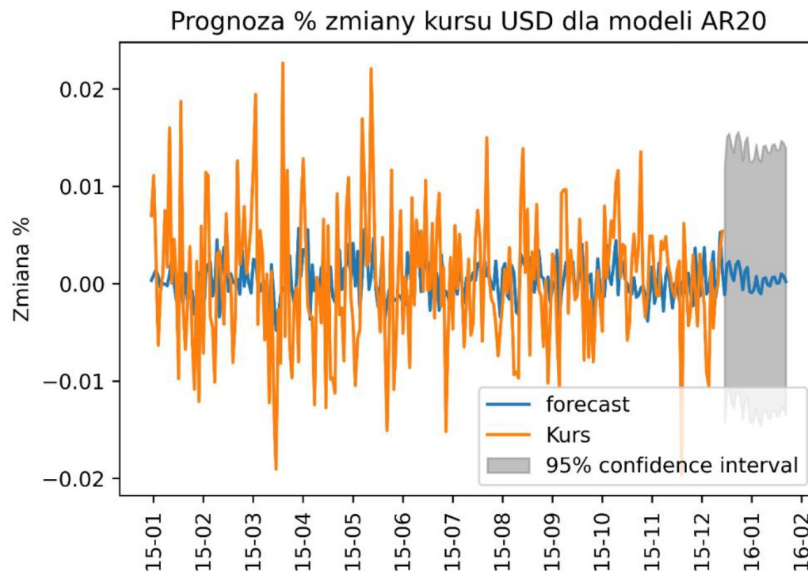
liniowe powiązanie statystyczne, potwierdzając tym samym przydatność włączenia liniowego modelu AR(1). Dla celów porównawczych rozważymy model AR(20).

Reprezentacja graficzna przedstawiona na rysunku 1.9 ilustruje procentową zmianę kursu dolara amerykańskiego (USD) w porównaniu z prognozą autoregresyjną (AR)(1).



Rysunek 1.9: Wykres % zmian kursu USD i prognozy wyznaczonej za pomocą modeli AR(1)

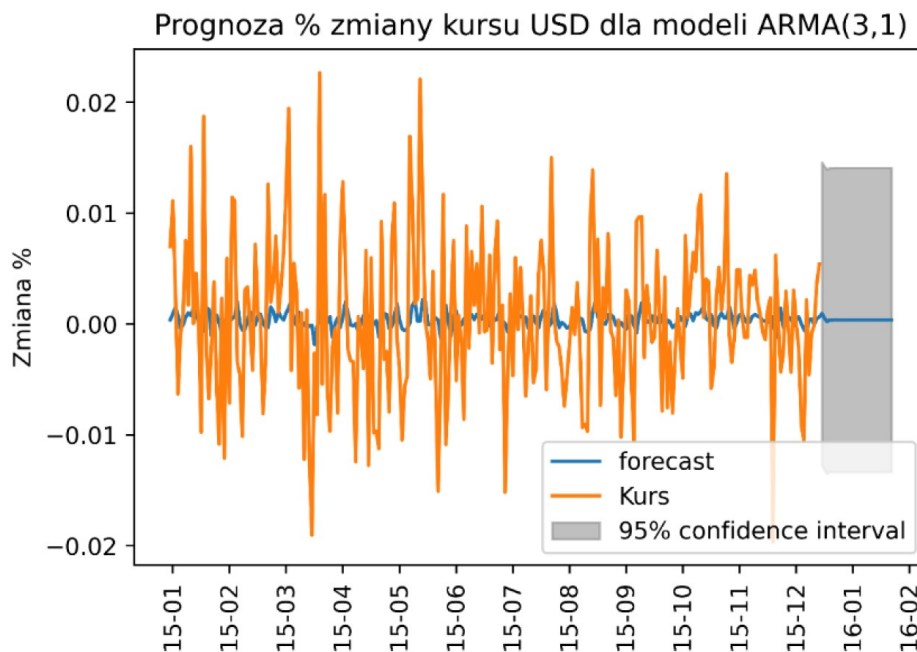
Przedstawiona na wykresach 1.9 i 1.10 prognoza podlega znacznemu uśrednieniu. Ponadto przewidywane informacje będą mieścić się w przedziałach od ujemnego punktu zerowego pięć do dodatniego punktu zerowego pięć. 014;0 . Wartość 0,014 dotyczy modelu autoregresyjnego pierwszego rzędu (AR(1)), natomiast wartość -0,3 odnosi się do ujemnego współczynnika korelacji. 0125; 0. Model AR(20) został oszacowany na 0,125 z 95% poziomem niezawodności.



Rysunek 1.10: Wykres % zmian kursu USD i prognozy wyznaczonej za pomocą modeli AR(20)

Wykres przedstawiony na rysunku 1.10 przedstawia procentowe zmiany kursu dolara amerykańskiego (USD) wraz z prognozą obliczoną za pomocą modeli autoregresyjnych (AR) na próbie o wielkości 20 osób.

Dla ułatwienia porównania przedstawimy modele ARMA(3,1) i ARMA(3,2) zastosowane do danych o procentowych zmianach USD. Ryciny 1.11 przedstawiają przebieg danych wraz z prognozami.



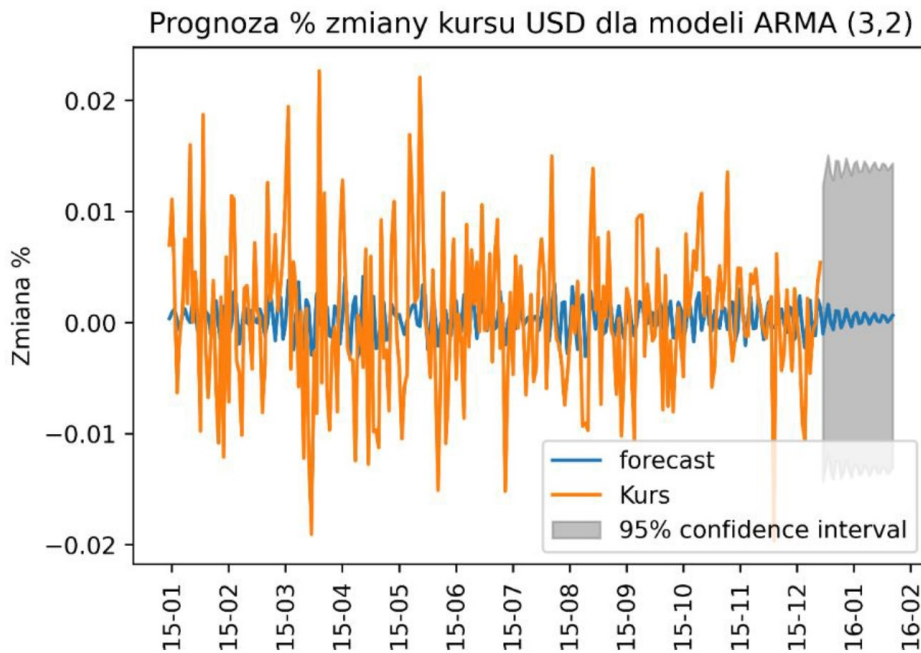
Rysunek 1.11: Wykres % zmian kursu USD i prognozy dla modeli ARMA(3,1)

Ponadto obliczono parametry, które przedstawiono w tabeli 1.5.

Tabela 1.5: Porównanie parametry modeli ARMA(3,1) i ARMA(3,2)

Model:	ARMA(3,1)	ARMA(3,2)
Log Likelihood	897.709	902.696
AIC	-1783.417	-1791.393
BIC	-1762.217	-1766.659
HQIC	-1774.888	-1781.442
const	0.0002 ± 0.000	0.0004 ± 0.000
AR.L1	-0.361 ± 0.781	-0.3346 ± 0.088
AR.L2	0.0524 ± 0.088	0.8833 ± 0.046
AR.L3	-0.0335 ± 0.072	0.0087 ± 0.073
MA.L1	0.1180 ± 0.780	0.4275 ± 0.065
MA.L2	–	0.9662 ± 0.052

Graficzna reprezentacja przedstawiona na rysunku 1.12 przedstawia procentowe zmiany kursu dolara amerykańskiego (USD) wraz z przewidywanymi wartościami wywiezionymi z modeli autoregresyjnej średniej ruchomej (ARMA) z parametrami (3,2).



Rysunek 1.12: Wykres % zmian kursu USD i prognozy wyznaczonej za pomocą modeli ARMA(3,2)

1.3.9 Model ARX i identyfikacja dla ciągu obserwacji procesu

Niniejsza praca dotyczy wykorzystania modelu Autoregresji z Wejściami Egzogenicznymi (ARX) do identyfikacji i analizy sekwencji obserwacji danego procesu technologicznego.

Metodologia ARX reprezentuje liniowy model jednostki dynamicznej z dyskretnym czasem, w którym zmienna wejściowa jest oznaczona jako x , a zmienna wyjściowa jako y .

Proponowany model można przedstawić w następującej formie:

$$y_n = ay_{n-1} + bx_{n-2} + \varepsilon_n, \tag{1.43}$$

Przedstawione równanie jest modelem autoregresyjnym drugiego rzędu, gdzie y_n jest funkcją dwóch poprzednich obserwacji, y_{n-1} i y_{n-2} , a także składnika błędów, ε_n . Współczynniki a i b reprezentują wagi nadane wcześniejszym obserwacjom w modelu.

Zmienna y jest reprezentowana jako proces autoregresyjny rzędu pierwszego (AR(1)).

Zmienna wejściowa podlega dwustopniowemu opóźnieniu.

Symbol ε oznacza proces stochastyczny, który jest całkowicie losowy.

Wartości parametrów modelu a i b można ustalić metodą najmniejszych kwadratów, wykorzystując zbiór danych $\{(x_n, y_n)\}$, gdzie $n = 1, 2, \dots, N$, wywodzący się z sekwencji obserwacji procesu technologicznego. Kryterium identyfikacji zaproponowane przez

Niederlińskiego (1979) oraz Mańczaka i Nahorskiego (1983) wyraża się następująco:

$$Q = \sum_{n=3}^N [y_n - (ay_{n-1} + bx_{n-2})]^2 \rightarrow \min. \quad (1.44)$$

Celem podanego równania jest zminimalizowanie sumy kwadratów różnic między obserwowanymi wartościami zmiennej y a jej wartościami przewidywanymi na podstawie liniowej kombinacji dwóch poprzedzających ją wartości y , reprezentowanych przez współczynniki a i b . Sumowanie obejmuje wszystkie wartości n większe lub równe 3.

Celem jest uzyskanie estymatorów \hat{a} i \hat{b} przez spełnienie warunku istnienia ekstremum funkcji (1.43).

$$\frac{\partial Q}{\partial a} = 0, \quad \frac{\partial Q}{\partial a} = \sum_{n=3}^N \{2[y_n - (ay_{n-1} + bx_{n-2})](-y_{n-1})\}$$

$$\frac{\partial Q}{\partial b} = 0, \quad \frac{\partial Q}{\partial b} = \sum_{n=3}^N \{2[y_n - (ay_{n-1} + bx_{n-2})](-x_{n-1})\}$$

$$\sum_{n=3}^N [-y_n y_{n-1} + ay_{n-1}^2 + bx_{n-2} y_{n-1}] = 0$$

$$\sum_{n=3}^N [-y_n x_{n-2} + ay_{n-1} x_{n-2} + bx_{n-2}^2] = 0$$

$$\begin{cases} a \sum_n y_{n-1}^2 + b \sum_n x_{n-2} y_{n-1} = \sum_n y_n y_{n-1} \\ a \sum_n y_{n-1} x_{n-2} + b \sum_n x_{n-2}^2 = \sum_n y_n y_{n-2} \end{cases}$$

Układ równań rozwiązuje się za pomocą arkusza kalkulacyjnego. Podane wyrażenia zostaną zastąpione literami A, B, C, D i E.

$$A = \sum_n y_{n-1}^2$$

$$B = \sum_n y_{n-1} x_{n-2}$$

$$C = \sum_n y_n y_{n-1}$$

$$D = \sum_n y_n x_{n-2}$$

$$E = \sum_n x_{n-2}^2$$

Otrzymuje się układ równań algebraicznych w odniesieniu do zmiennych a i b .

$$\begin{cases} \hat{a}A + \hat{b}B = C \\ \hat{a}B + \hat{b}E = D \end{cases}$$

Układ równań można przedstawić jako równanie macierzowe, w którym główny wyznacznik określają współczynniki zmiennych.

$$W = \begin{vmatrix} C & A \\ A & N \end{vmatrix} = AE - B^2, \quad W \neq 0,$$

$$W_{\hat{a}} = \begin{vmatrix} C & B \\ D & E \end{vmatrix} = CE - DB,$$

$$W_{\hat{b}} = \begin{vmatrix} A & C \\ B & D \end{vmatrix} = AD - CB,$$

Przedstawione równanie jest wyrażeniem matematycznym opisującym zależność między wektorami a i b oraz ich odpowiednimi przekształceniami za pomocą operatora liniowego W .

$$\hat{a} = \frac{W_{\hat{a}}}{W}, \quad \hat{b} = \frac{W_{\hat{b}}}{W}.$$

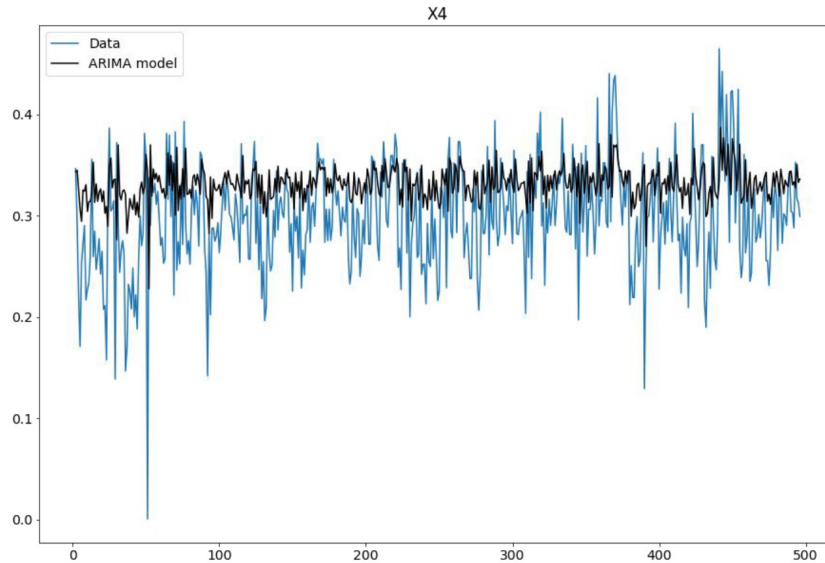
Model ARX jest rodzajem modelu autoregresyjnego, który uwzględnia zewnętrzne dane wejściowe. Modele ARX są wykorzystywane do charakteryzowania systemów, gdzie x reprezentuje zmienną wejściową systemu, a y reprezentuje zmienną wyjściową systemu. Określenie kolejności modelu ARX jest uzależnione od uwzględnienia liczby branych pod uwagę kroków wejściowych i wyjściowych oraz opóźnienia.

Model ARIMAX (1,0,1) jest oceniany za pomocą kryteriów informacyjnych przedstawionych w tabeli 1.6.

Tabela 1.6: Kryteria informacyjne dla modelu ARIMAX (1,0,1)

Model:	ARIMAX (1,0,1)
Method	MLE
Log Likelihood	675.0903
AIC	-1338.180618076955
BIC	-1312.9532715015428

Rysunek 1.13 przedstawia graficzną reprezentację prognozy.



Rysunek 1.13: Prognoza X4 dla modelu ARIMAX (1,0,1).

1.3.10 Prognoza przy stosowaniu modelu TAR

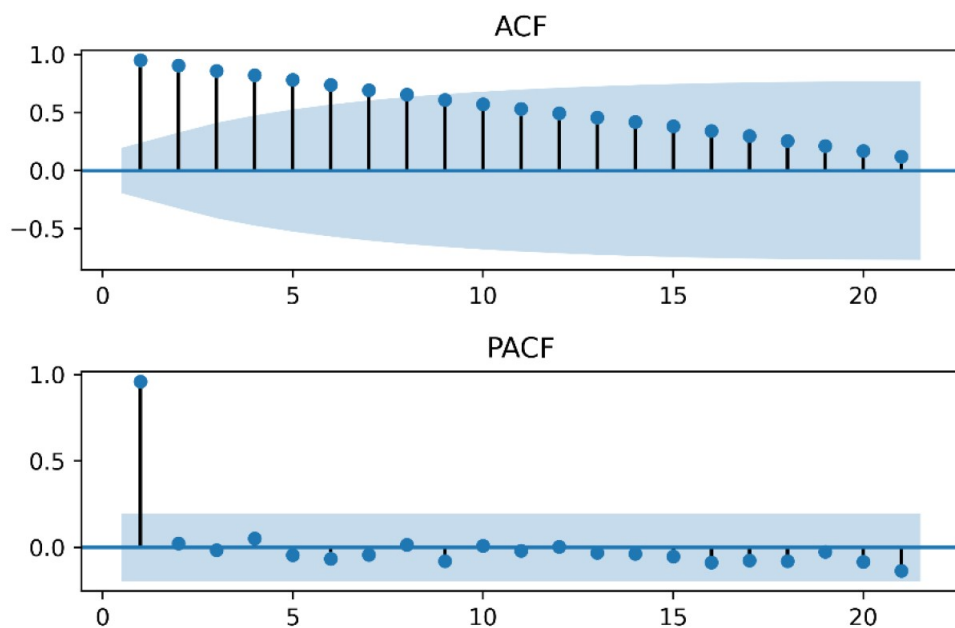
Do wygenerowania prognozy wykorzystano model autoregresji zmiennej w czasie (TAR).

Instancja modelu wygenerowała 100 obserwacji.

$$x[k + 1] = \begin{cases} 0.3x[k] + \varepsilon[k] & \text{dla } x < 10 \\ 0.99x[k] + \varepsilon[k] & \text{dla } x > 10 \end{cases} \quad (1.45)$$

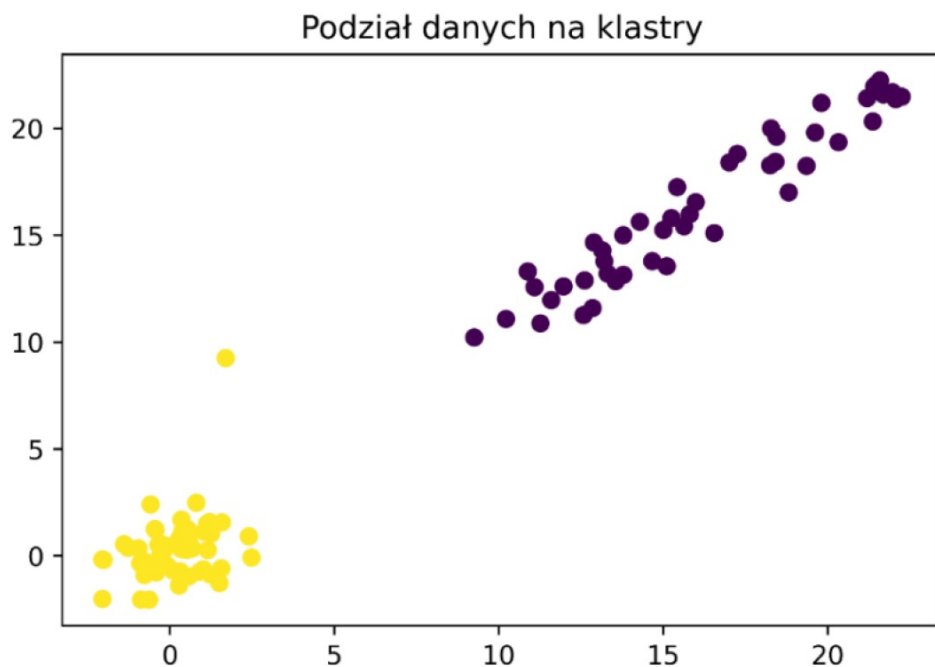
W sposób porównywalny do kwotowań USD/PLN, prognozy zostały obliczone z wykorzystaniem struktury ANFIS dla kwotowań EUR/PLN, jak pokazano na wykresie 1.14. Model rozmyty TSK został poddany modyfikacji strukturalnej, zgodnie z którą przyjęto, że przestrzeń wejściowa zawiera 3, 4, 5, 7 i 9 zbiorów. Przejaw zmienności błędów podczas procesu uczenia jest przedstawiony za pomocą graficznych reprezentacji przedstawionych na rysunkach 1.15, 1.16 i ???. Redukcja błędu predykcji podczas procesu uczenia odbywa się zgodnie z rozkładem wykładniczym. Wraz ze wzrostem liczby zbiorów rozmytych błęd stanu ustalonego zmniejsza się.

Prognozowane wartości sekwencji notowań EUR/PLN (ANFIS) zestawiono na wykresie 1.14 z danymi wejściowymi.



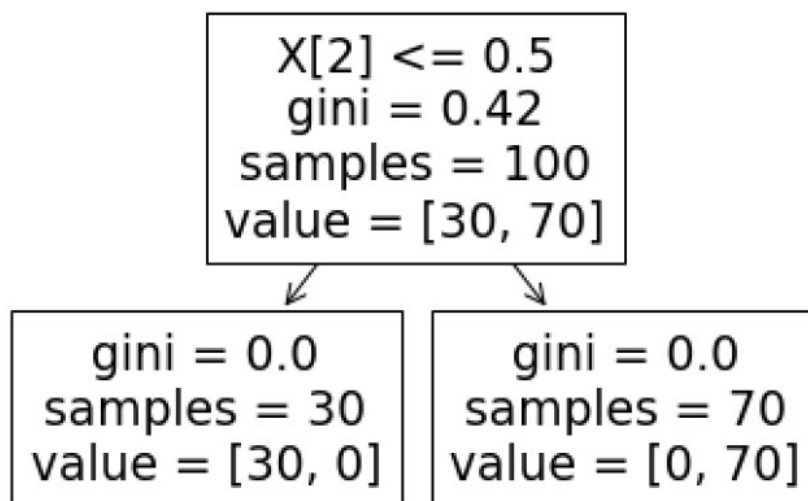
Rysunek 1.14: Wykresy funkcji autokorelacji oraz autokorelacji częściowej

W niniejszym opracowaniu przedstawiono rysunek 1.15, który obrazuje zmienność błędów w procesie uczenia prognozowania notowań EUR/PLN w systemie ANFIS z wykorzystaniem trzech zbiorów rozmytych.



Rysunek 1.15: Wykres podziału danych wejściowych na klastry

W niniejszym opracowaniu przedstawiono rysunek 1.16, który ilustruje zmienność błędów podczas procesu uczenia prognozowania notowań EUR/PLN w systemie ANFIS z wykorzystaniem czterech zbiorów rozmytych.



Rysunek 1.16: Drzewo decyzyjne

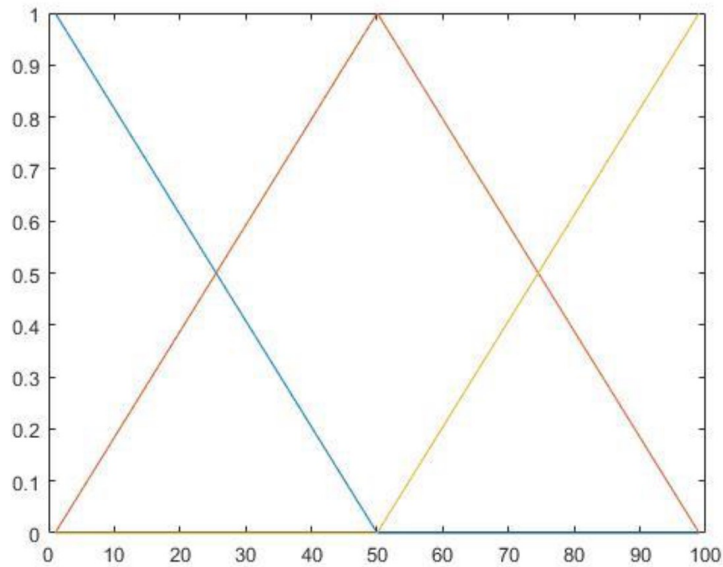
Podczas nabywania umiejętności prognozowania notowań DJIA można dostrzec charakterystyczne rodzaje błędów, jak pokazano na rysunkach ?? i ?. Wyzwania napotkane podczas dostosowywania modelu ANFIS do sekwencji danych szkoleniowych można przypisać znacznemu zakresowi zmienności wykazywanemu przez dane. Podstawowy ciąg cytatów podlega wahaniom od 1,5 do 3×10^4 w swojej wartości liczbowej.

Prognozowane wartości (ANFIS) kursu DJIA przedstawiono na rysunku ?? w odniesieniu do danych wejściowych.

Niniejsze badanie przedstawia rysunek ??, który przedstawia zmienność błędów podczas procesu uczenia modeli prognoz dla kwotowań DJIA w systemie ANFIS. Model wykorzystuje trzy zbiory rozmyte.

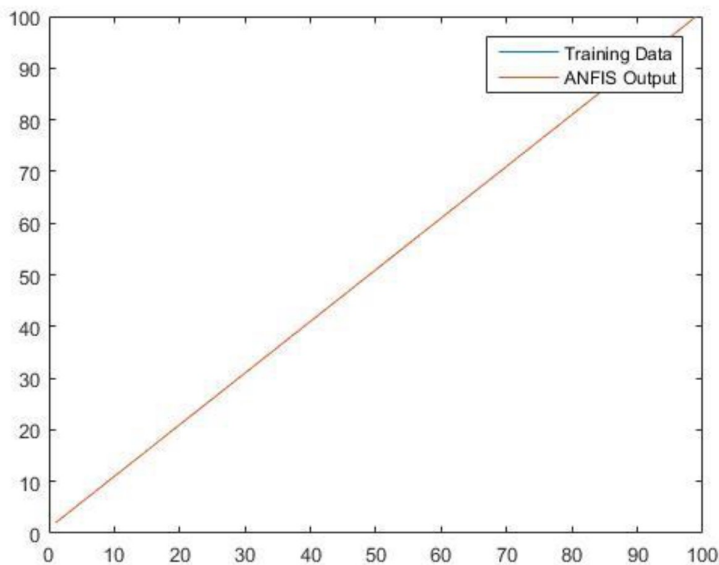
Na wykresach ?? i ?? przedstawiono wyniki testów identyfikacyjnych przeprowadzonych na sekwencji notowań polskiego indeksu giełdowego PMP. Zaobserwowano płynną malejącą funkcję w charakterze błędu dla wszystkich trzech badanych struktur modelu, które obejmowały 3, 4, 5 i 7 zbiorów w przestrzeni zmiennych wejściowych. Względna wartość błędu nie przekracza 8%. Górna granica stopnia przynależności dla struktury składającej się z siedmiu zbiorów rozmytych nie przekracza 8%.

Wartości prognozy ANFIS sekwencji notowań PMP zestawione zostały z danymi wejściowymi na wykresie ??.



Rysunek 1.17: Podział przestrzeni na zbiory rozmyte o funkcjach przynależności typu trójkątnego

Zmienność błędów w procesie uczenia systemu ANFIS (z 7 zbiorami rozmytymi) do prognozowania notowań PMP przedstawia wykres ??.



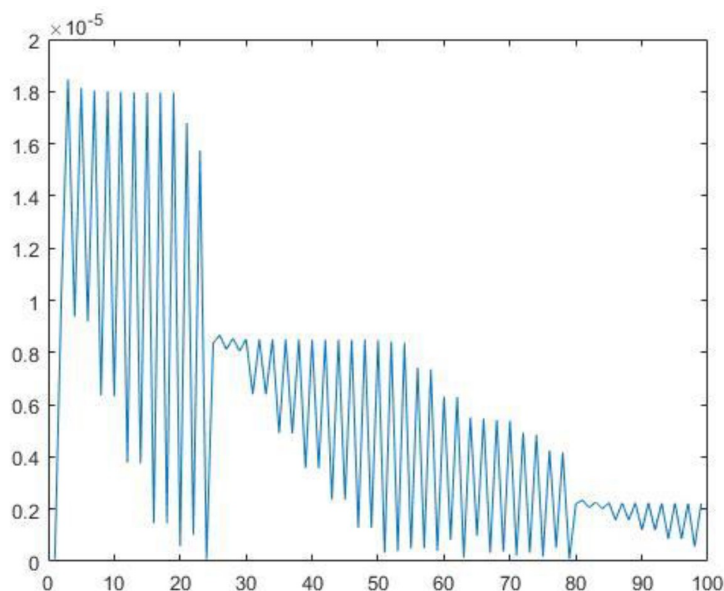
Rysunek 1.18: Wartości prognozy (ANFIS) ciągu danych testowych na tle danych wejściowych przy funkcjach przynależności typu trójkątnego

Stochastyczny charakter omawianych procesów wskazuje na proces białego szumu. Prognoza systemu ANFIS wykazuje niewielkie wahania i jest zbliżona do wartości średniej, co ilustrują wykresy ?? i ?. Błędy prognoz podczas uczenia dla poszczególnych procesów przedstawiono na rysunkach ??, ??, ?? i ?. Zaobserwowano, że błędy wykazują wykładniczy spadek w zakresie od 100 do 150 kroków. Wraz ze wzrostem liczby plików w

strukturze modelu ANFIS obserwuje się zmniejszanie się wartości błędu stałego. Niemniej jednak wielkości wartości błędów nadal są godne uwagi i wahają się od 15% do 20%.

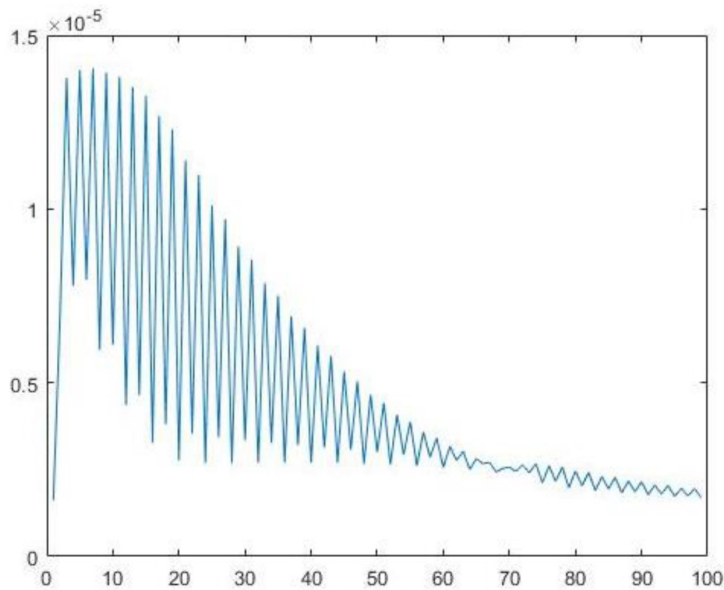
Diagram przedstawiony na rysunku ?? przedstawia prognozowane wartości (ANFIS) notowań cen masła w odniesieniu do danych wejściowych.

Niniejsze badanie przedstawia rysunek ??, który przedstawia zmienność błędów podczas procesu uczenia prognozowania kwotowań cen masła w systemie ANFIS z wykorzystaniem trzech zbiorów rozmytych.



Rysunek 1.19: Zmienność błędu w procesie uczenia prognozy dla danych testowych w systemie ANFIS (3 zbiory rozmyte).

Niniejsze badanie przedstawia rysunek 4.30, który przedstawia stopień zmienności błędów podczas procesu uczenia się prognoz dla kwotowań cen masła w systemie ANFIS, w którym zastosowano 7 zbiorów rozmytych.



Rysunek 1.20: Zmienność błędu w procesie uczenia prognozy dla danych testowych w systemie ANFIS (7 zbiorów rozmytych).

Niniejsza ilustracja przedstawia rysunek 4.31, który przedstawia dane wejściowe i ANFIS odnoszące się do notowań OMP.

W niniejszym opracowaniu przedstawiono rysunek 4.32, na którym przedstawiono zmienność błędów zaobserwowanych podczas procesu uczenia prognozy dla kwotowań cen masła w systemie ANFIS z wykorzystaniem trzech zbiorów rozmytych.

Niniejsze badanie przedstawia rysunek 4.33, który przedstawia zmienność błędów podczas procesu uczenia prognozy dla kwotowań cen masła w systemie ANFIS, z wykorzystaniem 5 zbiorów rozmytych.

Tabela 4.1 przedstawia obszerne zestawienie błędów związanych z ilością zbiorów rozmytych.

Tabela 4.1 przedstawia korelację między błędami a liczbą wykorzystanych zbiorów rozmytych.

Pełny wykaz błędów od ilości zbiorów rozmytych przedstawiono w tabeli 1.7.

Tabela 1.7: Zależność błędów od ilości zbiorów rozmytych

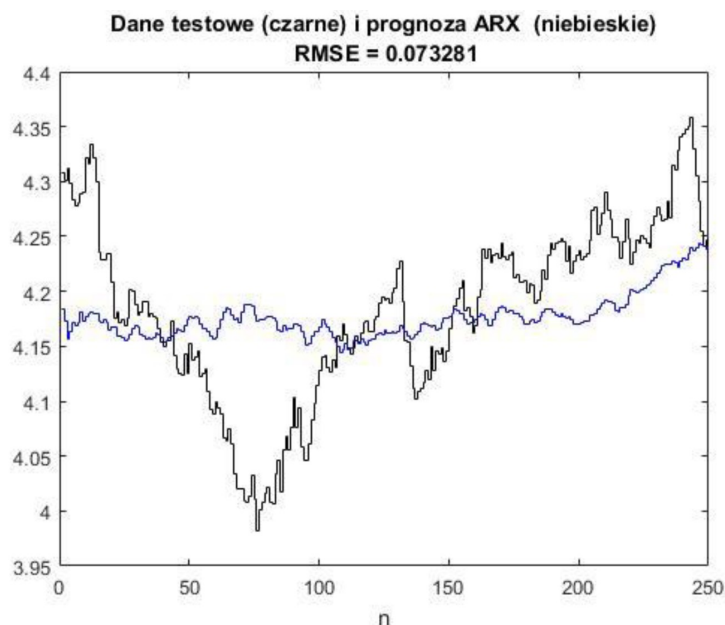
Dane	Liczby zbiorów rozmytych			
	3	4	5	7
USD2015	0,0252	0,0255	0,0251	0,0248
EUR2015	0,0147	0,0146	0,0144	0,0142
PMP	19,26	18,07	17,62	16,47
DJIA	186,276	186,168	185,98	185,652
OMP	0,1113	0,1111	0,1108	0,1098
Masło	0,0496	0,0495	0,0494	0,0491

Liczba zbiorów rozmytych, które dzielą przestrzeń zmiennych wejściowych, a co za tym idzie liczba reguł, jest krytycznym czynnikiem przy określaniu błędu predykcji.

Do identyfikacji obiektów dynamicznych wykorzystano model ANFIS z podwójnymi wejściami.

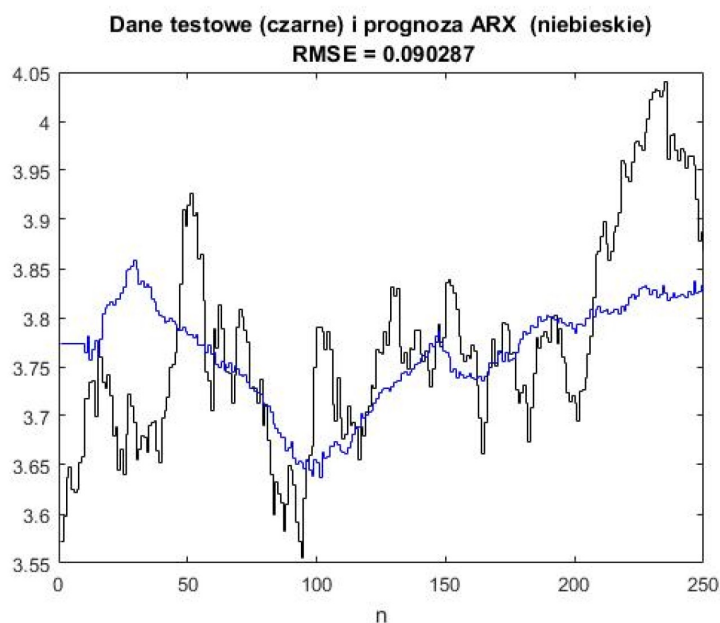
Model ARX jest jednym z celów modeli dynamicznych. W chwili n wyjście obiektu jest określone przez liniową kombinację jego przeszłych wartości i wejścia, które zostało opóźnione. Rozdział 4 pracy dotyczył wyboru dowolnych sekwencji obserwacji na potrzeby modelowania takich podmiotów. W konsekwencji przewidywanie jest obliczane za pomocą liniowego połączenia poprzedzających sygnałów wejściowych i wyjściowych. Przebiegi prognoz systemu ANFIS dla różnych sekwencji obserwacji przedstawiono na rysunkach od 1.21 do 1.25.

Model ARX został wykorzystany do predykcji notowań USD/PLN w stosunku do notowań EUR, jak pokazano na wykresie 1.21.



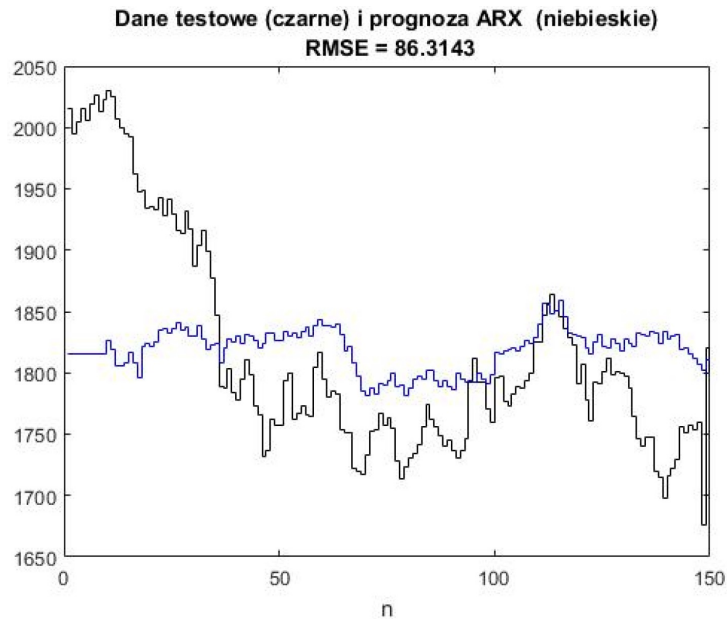
Rysunek 1.21: Przebieg notowań USD/PLN prognozowanych według modelu ARX w zależności od notowań euro.

Model ARX został wykorzystany do predykcji notowań EUR/PLN w stosunku do notowań USD, co obrazuje wykres 1.22.



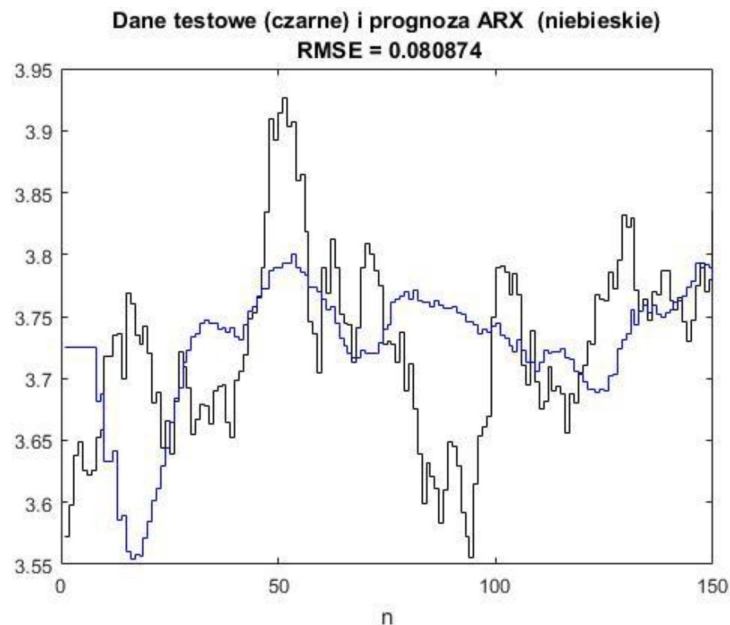
Rysunek 1.22: Przebieg notowań EUR/PLN prognozowanych według modelu ARX w zależności od notowań dolara.

Do predykcji notowań USD/PLN w stosunku do notowań PMP wykorzystano model ARX, co obrazuje wykres 1.23.



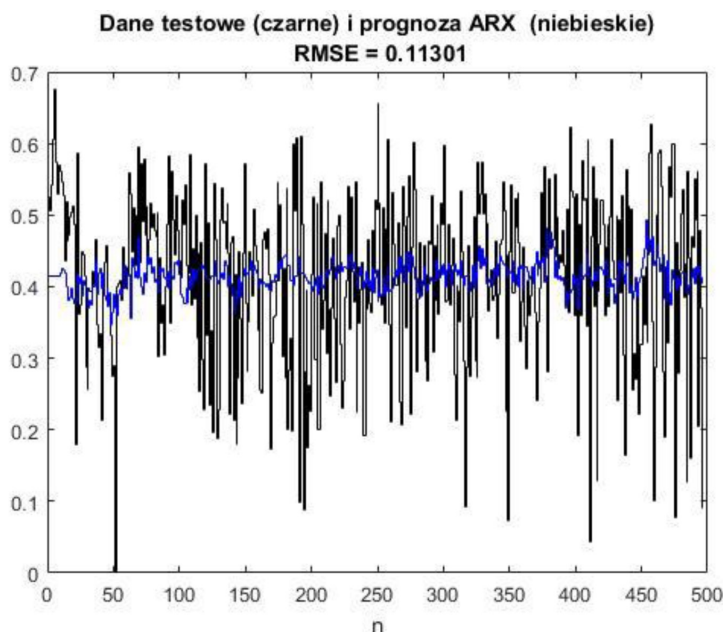
Rysunek 1.23: Przebieg notowań USD/PLN prognozowanych według modelu ARX w zależności od notowań PMP.

Zgodnie z modelem ARX prognoza notowań PMP jest uzależniona od notowań dolara, co obrazuje wykres 1.24.



Rysunek 1.24: Przebieg notowań PMP prognozowanych według modelu ARX w zależności od notowań dolara.

Model ARX został wykorzystany do śledzenia trajektorii prognoz zmiennej OMP , jak pokazano na rysunku 1.25, zależnych od zmiennej X4.



Rysunek 1.25: Przebieg prognoz zmiennej OMP według modelu ARX w zależności od zmiennej X4.

Model autoregresyjnej średniej ruchomej z danymi wejściowymi egzogenicznymi (ARX) można przedstawić jako formę agregacji, która przypisuje wagi różnym etapom czasowym w ramach dwóch odrębnych szeregów czasowych. Proces uśredniania jest łatwo widoczny na wykresach.

1.4 Sieci neuronowe jako nowoczesne narzędzie służące do prognozowania

Początek rozwoju sztucznych sieci neuronowych nastąpił w 1943 roku wraz z opracowaniem matematycznego modelu sztucznego neuronu przez McCullocha i Pittsa [22]. Jednak początek fascynacji i badaniem ludzkiego mózgu, i ogólne przetwarzanie wiedzy przez organizmy żywe trudno określić. Początki prac nad sieciami neuronowych należy łączyć z 1904 rokiem i pracą Pavlova na temat teorii odruchów warunkowych. Pomimo upływu lat sztuczne sieci neuronowe cieszą się nadal dużym zainteresowaniem. W latach 90 ubiegłego wieku była to tematyka prawie powszechna w środowiskach neurobiologów, fizyków, matematyków i informatyków. Wszędzie tam, gdzie nie znano algorytmu do rozwiązania problemu lub zadanie było zbyt złożone próbowano korzystać z sieci neuronowych do rozwiązania. Sztuczne sieci neuronowe budzą zainteresowanie z powodu ich odporności na uszkodzenia (podobnie jak naturalne sieci neuronowe) oraz (i przede wszystkim) na ich zdolność uczenia się i rozwiązywania złożonych problemów. Oprócz klasycznych metod przewidywania zjawisk ekonomicznych istnieje również zestaw metod nieklasycznych. Grupa tych metod koncepcyjnie łączy metody wykorzystujące no-

we techniki modelowania przepływu pracy i wpływu. Neuron, czyli komórka nerwowa, jest podstawowym elementem biologicznej sieci neuronowej, natomiast sztuczny neuron jest elementem, którego własności odpowiadają wybranym własnościom neuronu biologicznego. Z kolei sztuczna sieć neuronowa to model prognostyczny, opisany odpowiednim algorytmem w postaci programu komputerowego. Program ten naśladuje zdolność ludzkiego umysłu do klasyfikacji wzorów i sporządzania prognoz lub podejmowania decyzji na podstawie analizy retrospektywnej. Istotną cechą sieci neuronowych jest fakt, że w wyniku procesu uczenia sieć może nabyć zdolność predykcji (przewidywania). Sieć dzięki swoim zdolnościom do uczenia, adaptacji i uogólnienia doświadczeń, pozwala zautomatyzować procesy wnioskowania retrospektywnego, tzn. wykorzystującego wcześniej zgromadzone dane. Neurony, czyli komórki nerwowe, są podstawowymi elementami biologicznych sieci neuronowych, a sztuczne neurony to elementy, których charakterystyka odpowiada wybranym cechom neuronów biologicznych. Sztuczne sieci neuronowe od lat potwierdzają swoją skuteczność w rozpoznawaniu i klasyfikacji wzorców (przydzielaniu wzorcom kategorii), predykcji szeregów czasowych, zaawansowanej analizie danych statystycznych, odszumianiu i kompresji sygnałów jedno i wielowymiarowych oraz w zagadnieniach sterowania i automatyzacji [261].

Obecnie najpowszechniejszym zastosowaniem technologii sieci neuronowych jest problem rozpoznawania, zwłaszcza problem rozpoznawania kontekstu. Sieci neuronowe są również wykorzystywane do zadań klasyfikacyjnych oraz analizy i przetwarzania obrazu. Na przykład, w porównaniu z klasycznymi metodami szeregów czasowych, ich zaletą jest to, że dzięki nieliniowemu opisowi matematycznemu można odwzorować procesy nieliniowe i łatwo modelować procesy wielowymiarowe. Sieci neuronowe są efektem badań z zakresu sztucznej inteligencji, a ich struktura i funkcja, podobnie jak zbiory rozmyte, a ich koncepcja i topologią są oparte na zasadzie działania ludzkiego mózgu. Badania nad budową i działaniem sieci neuronowych prowadzone są od około 50 lat, a ich zakres zastosowań obejmuje wszystkie dziedziny nauki, od nauk humanistycznych po nauki społeczne i biologię, a następnie ściśle nauki techniczne. Sztuczna sieć neuronowa odwzorowuje rzeczywistą biologiczną sieć neuronową, aby stworzyć ludzki mózg. Odwzorowanie to jest przybliżone co ma związek z oczywistymi barierami budowy sztucznych sieci neuronowych. Mózg ludzki składa się z około 100 miliardów komórek nerwowych (neuronów) połączonych ze sobą w sieć, a każdy neuron ma średnio tysiące połączeń. Neurony przenoszą i przetwarzają sygnały elektryczne. Każdy neuron ma wiele wejść, tj. dendrytów. Sygnał docierający do neuronu przez dendryty jest w nim integrowany oraz zatrzymywany jako pojedynczy sygnał przez pojedynczy akson wyjściowy. Sygnał w aksonie jest kopiowany i wysyłany do innych neuronów przez gałąź wyjściową składającą się z synaps. Oznacza to, że akson komórki łączy się z dendrytami innych komórek poprzez synapsy,

które mają zdolność modyfikowania i przesyłania sygnałów. Te modyfikacje powodują, że neurony, do których dociera zmodyfikowany sygnał, stają się pobudzone i wchodzą w stan aktywny. W tym stanie wysyła sygnały do innych neuronów poprzez swoje aksony. Przejście ze stanu pasywnego do stanu aktywnego i generowanie zmodyfikowanych sygnałów w synapsach zachodzi następnie w neuronach w sposób skokowy, dlatego nazywa się to przejściem między stanami wzbudzenia neuronalnego. Wygenerowane sygnały docierają do innych neuronów przez synapsy i dendryty, a w tych neuronach mogą również wystąpić stany zapłonowe. Dopiero gdy skumulowany sygnał docierający do neuronu przez dendryty przekroczy pewien poziom progowy, neuron zapali się. Siła połączonego sygnału docierającego do danego neuronu i związane z nim prawdopodobieństwo zapłonu zależy od skuteczności odpowiedniej synapsy, która zależy od ich współczynnika wagowego. Zasada działania proponowanej biologicznej sieci neuronowej została przeniesiona do koncepcji sztucznej sieci. Istotna różnica między tymi dwoma typami sieci polega na tym, że w przeciwieństwie do sieci rzeczywistych, sieci sztuczne mają mniej neuronów i są podzielone na warstwy, w których te neurony się znajdują. Podział ten pomaga opisać sieć w sposób matematyczny. Każdy neuron w danej warstwie odbiera sygnały od innych neuronów w warstwie górnej i są to połączenia peer-to-peer. Każdy sygnał wprowadzony do neuronu jest mnożony przez współczynnik wagowy, który odpowiada skuteczności synapsy w neuronie biologicznym. Każdy neuron ma również określony próg, który określa intensywność, jaką należy odpalić, aby go stymulować. W neuronie obliczana jest ważona suma wejść i odejmowany jest od niej próg neuronu, a uzyskana różnica określa, czy neuron się zapali. Następnie poprzez funkcję aktywacji związaną z neuronem, konwertowany jest sygnał determinujący aktywację neuronu. Wartość sygnału obliczona przez funkcję aktywacji jest wartością wyjściową neuronu. Najczęściej stosowaną funkcją aktywacji w sieciach neuronowych jest nieliniowa funkcja logiczna. Podstawowymi warstwami sieci są warstwy wejściowe, wyjściowe i ukryte. Prosta sieć neuronowa ma budowę jednokierunkową, to znaczy sygnał przepływa z warstwy wejściowej przez ciągłą warstwę ukrytą, a następnie do warstwy wyjściowej, zapewniając tym samym stabilne zachowanie sieci w procesie obliczeniowym. Sieć neuronowa o przedstawionej strukturze nazywana jest wielowarstwowym perceptronem (MLP). Wśród operacji wykonywanych przez sieć wyróżnia się proces uczenia się, a ostateczna struktura sieci jest kształtowana w tym procesie. W wyniku tego procesu niektóre połączenia mają zerowy współczynnik wagowy, co eliminuje te połączenia z sieci [260]. W procesie uczenia zostaną wykorzystane dane ze zbioru uczącego. Najbardziej znanym przykładem algorytmu uczenia sieci neuronowej jest algorytm wstecznej propagacji (BP). Za pomocą tego algorytmu parametry sieci (współczynnik wagi i próg neuronu) można zmieniać w kolejnych krokach obliczeniowych, aby zminimalizować błąd generowany przez sieć i funkcję błędu (zwykle definiowaną jako resztkowa

suma kwadratów). Algorytm BP jest algorytmem optymalizacji gradientu i działa w sposób iteracyjny [262]. Początkowe wartości parametrów sieci są ustalane losowo. Istotnym problemem sieci uczącej, czyli dostosowywania jej do danych pomiarowych zgodnie ze zbiorem uczącym, jest jej zdolność do generalizacji. Uczenie sieci w sposób minimalizujący błędy w zbiorze uczącym nie oznacza, że będzie ona w stanie poprawnie modelować proces dla innych zbiorów danych. Jest to problem, który zawsze występuje przy tworzeniu modeli dla procesów predykcyjnych, w tym tzw. overfittingu sieci.

Oznacza to, że model sieci jest w pełni odpowiedni dla szczegółów reprezentowanych przez pojedyncze dane, ale nie modeluje już podstawowego kształtu przybliżonej relacji. Dobry model może odzwierciedlać ogólną postać poszukiwanych zależności, ale nie zawiera niewielkich zmian w danych wejściowych (zwykle zmiany trywialne). Zwykle zależy to od złożoności matematycznego opisu modelu. Większe sieci prawie zawsze mają niższe wartości błędów, ale zwykle oznacza to nadmierne dopasowanie, a nie dobrą jakość modelu. W celu uniknięcia przepełnienia sieci w modelowaniu wykorzystuje się proces weryfikacji, w ramach którego oprócz zbioru uczącego wydzielany jest dodatkowy zbiór weryfikacyjny ze zbioru danych wejściowych i wykorzystywany do niezależnej kontroli algorytmu uczącego. Jeżeli jakość uczenia się sieci uzyskana z danych uczących i danych walidacyjnych jest podobna, określony model sieci uważa się za prawidłowy. Proces walidacji to weryfikacja procesu uczenia się innego zbioru danych w celu uzyskania podobnych wyników obliczeń w celu udowodnienia jego poprawności. Po powtórzeniu każdego procesu uczenia się wykonywane są obliczenia weryfikacyjne. Jeśli błąd uczenia się zmniejsza się, a błąd walidacji zaczyna narastać podczas procesu obliczania, oznacza to, że sieć zaczęła wprowadzać zbyt wiele korekt danych uczenia się i utraciła zdolność do podsumowywania wyników uczenia się. Oznacza to, że jego opis jest zbyt skomplikowany i należy go uprościć poprzez zmniejszenie liczby warstw ukrytych lub liczby neuronów na tych warstwach. Najważniejszą informacją, którą należy wziąć pod uwagę podczas oceny sieci, jest wartość błędu weryfikacji. W celu dalszej poprawy wiarygodności określonego modelu, trzeci zestaw testowy jest wyodrębniany ze zbioru danych wejściowych. Ostateczna forma modelu (wyuczona przez zestaw uczący i sprawdzona przez zestaw walidacyjny) może być również przetestowana na zbiorze testowym. Po zakończeniu całego procesu uczenia sieci obliczenia przy użyciu zestawu testowego są wykonywane tylko raz.

Przy implementacji oraz zastosowaniu sieci neuronowych należy podjąć decyzje dotyczące:

- przygotowania danych do przetwarzania przez sieć;
- wyboru modelu sieci, czyli tzw. architektury sieci (typ sieci, struktura połączeń, liczba warstw i neuronów w warstwach, funkcja aktywacji);

- wyboru metod uczenia, czyli odpowiedniego procesu uczenia sieci;
- metod weryfikacji otrzymanych wyników.

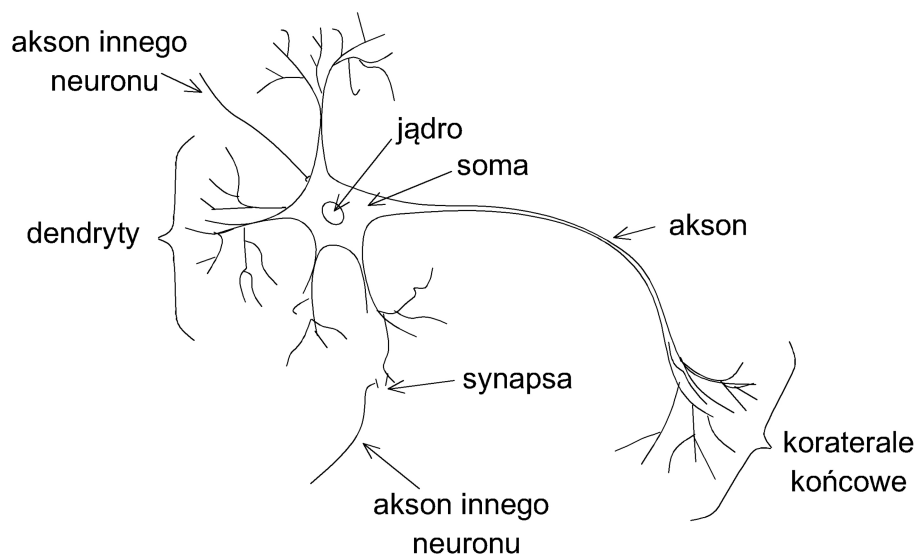
1.4.1 Podstawy budowy sieci neuronowych

Sztuczna inteligencja, a w szczególności sztuczne sieci neuronowe, to dziedzina nauk, która pomimo swojej nowatorskiej koncepcji, jest dziedziną interdyscyplinarną i odwołuje się do szeregu znanych już wcześniej mechanizmów współczesnych nauk inżynierskich. Podstawy tej dziedziny nauki bezpośrednio nawiązują do obserwacji i odkryć dokonanych przez psychologów i biologów. Zaobserwować można również liczne analogie do systemów przetwarzania informacji w złożonych, żywych organizmach [2]. Sztuczna sieć neuronowa to model matematyczny biologicznej sieci neuronowej stosowany w postaci programu komputerowego lub gotowego układu elektronicznego. W ten sposób zaprojektowany i zaprogramowany układ scalony, naśladuje zdolność ludzkiego mózgu do klasyfikowania wzorców w oparciu o przeszłe doświadczenia oraz do przewidywania i podejmowania decyzji [2], [44], [65], [110].

Mózg odbiera i przetwarza bodźce neuronalne pozyskiwane ze zmysłów, podobnie jak SSN, z tym że w przypadku SSN są to cyfrowe zbiory danych w postaci wektorów, które definiują określone części informacji [303]. Biologiczne neurony są połączone ze sobą połączeniami pomiędzy aksonami a dendrytami zwanymi synapsami, przez które złożone sygnały elektrochemiczne są przekazywane do innych neuronów. Proces ten pozwala na przejście sygnału przez ciągłą synapsę do kolejnego neuronu, który pod jego wpływem może ulec aktywacji [22]. W ten sposób wyróżniamy dwa stany każdego neuronu: stan spoczynkowy oraz stan wzbudzony, w którym wysyłany jest potencjał czynnościowy, co świadczy o tym, że neuron ma cechy dwóch stanów i niewątpliwie sprzyja jego matematycznemu odwzorowaniu [48]. Sztuczna sieć neuronowa to bardzo złożona technika modelowania, która jest wykorzystywana jako ogólny system aproksymacji reprezentujący wielowymiarowe zbiory danych w wielu praktycznych zastosowaniach. Sztuczne neurony (ang. *nodes* lub *neurons*) - matematyczny odpowiednik biologicznych komórek nerwowych ma połączenie umożliwiające wysyłanie informacji do sieci (z warstwy wejściowej przez warstwę ukrytą do warstwy wyjściowej) lub w odwrotnej kolejności (sieć rekurencyjna) [47]. Oprócz dużej liczby połączeń między neuronami charakterystyczną cechą jest dynamiczna zależność, która zachodzi na każdym etapie uczenia się. Zmiany w neuronach są przenoszone do innych neuronów [121]. Istotą inteligencji jest pamięć i uczenie się na tej podstawie, a następnie wykorzystanie tej wiedzy do celów klasyfikacji i predykcji. Sposób, w jaki przebiega proces uczenia się, polega na wzmocnieniu połączeń synaptycznych między prawą częścią kory (odpowiedzialną za określone czynności) lub kilkoma częściami

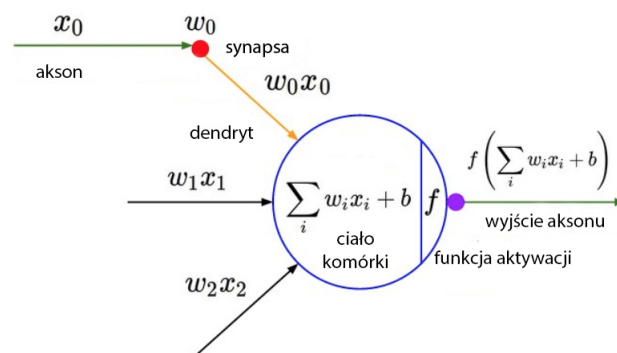
kory w kolejnych doświadczeniach. Proces ten odbywa się iteracyjnie. Wiedza zdobyta podczas uczenia się i jej wykorzystywania musi być przechowywana w pamięci, co wymaga wykonania wielu procesów w tej samej jednostce czasu. Dlatego sieć neuronowa to struktura złożona z prostych jednostek, które wykonują wiele zadań przetwarzania równoległego [37]. Niezaprzeczalną zaletą przetwarzania równoległego jest szybkość i niezawodność takich systemów [47]. Jego cechą charakterystyczną jest to, że w przeciwieństwie do dużej liczby obliczeń szeregowych z tylko jednym lub kilkoma elementami konwersji, użycie dużej liczby elementów do przetwarzania danych w tym samym czasie zwykle generuje problematyczne tzw. kolejki wejściowe. Ze względu na dużą gęstość połączeń między neuronami oraz proste operacje wykonywane przez pojedynczy neuron, błędy w obliczeniach pojedynczego neuronu są bardzo nieistotne lub nie mają wpływu na ostateczny wynik. W przypadku sekwencyjnego przetwarzania szeregowego bardzo złożone operacje i błędy obliczeniowe wykonywane przez jeden lub najwyżej kilka procesorów mogą spowodować całkowite zniekształcenie wyników.

Sztuczna sieć neuronowa o prostej strukturze może przybliżać i ekstrapolować najbardziej złożone funkcje nieliniowe oraz przewidywać lub klasyfikować wzorce. Na uwagę zasługuje zdolność sieci neuronowych do rozwiązywania problemów optymalizacji. Cechami, które sprawiają, że są one najbardziej odpowiednie do wykorzystania jako narzędzia do prognozowania wartości szeregów czasowych, są równoległe modele przetwarzania danych, odporność na błędy i zniszczenia, umiejętność uczenia się i uogólniania wiedzy, zdolność adaptacji do środowiska oraz rekurencja [303]. Ludzki mózg jest ogromną siecią połączonych ze sobą neuronów, które przetwarzają informacje i modelują świat w którym żyjemy. Mózg zawiera około 100 miliardów neuronów i mniej więcej 1000 razy więcej połączeń między nimi, nazwanych synapsami (rys. 1.26).



Rysunek 1.26: Schematyczny rysunek neuronu biologicznego [48]

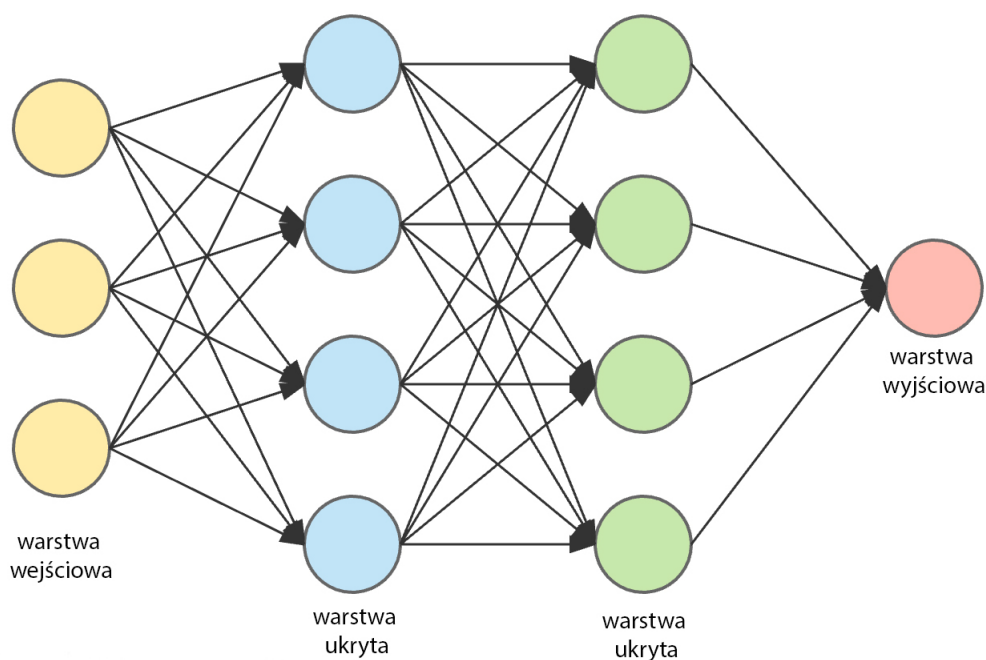
Neuron jest pryncypialną jednostką obliczeniową mózgu. W praktyce otrzymuje on sygnał wejściowy od połączonych z nim dendrytów, a następnie generuje sygnał wyjściowy przesyłając go wzdłuż jego jedyne aksonu. Ten ostatni jest, poprzez synapsy, połączony z dendrytami innych neuronów, transmitując tym samym sygnał do dalszych elementów. Moc, z jaką każda synapsa przekazuje sygnał ulega zmianie i jest determinowana przez wartości ustalone w trakcie uczenia sieci. W sumatorze wyjściowym sumowane są sygnały przekazywane przez dendryty do wejść neuronu. Jeżeli wartość sumy przekracza pewną wartość progową, to do aksonu przesyłany jest sygnał elektryczny. Zakładając, że czas transmisji sygnałów elektrycznych jest pomijalny, sygnał ten można kształtować przy użyciu statycznej funkcji aktywacji [16]. Taki model neuronu został opisany formalnie poprzez równanie (1.46) oraz zobrazowany na rysunku 1.27. Sygnał wyjściowy neuronu y jest ważoną sumą N sygnałów wejściowych x_i powiększoną o składową stałą b , modelowaną przez funkcję aktywacji f . Nie mający polskiego określenia *bias* jest wartością modelującą próg, powyżej którego biologiczny neuron transmituje sygnał wyjściowy.



Rysunek 1.27: Matematyczny model pojedynczego neuronu [15]

$$y = f\left(\sum_{i=1}^N w_i x_i + b\right) \quad (1.46)$$

Sieci neuronowe są modelowane jako zbiory neuronów połączonych w tzw. skierowany graf acykliczny. Sygnały wyjściowe niektórych neuronów trafiają bezpośrednio na wejścia kolejnych. Ponadto, sieci neuronowe mają budowę warstwową. Zazwyczaj każda z nich jest w pełni połączona z warstwami sąsiadującymi, tj. każdy neuron danej warstwy jest połączony z wszystkimi neuronami warstw sąsiadujących. Jednocześnie nie on jest połączony z żadnym neuronem swojej warstwy, rys. 1.28. Sieć o takiej strukturze nosi nazwę *Multi Layer Perceptron (MLP)*, a jej nazwa nawiązuje do jednowarstwowej sieci z progową funkcją aktywacji - *Perceptronu*. Termin ten może budzić wieloznaczne skojarzenia, bo w sieciach MLP progowe funkcje aktywacji zazwyczaj nie są w ogóle stosowane [309].



Rysunek 1.28: Schemat sztucznej sieci neuronowej [16]

Warstwy sieci neuronowej można podzielić na:

1. **Warstwę wejściową**, która przechowuje dane dostarczone do sieci.
2. **Warstwę ukryta**, czyli standardową warstwę przetwarzającą informacje według podanego wyżej schematu.
3. **Warstwę wyjściową**, która przechowuje wyniki sieci.

Określenie sieć n -warstwowa w literaturze odnosi się do sieci o n warstwach ukrytych. Transmisja sygnału od warstwy wejściowej do warstwy wyjściowej określana jest terminem propagacji w przód (ang. *Forward propagation*). Jednym z głównych powodów organizacji sieci neuronowej w warstwy jest prostota oraz efektywność wynikająca z użycia takiego modelu [210]. Dla danego modelu neuronu (1.46) można określić strukturę M -neuronowej warstwy w następujący sposób:

$$y_j = f\left(\sum_{i=1}^N w_i x_i + b_j\right), \quad j = 0 \dots M - 1 \quad (1.47)$$

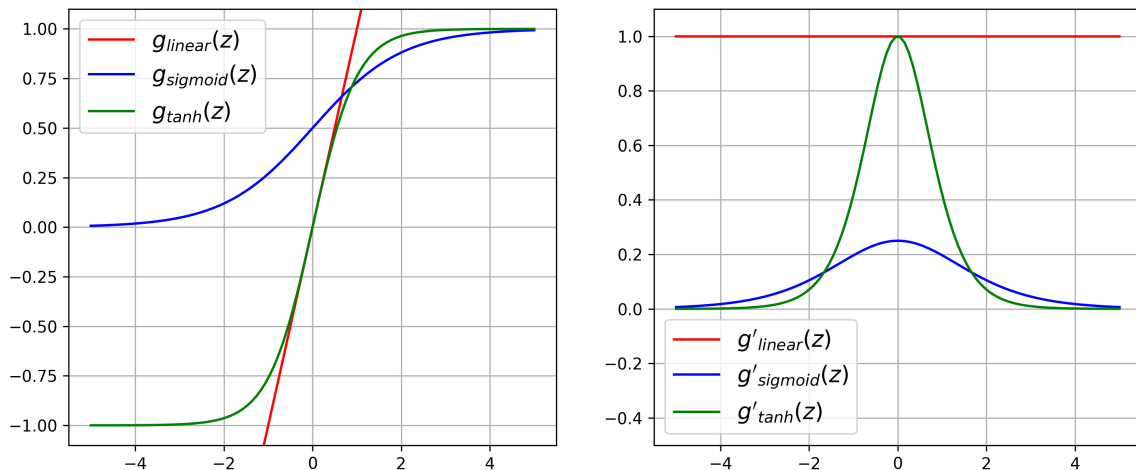
W postaci macierzowo-wektorowej sprowadza się to do:

$$y = f(W * x + b) \quad (1.48)$$

Gdzie $y \in \mathcal{R}^M$ jest wektorem wyjść, $x \in \mathcal{R}^M$ wektorem wejść, $W \in \mathcal{R}^{M \times N}$ macierzą wag, $a, b \in \mathcal{R}^N$ wektorem biasu.

1.4.2 Funkcje aktywacji

Rolą funkcji aktywacji jest zdecydowanie, w jakim stopniu dany neuron jest pobudzony bieżącymi, docierającymi do niego wartościami. Jej wybór ma jednak istotny wpływ na dalszy przebieg uczenia sieci neuronowej. W dalszej kolejności, wskazując tym samym ich wady i zalety na tle konkurencyjnych rozwiązań (rys. 1.29, 1.30), przedstawiono kilka z nich.



Rysunek 1.29: Przebieg powszechnie stosowanych funkcji aktywacji wraz z ich gradientem

Funkcja liniowa

$$f(x) = ax \quad (1.49)$$

Prosta funkcja używana na początku historii sieci neuronowych. Przez brak nieliniowości znacznie zmniejsza możliwości sieci. Powoduje niestabilną zbieżność modelu podczas nauki, gdyż w żaden sposób nie ogranicza przetwarzanych wartości. Ponadto wiele warstw sieci o liniowej funkcji aktywacji jest równoważne jednej liniowej warstwie, więc ten rodzaj aktywacji nie ma zastosowania w wielowarstwowych sieciach. **Funkcja progowa**

$$f(x) = \begin{cases} a, & x \geq 0 \\ 0, & x < 0 \end{cases}, \quad (1.50)$$

Funkcja wolna od wymienionych wad funkcji liniowej, jednak poprzez swoją nieciągłość oraz zerowy gradient powoduje duże problemy z nauką modelu. **Funkcja sigmoidalna**

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \quad (1.51)$$

Najczęściej stosowaną funkcją aktywacji w sztucznych sieciach neuronowych jest funkcja sigmoidalna. W zależności od charakteru danych wejściowych stosuje się również inne

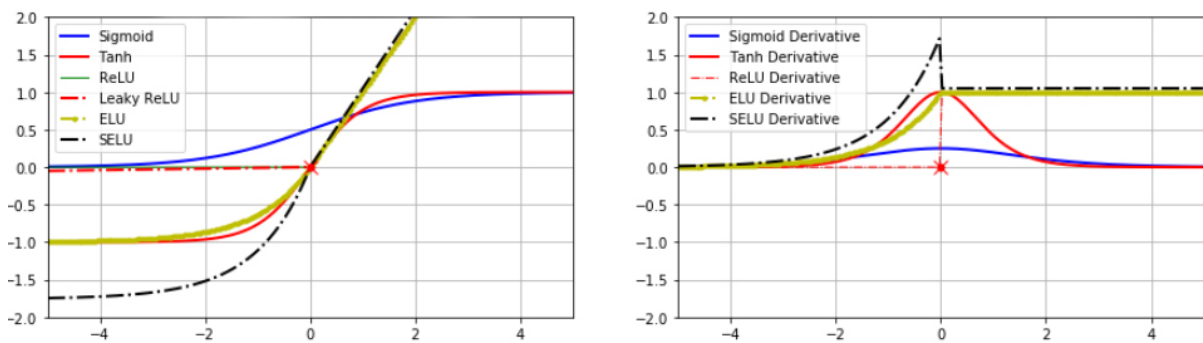
funkcje, które mogą skutkować lepszym odwzorowaniem analizowanych zależności [212]. Jest ona ciągłym przybliżeniem funkcji progowej, ograniczającym rzeczywiste wartości do zakresu pomiędzy 0 a 1. Charakterystyka tej funkcji jest również łatwo interpretowalna biologicznie, tj. od braku aktywności neuronu (0) do maksymalnej częstotliwości wysyłania impulsów (1). Funkcja ta w ostatnim czasie straciła jednak na znaczeniu, ponieważ:

- Gdy charakterystyka zbliża się do wartości ekstremalnych, tj. jej wartości są bliskie 0 lub 1, wartość jej gradientu jest prawie bliska zeru (rys 1.29), co w praktyce bardzo utrudnia naukę danego neuronu oraz sprawia, że neuron nie przesyła sygnału do wcześniejszych warstw podczas propagacji wstecznej. Neurony, które na wyjściu osiągają wartości równe "zawsze 0" lub "zawsze 1", są nazywane "martwymi"[80]. Gdy początkowe wartości współczynników wagowych lub współczynnik nauki sieci osiągają zbyt duże wartości, większość neuronów w dużym tempie osiąga nasycenie, w efekcie czego, sieć traci zdolność do nauki [174].
- Wartości wyjściowe funkcji sigmoidalnej nie są skoncentrowane wokół zera, co sprawia, że kolejne warstwy sieci otrzymują dane o coraz większej średniej wartości, co znowu często prowadzi do zaistnienia zjawiska zygzakowania podczas optymalizacji modelu.

Funkcja Tanh

$$f(x) = \tanh(x) = 2\sigma(x) - 1 \quad (1.52)$$

Tangens hiperboliczny można interpretować jako przeskalowaną funkcją sigmoidalną. Funkcja ta również nasycą się w tych samych przedziałach, lecz wyjścia neuronów o tej funkcji aktywacji są skupione wokół zera.



Rysunek 1.30: Przebieg powszechnie stosowanych funkcji aktywacji wraz z ich gradientem

Funkcja Softplus

$$f(x) = \sum_{i=1}^{\infty} \frac{1}{1 + e^{-(x-i+0.5)}}, \approx \ln(1 + e^x) \quad (1.53)$$

Funkcja softplus jest aproksymacją sumy wielu sigmoid przesuniętych względem siebie na osi x , co można przedstawić jako suma wielu neuronów z sigmoidalną funkcją aktywacji o tych samych wagach, lecz różnych biasach.

Funkcje ReLU

$$f(x) = \max(0, x) \quad (1.54)$$

Funkcje *Rectified Linear Units* zyskały na znaczeniu w ostatnim czasie. Funkcje te są przybliżeniem funkcji Softplus, co jest realizowane przez proste progowanie w zerze. Czynność ta prowadzi do przyspieszenia zarówno implementacji jak i operacji arytmetycznych w porównaniu do funkcji sigmoidalnej i tanh. Dowiedzono też, że znacznie przyspiesza zbieżność stochastycznej metody gradientu prostego. Z innego punktu widzenia, neuron może łatwo "umrzeć", gdy jego wagi dojdą do stanu, w którym będzie on zawsze zwracał 0. Osiągnięcie takiego stanu jest dla neuronu nieodwracalne. Mała wartość współczynnika nauki prowadzi do zmniejszenia prawdopodobieństwa zaistnienia tego problemu [196].

Funkcja LeakyReLU

$$f(x) = \max(ax, x) \quad (1.55)$$

Funkcja ta jest udoskonaleniem funkcji ReLU. Usiłuje ona rozwiązać problem "umierających" neuronów. W zakresie w którym $x < 0$ funkcja aktywacji cechuje się małą wartością dodatniego gradientu ($a \approx 0.01$), neuron może opuścić, nieodwracalny w ReLU, stan.

Podobne rozwiązania zastosowano w następujących funkcjach:

1. **PReLU** (*Parametric Rectified Linear Unit*), gdzie a jest oddzielne dla każdego neuronu i dostosowuje się podczas nauki modelu.
2. **ELU** (*Exponential Linear Unit*), w którym krzywa w II ćwiartce wykresu charakteryzuje się kształtem eksponencjalnym.

Funkcja Softmax

$$f(x)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}}, \quad j = 1 \dots K \quad (1.56)$$

Funkcja softmax, inaczej zwana znormalizowaną funkcją wykładniczą, jest uogólnieniem funkcji sigmoidalnej, Przekształca ona K -wymiarowy wektor wartości rzeczywistych do K -wymiarowego wektora wartości w zakresie $(0; 1)$, którego wartości sumują się do 1. Funkcja ta jest często realizowana w ostatniej warstwie sieci wykorzystywanej do zadań klasyfikacji, gdy klasy wykluczają się wzajemnie. Wartość $f(x)_j$ charakteryzuje prawdopodobieństwo przynależenia wartości wejściowej do klasy j .

1.4.3 Metody uczenia sieci neuronowych

Prosta struktura sieci jednokierunkowych czyni je powszechnie stosowanymi. Ponadto metody uczenia ich należą również do metod dobrze rozpoznanych w innych dziedzinach inżynierskich i są łatwe w realizacji. Założywszy dostęp do odpowiedniej liczby wektorów trenujących można dostosować wagi sieci neuronowej w taki sposób, by z dużą poprawnością uogólniała nowe, dotąd niepoznane wzorce. W przypadku konieczności wyliczenia wag jednokierunkowych sieci wprzód powszechne jest użycie metod gradientowych. Techniki te służą w propagacji wstecznej jako iteracyjny mechanizm korekcji wag, który w każdym kroku (zakładając zbieżność algorytmu) prowadzi do poprawy zdolności klasyfikacyjnych modelu. Oznacza to, że wraz z wykonaniem kolejnych iteracji algorytmu całkowity błąd niedopasowania wag ulega zmniejszeniu. W praktyce procedura ta jest poszukiwaniem minimum globalnego kierując się wyznaczonym w każdym kroku tzw. kierunkiem największego spadku poprzez gradient funkcji w kolejnych punktach przestrzeni poszukiwań. W praktyce jednak może zachodzić skłonność do "utykania" w minimach lokalnych funkcji błędu sieci i tym samym oddalania się od minimum globalnego. Zanim sieć neuronowa będzie zdolna do realizacji zadania klasyfikacji musi zostać odpowiednio przygotowana. Przygotowanie sieci odbywa się w procesie uczenia, w trakcie którego ustalane są wartości wag wszystkich neuronów sieci. Procedura uczenia oparta jest o zgromadzone wejściowe dane wzorcowe, dla których tworzy się odpowiadającą (pożądaną) macierz wyjść. Jedną z powszechniejszych metod uczenia sieci jest metoda wstecznej propagacji błędów. Jest to metoda gradientowa poszukiwania kierunku najszybszego spadku funkcji błędu. Uczenie sieci neuronowych można sprowadzić do poziomu optymalizacji pewnej funkcji kosztu, którą należy wcześniej zdefiniować wskazując zmienne decyzyjne, które są przedmiotem zmian realizowanych zazwyczaj iteracyjnie. Można zatem stwierdzić, że uczenie klasyfikatora opiera się na minimalizowaniu jakiejś funkcji, zwanej funkcją błędu lub kosztu. Po zdefiniowaniu owej funkcji kosztu, uczenie sieci opiera się na procedurze minimalizacji błędów metodami gradientowymi optymalizacji ciągłej. Spośród dostępnych algorytmów uczenia maszynowego wyróżnić można kilka metod uczenia, do których porządkowanie jest uzależnione od tego, czy model ma nauczyć się wnioskować na podstawie opisanych danych, czy też wnioskować odnosząc się do ukrytej struktury danych nieetykietowanych. Techniki te można podzielić według następujących kryteriów:

1. **Uczenie nadzorowane** (ang. *Supervised learning*). Do przynależności do tej grupy metod uczenia decyduje fakt czy pożądaný rezultat jest dostarczony razem z danymi wejściowymi podczas uczenia. Dostarczając obie informacje za jednym razem możliwym jest obliczenie błędu pomiędzy pożądanym rezultatem, a tym otrzymanym od sieci neuronowej i na tej podstawie wyliczyć korektę do zaktualizowania jej wag.

2. **Uczenie nienadzorowane** (ang. *Unsupervised learning*). W tej grupie algorytmów uczenia sieci neuronowej podawane są tylko dane wejściowe, a zadaniem sieci jest wyznaczenie struktury dostarczonych danych bez pomocy z zewnątrz. W uczeniu nadzorowanym istotne są zatem te informacje, które kierują wynik ku wartości pożądaney. W nienadzorowanym uczeniu natomiast istotne są te informacje, które pozwalają jak najniższym kosztem wyznaczyć rozkład danych wejściowych. Zazwyczaj w metodach tego typu wbudowane są tzw. wąskie gardła (*Bottleneck*), co uniemożliwia sieci przechowywanie nieistotnych informacji.
3. **Uczenie ze wzmocnieniem** (ang. *Reinforcement learning*) jest metodą podobną do uczenia nadzorowanego, jednak zamiast podania wprost pożądaných danych wyjściowych, sieć otrzymuje wskaźniki jakości otrzymanego rozwiązania, czasem nazywany nagrodą. Celem sieci jest numeryczna maksymalizacja tej nagrody. Ta forma nauki jest jednak rzadziej stosowana.

1.4.3.1 Funkcja kosztu

Funkcja kosztu (ang. *cost function*) definiuje miarę błędów popełnionych przez sieć. W przypadku regresji liniowej koszt można opisać jako średnią kwadratów różnic (ang. *mean square error*) pomiędzy pożądanymi rezultatami (Y), a tymi wnioskowanymi przez sieć (Z):

$$\mathcal{L}(Z, Y) = \frac{1}{N}, \sum_{n=1}^N (y_n - z_n)^2 \quad (1.57)$$

W przypadku klasyfikacji dyskretnej korzysta się z tzw. średniej entropii krzyżowej (ang. *cross entropy*):

$$\mathcal{L}(Z, Y) = -\frac{1}{N}, \sum_{n=1}^N \left(y_n \log z_n + (1 - y_n) \log(1 - z_n) \right) \quad (1.58)$$

1.4.3.2 Uogólniona reguła delta

Reguła delta sprowadza się do zastosowania metody gradientu prostego do aktualizacji wag sieci neuronowej. Z tego powodu zmianę wagi j -tego neuronu względem jego i -tego wejścia można zapisać jako:

$$w_{ij} \rightarrow w'_{ij} = w_{ij} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}} \quad (1.59)$$

A zmianę biasu j -tego neuronu można opisać jako:

$$b_j \rightarrow b'_j = b_j - \eta \frac{\partial \mathcal{L}}{\partial b_j} \quad (1.60)$$

Regułę można sprowadzić do podążania krokami o długości η w kierunku antygradientu (ujemnego wartości gradientu) funkcji kosztu ($\frac{\partial \mathcal{L}}{\partial w_{ij}}$), czyli w kierunku najszybszego spadku wartości tej funkcji. Długość kroku η określana jest mianem współczynnika nauki (ang. *learning rate*), a jego wartość ma wpływ na zbieżność procesu minimalizacji i powinna być odpowiednio dobrana do stosowanej architektury sieci. W praktyce stosuje się często dodatkowe zabiegi mające na celu przyspieszenie zbieżności algorytmu minimalizacji funkcji kosztu.

Reguła jest nazywana uogólnioną, ponieważ jej pierwotna definicja uwzględniała tylko funkcję kosztu, określoną jako średnia kwadratów różnic.

1.4.3.3 Wsteczna propagacja błędu

Reguła delta wykorzystuje funkcję kosztu jako miarę błędu popełnianego przez neuron. Ma to swoje uzasadnienie, gdy sieć jest jednowarstwowa lub gdy dotyczy to ostatniej warstwy sieci wielowarstwowej. W takiej sytuacji pojedynczy neuron ma bezpośredni wpływ na ostateczną wartość funkcji kosztu. Zakładając, że przedmiotem naszych zainteresowań jest sieć wielowarstwowa należy w pierwszej kolejności określić, jaki wpływ miał konkretny neuron na wyznaczony błąd.

Metoda wstecznej propagacji błędu (ang. *backpropagation*) propaguje błąd otrzymany w ostatniej warstwie do warstw wcześniejszych. Metodologia aktualizacji wag (1.59) zostaje zachowana, jednak pochodna cząstkowa ($\frac{\partial \mathcal{L}}{\partial w_{ij}}$) jest określana przez regułę łańcuchową pochodnych funkcji złożonych. Propagację wstecz można przedstawić w następujący sposób. Na podstawie reguły delta wyliczona jest zmiana wag ostatniej warstwy, następnie liczony jest wpływ warstwy przedostatniej na błędne wagi warstwy ostatniej i na tej podstawie wagi warstwy przedostatniej są aktualizowane. W analogiczny sposób jest liczony wpływ wag warstwy wcześniejszej itd., aż do wyliczenia poprawki wag warstwy pierwszej.

1.4.3.4 Warianty metody gradientu prostego

W metodzie gradientu prostego (ang. *gradient descent*) zakłada się posłużenie gradientem i wyliczenie wartości funkcji kosztu dla całego zbioru danych uczących i na tej podstawie przeprowadza się aktualizację wag sieci. Taka strategia określana jest mianem *Batch Gradient Descent*. W praktyce metoda ta jest jednak dość powolna, co nasila się zwłaszcza przy osiagających dziś ogromne rozmiary zbiorach danych uczących. Z tego powodu przedstawione zostały usprawnienia tej metody, polegające na mniejszych, za to częstszych krokach w kierunku gradientu przybliżonego. Na podstawie sposobu aproksymacji gradientu można wyróżnić:

1. **Stochastic Gradient Descent (SGD)**, która aproksymuje gradient na podstawie tylko jednego przykładu ze zbioru uczącego. Proces aktualizacji wag jest szybki i bardzo niedokładny. Prowadzi to do zmniejszenia stabilności algorytmu, ale również wprowadza „szum” do metody, co redukuje znaną wadę metody bazowej - utknięcia w minimach lokalnych. Jest to powszechna bolączka wielu iteracyjnych metod optymalizacji ciągłej, zwłaszcza metod gradientowych. Utykanie w minimach lokalnych (w istocie rzeczy szukamy minimum globalnego, gdyż tylko ono jest równoznaczne optymalnym współczynnikom wagowym sieci) jest dość powszechne i wynika z nieoptymalnych wartości parametrów roboczych algorytmu, tj. długości kroku i kierunku poszukiwań. Nie ma bowiem jednoznacznej metody wyznaczenia "właściwej" dla danego przypadku długości kroku i kierunku poszukiwań.

W niektórych przypadkach zbyt długi krok prowadzi do ryzyka "przekroczenia" miejsca lokalizacji minimum globalnego a z kolei zbyt mała wartość długości kroku prowadzi do znacznego zwiększenia nakładu obliczeniowego i tym samym ryzyka niezalezienia rozwiązania w skończonym czasie [40]. W praktyce stosuje się kompromis w postaci *Mini-Batch Gradient Descent* lub dodaje silny wpływ opisanego niżej momentum. W wielu praktycznych przypadkach stosuje się również metody pomocnicze wyznaczające wstępną wartość położenia minimum globalnego, tj. estymaty inicjującej, co w rzeczywistości przyspiesza zbieżność celowego algorytmu minimalizacji funkcji kosztu, bo nie jest on zmuszony do przeszukiwania kompletnej przestrzeni rozwiązań.

2. **Mini-Batch Gradient Descent** aproksymuje wartość gradientu na podstawie małego podzbioru zbioru uczącego. Fizyczny rozmiar takiego wolumenu danych (ang. *batch size*) kształtuje się w granicach 10-100. Zastosowanie takiej ilości danych uczących ciągle jest dużo szybsze niż zastosowanie całego zbioru, jednocześnie jest dużo stabilniejsze niż SGD. W przypadku klasyfikacji istotne jest, aby w zbiorze danych znalazła się równa, tj. reprezentatywna liczba przykładów każdej z klas.

W obydwu wariantach konieczne jest wybieranie podzbioru metodą losową. Innym udoskonaleniem metody jest tak zwane *momentum*. Jest to uwzględnienie ostatniej dokonanej zmiany wagi w bieżąco dokonywanej, gdzie parametr $\zeta \in (0; 1)$ jest siłą tego wpływu.

$$\Delta w'_{ij} = \eta \frac{\partial \mathcal{L}}{\partial w_{ij}} + \zeta \Delta w_{ij} \quad (1.61)$$

Komponent ten ma za zadanie kontrolować zbieżność minimalizacji przez przeciwdziałanie tzw. zjawisku zygzakowania. W trakcie procesu uczenia sieci użycie całego zbioru danych, abstrahując od omówionych powyżej metod, nazywane jest mianem pojedynczej *epoki* uczenia.

1.4.3.5 Inicjalizacja wag

Wszystkie neurony jednej warstwy otrzymują ten sam wektor danych wejściowych, lecz uzyskują inne wartości wyjściowe tylko dlatego, że każdy z nich dysponuje innym zestawem wag. Z tego powodu nie zaleca się inicjalizowania wag neuronów przy użyciu tej samej wartości. W początkowej fazie uczenia sprowadzałoby się to do powstania identycznych, a w dalszej kolejności bardzo podobnych wartości każdego neuronu danej warstwy, a to prowadziłoby do obiektywnych trudności w uczeniu sieci. By temu zapobiec, stosuje się proces nazywany łamaniem symetrii (ang. *breaking symmetry*). W rzeczywistości jest to losowanie początkowych wartości wag z małego zakresu, który jednocześnie musi być wystarczająco duży, by złamać symetrię, a jednocześnie na tyle mały, aby nie wprowadzać sieci do stanu, którego ciężko będzie ją oduczyć. Dla sieci głębokich dobór wag ma jeszcze większe znaczenie. Jeśli wagi są zbyt małe, to siła sygnału maleje wraz z przechodzeniem przez kolejne warstwy i sygnał powoli staje się bezużyteczny. Z kolei, gdy wagi są zbyt duże, to siła sygnału rośnie przechodząc przez kolejne warstwy, co również zmniejsza jego przydatność.

Metodologia doboru odpowiedniego zakresu był tematem prac badawczych [290], [178]. Dowiedziono, że aby siła sygnału była stała, wariancja wag powinna wynosić $\frac{1}{n_{in}}$ w czasie propagacji w przód i adekwatnie $\frac{1}{n_{out}}$ w czasie propagacji wstecznej, gdzie n_{in} oraz n_{out} oznaczają odpowiednio liczbę wejść i liczbę wyjść neuronu. Aby spełnić oba warunki zaproponowano generowanie początkowych wag według:

$$w \sim \mathcal{N}\left(0, \frac{2}{n_{in} + n_{out}}\right) \quad (1.62)$$

Korzystając jednak z funkcji aktywacji *ReLU*, lepsze rezultaty osiąga się, gdy proces inicjalizacji wag przebiega zgodnie z poniższym schematem:

$$w \sim \mathcal{N}\left(0, \frac{2}{n_{in}}\right) \quad (1.63)$$

Przedstawione metody inicjalizacji wag opierają się na wielu założeniach co do architektury SSN. Optymalny zakres wartości inicjujących wag jest od niej bezpośrednio zależny. Oznacza to, że zachodzi konieczność empirycznego sprawdzenia, czy niewielkie zmiany wartości tych parametrów nie poprawią skuteczności procesu uczenia.

1.4.3.6 Regularyzacja

Z uczeniem sieci neuronowej w sensie uczenia modelu statystycznego, wiąże się kilka istotnych pojęć:

1. **Pojemność** (ang. *capacity*) jest własnością sieci determinowana przez jej architekturę oraz wszystkie jej metaparametry i odnosi się do możliwości modelowania przez

się dowolnych zależności zachodzących w wektorach danych. Można ją również zinterpretować jako ilość informacji, która może być przechowywana w sieci i na której podstawie sieć może wnioskować.

2. **Niedotrenowanie** (ang. *underfitting*) występuje gdy model jest zbyt prosty, ma zbyt małą pojemność i nie jest w stanie prawidłowo aproksymować zbyt skomplikowanej struktury danych. Rozwiązaniem jest zwiększenie pojemności tego modelu. Jest to możliwe na przykład przez zwiększenie rozmiaru bądź liczby warstw sieci neuronowej.
3. **Przetrenowanie** (ang. *overfitting*) jest nadmiernym dopasowaniem modelu do danych uczących. Model ma zbyt dużą pojemność i oprócz istotnej struktury zaprezentowanych mu danych, udaje mu się nauczyć także „szumu danych”. Przetrenowanie można uchwycić stosując walidację, natomiast przeciwdziałać mu można zmniejszając model lub też ograniczając możliwości modelu przez regularyzację.
4. **Regularyzacja** to ogólna nazwa grupy algorytmów ograniczających, czy też regulujących pojemność sieci w trakcie uczenia. Proste zmniejszenie wymiarów sieci zapobiega przetrenowaniu, lecz także zmniejsza praktyczne możliwości sieci. Preferuje się więc metody regularyzacji, które starają się ograniczać model w miejscach najbardziej podatnych na przetrenowanie oraz tylko w czasie uczenia, co nie zmniejsza możliwości nauczonej już sieci.

W sieciach neuronowych powszechnie stosuje się następujące metody regularyzacji:

1. **Regularyzacja L2** jest najbardziej powszechną formą regularyzacji. Implementowana jest przez dodanie parametru kary do funkcji kosztu na kwadrat wartości wszystkich wag modelu, tj. dodanie członu $\frac{1}{2} \lambda w^2$, gdzie λ jest parametrem określającym siłę tej regularyzacji. Regularyzację L2 można interpretować jako nakładanie parametru kary na wagi o znacznie większej wartości od pozostałych i preferowanie ich równomiernego rozłożenia. Skłania to sieć do korzystania z wszystkich wymiarów danych wejściowych.
2. **Regularyzacja L1** w implementacji podobna do L2, lecz dodająca człon $\lambda|w|$. Możliwe jest łączenie obu poprzez dodanie $\lambda_1|w| + \lambda_2 w^2$. Regularyzacja L1 prowadzi do faktu, że wektory wag stają się bardziej rzadkie podczas optymalizacji (większość wartości w wektorze wag neuronu jest bliska zeru). Inaczej mówiąc, neurony używają tylko kilku najistotniejszych wejść i stają się niezależne od szumu w innych.
3. **Max-norm** wyznacza górną granicę wartości wektora wag dla każdego neuronu i skaluje ten wektor w dół, jeśli wartość graniczna zostaje przekroczona. Ograniczenie można przedstawić jako $\|\vec{w}_2\| < d$, gdzie d jest górną granicą. Głównym atutem

tego rodzaju regularyzacji jest uniemożliwienie „eksplozji” wag, gdy współczynnik nauki osiąga zbyt dużą wartość. Dla niektórych modeli pozwala to na bezproblemowe zwiększanie jego wartości, co znacznie prowadzi do przyspieszenia procesu uczenia.

4. **Dropout** jest prostą i bardzo efektywną metodą przedstawioną pierwszy raz w [246]. Stanowi ona rozszerzenie powyżej opisanych metod. Podstawową jej ideą jest losowe omijanie pewnych neuronów podczas uczenia sieci (rys. 1.31). W trakcie procesu uczenia pozostawiona jest tylko część p neuronów warstwy, a pozostałe $1 - p$ są ignorowane. Procedura ta realizowana jest przez nałożenie binarnej maski r_n na wartości wyjściowe każdej warstwy:

$$y'_n = \frac{r_n * y_n}{p} \quad (1.64)$$

Zmodyfikowany wektor wyjść y'_n jest używany dalej w propagacji w następujący sposób:

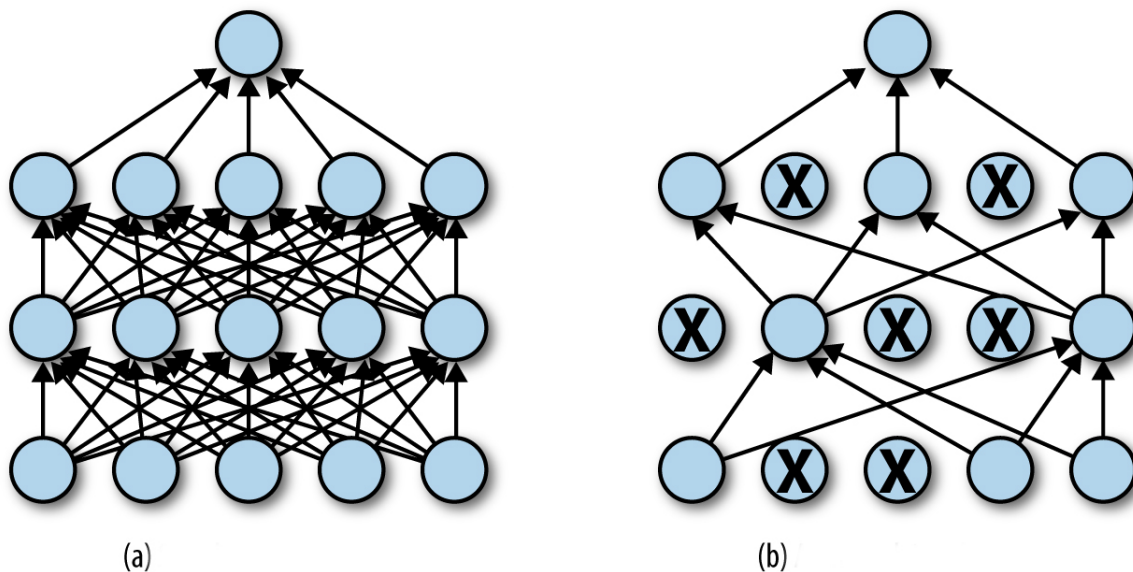
$$y_{n+1} = f(w_n y'_n + b_n) \quad (1.65)$$

Maska r_n jest inna dla każdej warstwy i generowana przy każdej propagacji w przód zgodnie z:

$$r_n \sim \text{Bernoulli}(p) \quad (1.66)$$

Metoda ma kilka interpretacji, interpretujących ją w różny sposób, rys. 1.31:

- Neurony nie mogą polegać na innych neuronach tej samej warstwy, dlatego starają się niezależnie wyodrębnić istotne informacje z danych wejściowych. Cechy te są bardziej ogólne i nadmiarowe. Pozwala to na przekazywanie tych samych informacji kolejnym warstwom w różnych formach. W praktyce ułatwia to wnioskowanie.
- Zastosowanie techniki dropoutu może być też interpretowane jako próbkowanie podsieci z kompletnej topologii sieci (rys. 1.31). Według tego kryterium oceny, całą sieć można traktować jako duży zbiór modeli o wspólnych parametrach, a wynik otrzymany przez pełną sieć jako uśredniony rezultat wszystkich modeli.
- Z innego punktu widzenia można dropout przedstawić jako szum dodany do informacji na wszystkich warstwach sieci. Sieć stara się przeprowadzić proces uczenia mimo obecności szumu, więc zauważa tylko istotne informacje, których wpływu szum nie zdołał zredukować.



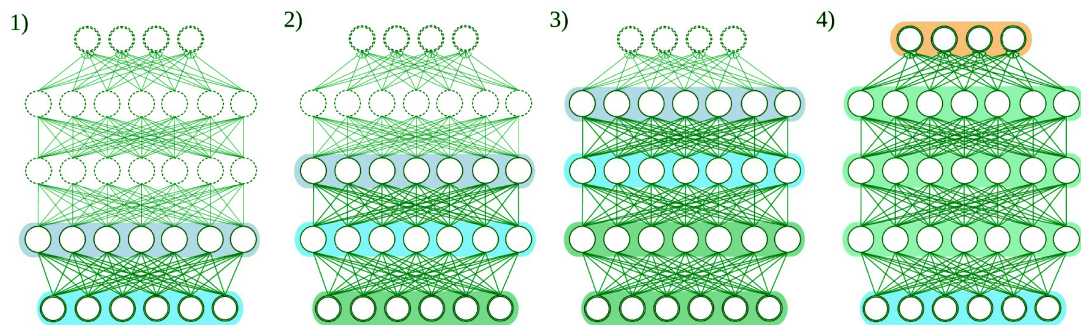
Rysunek 1.31: (a) Typowa topologia połączeń sieci wielowarstwowej, (b) topologia zoptymalizowana dzięki technice dropout

1.4.3.7 Etapy uczenia głębokiej sieci neuronowej

Wsteczna propagacja błędu jest algorytmem uzależnionym od wyliczonego gradientu funkcji aktywacji. Gdy jego wartość jest mała, wstrzymywana jest propagacja. Prowadzi to w dalszej kolejności do minimalnych lub nawet zerowych zmian w wagach neuronów warstw początkowych. Neurony przynależne do warstw wyższych, reakcję podejmują na podstawie warstw niższych, dlatego też, gdy one nie potrafią się nauczyć poprawnej reakcji, uczenie całej sieci jest niemożliwe. Niedogodność ta jest określana mianem *problemem zanikającego gradientu*. Możliwością prowadzącą do rozwiązania tego problemu jest rozpoczęcie nauki nienadzorowanym uczeniem wstępnym, a następnie dostrojenie tak otrzymanego modelu poprzez nadzorowane "douceenie".

1. **Uczenie wstępne** (ang. *Pre-training*) wykonywane jest przez nienadzorowane uczenie kolejno, dla każdej ukrytej warstwy sieci (rys. 1.32). W pierwszej kolejności uczona jest pierwsza warstwa ukryta w taki sposób, aby jak najlepiej reprezentowała dane wejścia. W kolejnym etapie, wyuczoną warstwę używamy jako warstwę wejściową dla warstwy kolejnej, powtarzając schemat aż do ostatniej warstwy ukrytej. Jest to tzw. algorytm zachłanny (ang. *greedy algorithm*) i nie aktualizuje on wag w już wstępnie nauczonych warstwach. Pomimo tego wagi sieci po wstępnym uczeniu są dużo lepiej dostosowane do występujących w zbiorze uczącym danych, niż byłyby w drodze losowej inicjalizacji.
2. **Douceenie** (ang. *Finetuning*) jest uczeniem dostrajającym wagi zainicjalizowane przez uczenie wstępne w taki sposób by były lepiej dopasowane do rozwiązywanego

problem. W wariacie nauki z nauczycielem z wstępnie nauczonych warstw można zbudować głęboką sieć MLP. Można je również wykorzystać do zbudowania głębokich modeli uczonych w sposób nienadzorowany. Do najbardziej powszechnych należą meto*a**Stacked denoising Autoencoders* oraz algorytm *Deep Belief Networks*.

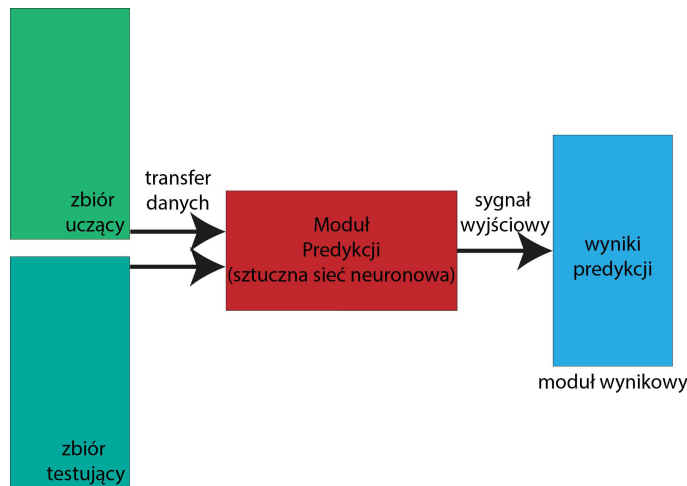


Rysunek 1.32: (1-3) Wstępne szkolenie kolejnych warstw, (4) douczenie całości modelu

Zastosowanie ostatnich dokonań w obszarze technik uczenia takich, jak funkcja aktywacji ReLU wraz z regularyzacją dropout, pozwala zrezygnować z fazy uczenia wstępnego, gdy możliwe jest użycie dużej ilości danych etykietowanych. W przypadku gdy mamy do czynienia z mniejszym zbiorem danych etykietowanych, a zwłaszcza gdy jest on uzupełniony nieetykietowanym zbiorem, stosowanie wstępnego uczenia jest wciąż zasadne.

1.4.4 Sieci neuronowe a prognozowanie szeregów czasowych

Prognozowanie (predykcja) jest niezwykle ważnym elementem, występującym w codziennym życiu człowieka. Głównym celem prognozowania jest podjęcie właściwej decyzji lub ograniczenia ryzyka mającego związek z podejmowaniem decyzji. Proces prognozowania jest procesem złożonym i wieloetapowym, co należy uwzględnić podczas projektowania neuronowego systemu predykcyjnego. Etapy te można przedstawić w sposób następujący: 1. zdefiniowanie i określenie odpowiedniego problemu, dla którego będzie przeprowadzana prognoza. 2. Zgromadzenie wektorów danych źródłowych. 3. Analiza konkurencyjna metod predykcyjnych oraz najodpowiedniejszej metody dla wybranego problemu. 4. Zaprojektowanie i implementacja prognostycznego z wykorzystaniem wcześniej zgromadzonych danych. 5. Wyznaczenie prognozy. 6. Ocena wyników i ewentualne poprawienie metody predykcji lub danych wejściowych [174].



Rysunek 1.33: Prognozowanie z użyciem sztucznych sieci neuronowych

Źródło: opracowanie własne.

Jedną z metod nieklasycznych jest częściowe (tj. w połączeniu z innymi technikami) lub całkowite wykorzystanie sztucznych sieci neuronowych do prognozowania [144], [154]. Analiza literaturowa tematu zastosowania sztucznych sieci neuronowych w prognozowaniu ekonomicznym pozwala stwierdzić, że obszarem tych zastosowań są te dziedziny, w których można uzyskać bardzo obszerne szeregi czasowe wybranych charakterystyk realizowanego procesu [21]-[299]. W tych okolicznościach rodzi się zatem pytanie: jak analizować szeregi czasowe o mało licznych zbiorach wartości realizowanej cechy? W wielu praktycznych okolicznościach przecież pozyskanie dostatecznie długiej realizacji czasowej nie jest możliwe [174]. Nie jest to w rzeczywistości problem, który by należało marginalizować, ponieważ analiza stabilności szeregu czasowego zazwyczaj wymaga zazwyczaj odrzucenia danych znacznie różniących się od pozostałych elementów szeregu [239]. Czy sztuczna sieć neuronowa znajdzie zastosowanie w tej sytuacji? Baxter oraz King stoją na stanowisku, że niewłaściwe jest używanie ich do rozwiązywania problemów, w których istnieje rozwiązanie analityczne [140], [141]. Szeregi czasowe są wykorzystywane przez operatory predykcji z użyciem sztucznych sieci neuronowych. Wśród topologii SSN wykorzystywanych w predykcji szeregów czasowych dominują jednak sieci jednokierunkowe [101]. W dużym uproszczeniu, zasadniczy proces prognozowania przy użyciu SSN jest realizowany zgodnie ze schematem przedstawionym na rysunku 1.33. Studia literaturowe [21]-[299] na temat prognozowania ekonomicznego koncentrują się na zastosowaniu prognozowania do:

- zachowania rynku akcji,
- bankructw,
- zmian trendu rynkowego,

- kursów walutowych,
- ryzyka kredytowego,
- wyników finansowych,
- wielkości i wartości sprzedaży.

1.4.5 Analiza predykcji z wykorzystaniem sztucznych sieci neuronowych

Do prawidłowego działania algorytmu predykcyjnego opartego na SSN potrzebne jest określenie zbioru danych oraz danych kontrolnych będącą podstawą sprawdzenia dokładności predykcji wykorzystującej konkretny algorytm predykcyjny [101], [334], [195]. W sieciach neuronowych realizowanie są następujące obliczenia:

1. Selekcja oraz przygotowanie danych do przetwarzania przez sieć neuronową, w pełni charakteryzujące rozwiązywane zagadnienie.
2. Dobór właściwej topologii.
3. Wybór właściwego procesu uczenia oraz filtracja danych uczących.
4. Weryfikacja otrzymanych wyników.

Należy podkreślić, że są one równie ważnymi składnikami w całym procesie uczenia się sieci neuronowych, a tym samym w procesie wykorzystywania sieci neuronowych do prognozowania.

Powyższy algorytm ma charakter iteracyjny, a obliczenia prowadzone są aż do momentu osiągnięcia zadowalającej wartości funkcji błędu (ewentualnie przekroczenia założonego czasu trwania obliczeń).

Przygotowanie danych do przetwarzania przez sieć neuronową,

Określenie właściwych zmiennych wejściowych dla zaproponowanej architektury sieci jest elementem dość skomplikowanym, ściśle zależy od wybranego programu do symulacji nauki sieci. Jedną z proponowanych struktur sieci jest struktura prezentowana poniżej:

- dane uporządkowane są w wierszach;
- pierwsze dziesięć to dane wejściowe (IN), ostatnia jedenasta jest daną wyjściową (OUT);
- liczba wierszy zależy od liczby wektorów wejściowych dobranych według miary Vapnika-Chervonenkisa.

Architektura sieci, Kolejnym z elementów w konstrukcji sieci neuronowej jest odpowied-

nio dobrana liczba neuronów w warstwie ukrytej. Liczba ta pociąga za sobą odpowiednią liczbę wzorców podawanych sieci (liczba wektorów wejściowych). W celu dobrania odpowiedniej konfiguracji wykorzystano miarę Vapnika-Chervonenkisa ($VCdim$) postaci

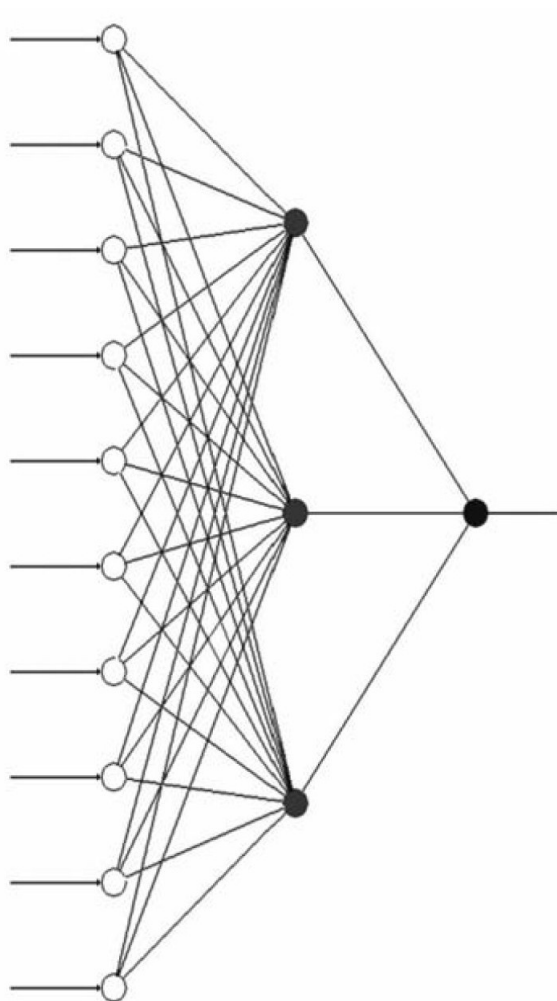
$$2\left[\frac{K}{2}, \right]N \leq VCdim \leq 2N_w(1 + \log N_n) \quad (1.67)$$

gdzie:

- $\lceil \cdot \rceil$ – część całkowita liczby,
- N – wymiar wektora wejściowego,
- K – liczba neuronów w warstwie ukrytej, , Miara ta pozwala jedynie na oszacowanie początkowej liczby wektorów, więc w praktyce wykonywanych jest wiele testów, a następnie dobierana jest konfiguracja sieci, która daje najlepsze wyniki. W ten sposób dobierana jest struktura sieci, liczba neuronów w warstwie ukrytej winna być jak najmniejsza, co jest kontrolowane przez funkcję błędu procesu uczenia. Właściwy dobór omawianych parametrów ma duży wpływ na szybkość generalizacji sieci. Przeprowadzenie procesu minimalizacji funkcji błędu w odniesieniu do zbyt małej, w stosunku do liczby wag sieci, liczbie danych uczących powoduje dowolność wartości wielu wag, co przy zmianie punktów testujących względem uczących jest przyczyną znacznego odchylenia aktualnej wartości od spodziewanej. Przy tej samej liczbie próbek trenujących dalsze zmniejszenie liczby neuronów w warstwie ukrytej pozwala jednocześnie uzyskać mniejsze błędy uczenia się i dobre możliwości generalizacji.
- N_w – całkowita liczba wag w sieci,
- N_n – całkowita liczba neuronów w sieci.

cowanie początkowej liczby wektorów, więc w praktyce wykonywanych jest wiele testów, a następnie dobierana jest konfiguracja sieci, która daje najlepsze wyniki. W ten sposób dobierana jest struktura sieci, liczba neuronów w warstwie ukrytej winna być jak najmniejsza, co jest kontrolowane przez funkcję błędu procesu uczenia. Właściwy dobór omawianych parametrów ma duży wpływ na szybkość generalizacji sieci. Przeprowadzenie procesu minimalizacji funkcji błędu w odniesieniu do zbyt małej, w stosunku do liczby wag sieci, liczbie danych uczących powoduje dowolność wartości wielu wag, co przy zmianie punktów testujących względem uczących jest przyczyną znacznego odchylenia aktualnej wartości od spodziewanej. Przy tej samej liczbie próbek trenujących dalsze zmniejszenie liczby neuronów w warstwie ukrytej pozwala jednocześnie uzyskać mniejsze błędy uczenia się i dobre możliwości generalizacji.

Należy podkreślić, że dalsze zmniejszanie liczby ukrytych neuronów powoduje, że sieć nie jest w stanie zmapować danych uczących. Sieć nie może poprawnie odwzorować danych uczących, ponieważ stopień swobody tej sieci jest zbyt mały w stosunku do wymagań stawianych na proces odwzorowania.



Rysunek 1.34: Przykładowa architektura sieci

p

Źródło: opracowanie własne.

Wśród parametrów przykładowej sieci neuronowej należy wymienić (rys. 1.34):

- warstwa wejściowa: 10 neuronów,
- warstwa ukryta: 3 neurony,
- warstwa wyjściowa: 1 neuron,
- liczba wektorów wejściowych: 50.

Proces uczenia, Kolejnym, po wyznaczeniu właściwej dla danego celu topologii sieci, etapem jest dobór odpowiedniej procedury uczącej. Jednym z najbardziej powszechnych metod uczenia SSN jest algorytm wstecznej propagacji błędów, który sprowadza się do realizacji następujących kroków:

1. Przeprowadź analizę struktury sieci, kierunku przepływu sygnałów. Przeprowadź symulację propagacji sygnału pomiędzy warstwami.
2. Generuj sieć propagacji wstecznej. Dla tak skonstruowanej sieci oblicz wartości odpowiednich różnic wstecznych, tzn. zdefiniuj sieć propagacji wstecznej poprzez zainicjowanie procesu zmiany kierunku przepływu sygnału, od warstwy wyjściowej do warstwy wejściowej oraz zmianę funkcji aktywacji na jej pochodne.
3. Przeprowadź proces uczenia na zbiorze wyników otrzymanych w etapach poprzednich, ale w sieci zwykłej, czyli o normalnym przepływie, oraz w sieci wstecznej propagacji o przepływie odwróconym.
4. Powtórz powyższe etapy dla całego zbioru danych aż do chwili spełnienia kryterium algorytmu, tzn. gdy spełniony jest warunek - norma gradientu będzie mniejsza od zadanej wielkości dokładności procesu uczenia.

W praktyce funkcje aktywacji mogą przyjmować formy funkcji sigmoidalnych.

Weryfikacja otrzymanych wyników, Do oszacowania zasadności wykorzystania SSN do prognozowania krótkich szeregów czasowych wykorzystywano:

- porównanie błędów predykcji prognozowania z wykorzystaniem sztucznych sieci neuronowych z błędami tego samego rodzaju, ale uzyskanymi w wyniku prognozowania szeregu czasowego za pomocą następujących metod:
 - logarytmicznego modelu tendencji rozwojowej,
 - średnich ruchomych prostych,
 - średnich ruchomych ważonych, wykorzystujących wagi liniowe,
 - średnich ruchomych ważonych, wykorzystujących wagi harmoniczne,
 - wyrównania wykładniczego,
 - wag harmonicznych;
- przedstawienie rankingu ww. metod uwzględniających zarówno dokładność predykcji, jak również szybkość obliczeń, skomplikowanie wybranej metody, zakres koniecznej wiedzy badacza oraz konieczność wykorzystania określonego warsztatu badawczego (maszyny liczące, programy komputerowe itp.).

Przy realizacji pierwszego etapu weryfikacji algorytmu predykcyjnego należy wyliczyć błędy predykcji:

- średni absolutny błąd MAE

$$MAE = \frac{1}{T}, \sum_{t=1}^T |y_t - \hat{y}_t| \quad (1.68)$$

- średni absolutny błąd procentowy $MAPE$

$$MAPE = \frac{1}{T}, \sum_{t=1}^T \left| \frac{y_t - \hat{y}_t}{y_t} \right| \cdot 100\% \quad (1.69)$$

- procentowy błąd średniokwadratowy $MSPE$

$$MSPE = \frac{1}{T}, \sum_{t=1}^T \left(\frac{y_t - \hat{y}_t}{y_t} \right)^2 \cdot 100\% \quad (1.70)$$

gdzie:

y_t – dane rzeczywiste (empiryczne) szeregu czasowego,

\hat{y}_t – dane teoretyczne szeregu czasowego, ,

T – długość szeregu czasowego.

1.4.6 Głębokie sieci neuronowe

Głębokie sieci neuronowe będąc konsekwencją naturalnej ewolucji jakiej poddawane są sztuczne sieci neuronowe, są rozwinięciem tej koncepcji, próbują rozwiązywać problemy o hierarchicznej strukturze, inspirowane zasadą działania mózgu. Gwałtowny postęp w dziedzinie algorytmów głębokiego uczenia doprowadził do nowych prób wykorzystania sieci neuronowych w wielu praktycznych okolicznościach, w tym w przetwarzaniu obrazu i dźwięku. Dzięki nim dokonał się przełom w dziedzinie rozpoznawania mowy. Osiągnięte obiecujące wyniki zastosowania głębokich sieci neuronowych w tym obszarze przyczyniło się do wzrostu zainteresowania badaczy z innych obszarów nauki, w tym ekonomii oraz ekonometrii. Głównym celem pracy jest dowiedzenie jednej z tez pracy mówiącej o wykorzystaniu głębokich sieci neuronowych w systemach predykcji szeregów czasowych cen mleka w proszku. Algorytmy głębokich sieci neuronowych są technikami powszechnie stosowanymi w rozwiązywaniu wielu współczesnych problemów. Nie istnieje jednak jakakolwiek wspólna forma topologii sieci, która bez wymaganych modyfikacji mogłaby być użyta jako uniwersalne narzędzie inżynierskie. Z tego powodu, na potrzeby niniejszej pracy przeanalizowane zostaną konkurencyjne rozwiązania, co umożliwi wybranie najbardziej odpowiedniego z nich dla potrzeb predykcji cen mleka w proszku.

1.4.7 Architektury głębokich sieci neuronowych

Przedstawiona powyżej MLP nie jest jedyną architekturą głębokiej sieci neuronowej. W literaturze bowiem można odnaleźć wiele alternatywnych rozwiązań, które to zostaną przedstawione w dalszej części niniejszej pracy.

1.4.7.1 Autokodery

Autokoder to sieć neuronowa, która jest uczona tak, by na wyjściu przedstawić wartości podane na wejście. Jej wewnętrzna struktura obejmuje ukrytą warstwę h , która opisuje kod reprezentujący wartości wejściowe. W skład sieci wchodzi dwie części kodera (3.28a) oraz dekodera (3.28b), który tworzy i rekonstruuje sygnał y . Architektura tego typu jest przedstawiona na rysunku, 1.35.

$$h = f(W_e x + b_e), \quad (3.28a)$$

$$y = f(W_d x + b_d) \quad (3.28b)$$

Gdyby jednak autokodery miały nauczyć się zależności $y = x$, nie byłyby one specjalnie użyteczne. Z tego powodu są one projektowane tak, aby nie było możliwe powielenie wartości wejściowej w sensie dosłownym, tym samym możliwe było pominięcie części informacji. W celu jak najlepszego odwzorowania wartości wejściowych mimo tych ograniczeń, skłonione są one do znalezienia najistotniejszych informacji w wektorze wejściowym i tę cechę autokoderów wykorzystuje się w praktyce. Jeden z bardziej efektywnych, a zarazem prostych sposobów ograniczenia pojemności autokodera, jest zrealizowany w algorytmie *Denoising Autoencoder (DAE)*. Można go interpretować jako nałożenie losowej maski binarnej, opisanej już przy okazji definicji techniki dropout (1.66), jednak jest ona nakładana na strukturę neuronów warstwy wejściowej. Część kodującą można zatem przedstawić jako:

$$r \sim \text{Bernoulli}(p), \quad (3.29a)$$

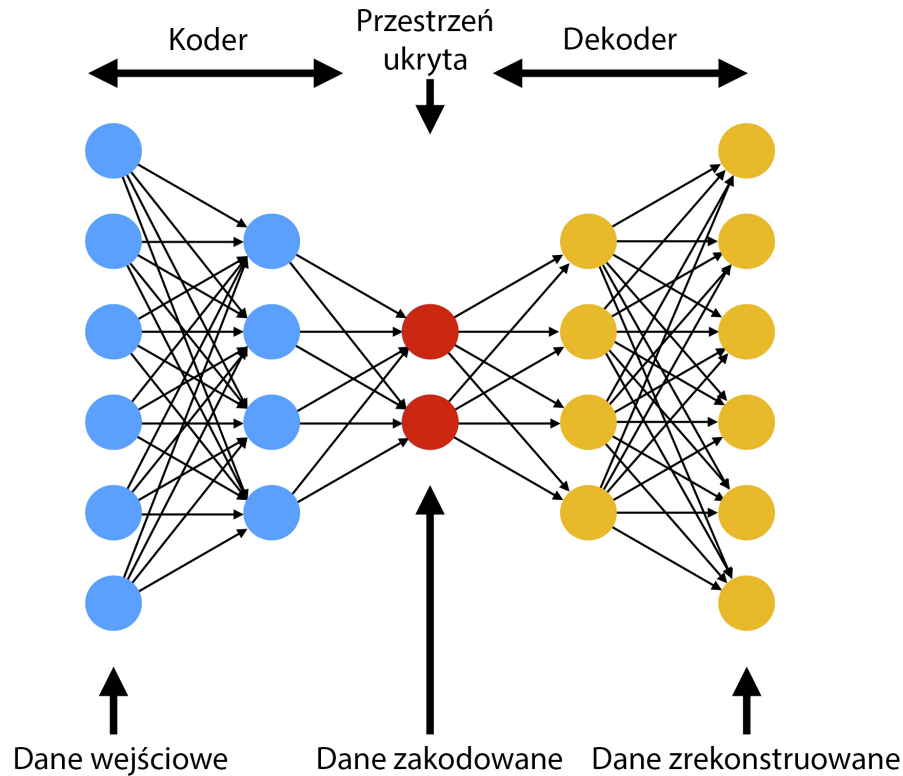
$$x' r_n * x, \quad (3.29b)$$

$$h = f(w x' + b) \quad (3.29c)$$

Ten wariant autokodera zmuszony jest do odtworzenia prawdziwej wartości wejściowej, mimo jej uszkodzenia w fazie kodowania. Staje się przez to odporny na szum i zniekształcenia. Algorytm ten łączy się często z regularyzacją dropout, co prowadzi do większej generalizacji cech wytypowanych poprzez autoenkoder. Inną metodą pozwalającą zapobiec nauczeniu funkcji tożsamościowej jest skorelowanie wag autoenkodera:

$$W_d = W_e^T \quad (3.30)$$

Jednowarstwowe autoenkodery zazwyczaj używa się do wstępnego uczenia głębokich sieci neuronowych (rys. 1.32). Natomiast rozszerzając obydwie części, koder oraz dekodery, o dodatkowe warstwy, można uzyskać *Stacked Autoencoders*, głęboki autokoder wykorzystywany do modelowania złożonych zależności danych.

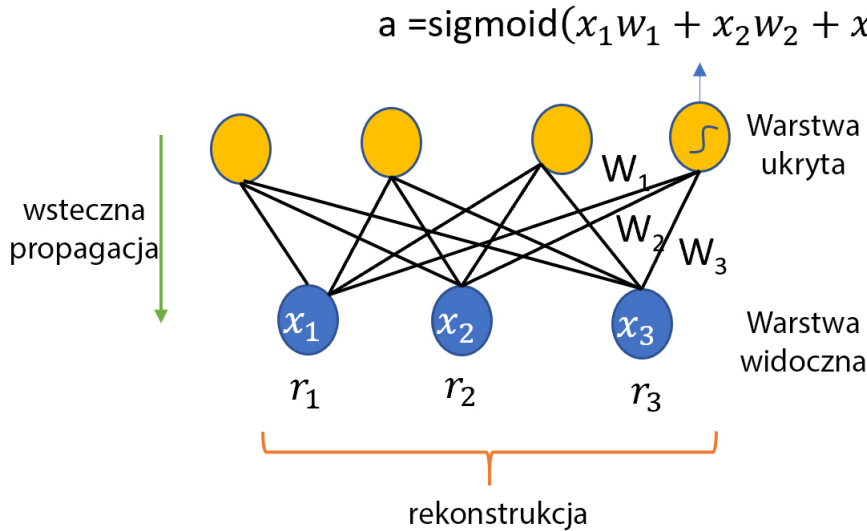


Rysunek 1.35: Architektura autokodera

1.4.7.2 Ograniczone Maszyny Boltzmann

Ograniczona Maszyna Boltzmann (ang. *Restricted Boltzmann Machine, RBM*) to stochastyczna, generatywna SSN składająca się z warstwy widocznej oraz warstwy ukrytej (rys. 1.36). Względem zwykłej Maszyny Boltzmann różni się ona ograniczeniem nałożonym na połączenia, tzn. model nie zawiera połączeń pomiędzy neuronami tej samej warstwy, co pozwala na bardziej wydajny proces uczenia. Generatywność charakteryzuje autonomiczny system, którego użytkownicy zdobywają niezależną zdolność generowania, kreowania oraz tworzenia. W ten sposób przygotowany algorytm jest zdolny do generowania próbek bazując na nauczonej rozkładzie danych. Pozyskanie wiedzy na temat owego rozkładu jest realizowane poprzez zwiększanie częstości pojawiania się przykładów z wektora trenującego, tj. tzw. próbek pozytywnych oraz tym samym prowadząc do zmniejszenia prawdopodobieństwa pojawiania się przykładów generowanych przez model, tj. takich, które nie występują w wektorach uczących, czyli próbek negatywnych. Model RBM jest również tzw. modelem opartym na energii (ang. *Energy-Based Models*). Działa-

nie takich modeli sprowadza się do przypisania wartości skalarnej energii w trakcie każdej zmiany zestawu danych. Ta wartość skalarna jest przypisywana do konfiguracji w taki sposób by osiągnąć niską wartość, gdy prawdopodobieństwo jej wygenerowania przez model jest wysokie. Proces uczenia sprowadza się zatem do dopasowania funkcji energetycznej modelu w taki sposób, by dane wejściowe ze zbioru uczącego posiadały niską energię.



Rysunek 1.36: Struktura modelu ograniczonej maszyny Boltzmanna

W modelu energetycznym prawdopodobieństwo wygenerowania próbki przez model $p(x)$ można przedstawić jako (3.31), co gwarantuje niezerowe prawdopodobieństwo dla dowolnego przykładu x . Czynnikiem normalizujący Z jest określany mianem sumy statystycznej, a $E(x)$ i jest wartością energii związanej z daną próbką.

$$p(x) = \frac{e^{-E(x)}}{Z},$$

$$Z = \sum_x e^{-E(x)}, \quad (3.31)$$

Jednak gdy modele posiadają nieobserwowalne zmienne (ang. *hidden*, h), tak jak warstwa ukryta w RBM, prawdopodobieństwo (3.31) należy wyliczyć sumę dla każdej możliwej konfiguracji wartości ukrytych:

$$p(x) = \sum_h P(x, h) = \sum_h \frac{e^{-E(x, h)}}{Z}, \quad (3.32)$$

W takim wypadku miarę dopasowania modelu do rozkładu danych przejmuje funkcja swobodnej energii (1.33), czyli energii liczonej tylko na podstawie zmiennych widocznych. Pozwala to zapisać prawdopodobieństwo znów w formie (3.34)

$$\mathcal{F}(x) = -\log \sum_h e^{-E(x, h)} \quad (1.33)$$

$$p(x) = \frac{e^{-\mathcal{F}(x)}}{Z},$$

$$Z = \sum_x e^{-\mathcal{F}(x)}, \quad (3.34)$$

Energię *Ograniczonej Maszyny Boltzmannna* determinuje zależność (1.35), w której parametry W, b, c to kolejno: macierz wag połączeń pomiędzy warstwami, wektor biasu neuronów ukrytych, wektor biasu neuronów widocznych.

$$E(v, h) = -c^T x - b^T h - h^T W x \quad (1.35)$$

Dzięki ograniczeniu wpisanemu w model RBM, widoczne oraz ukryte neurony są wzajemnie warunkowo niezależne. Korzystając z tej własności można zapisać:

$$P(h|x) = \prod_i P(h_i|x), \quad (3.36a)$$

$$P(h|x) = \prod_j P(x_j|h) \quad (3.36b)$$

Podstawowy model RBM jest zbudowany z neuronów binarnych i dlatego nazywany jest często *Bernoulli-Bernoulli RBM*. W tym przypadku prawdopodobieństwa aktywacji neuronów widocznych i ukrytych można zapisać w prostej formie funkcji sigmoidalnych:

$$P(h_i = 1|x) = \sigma(W_i x + b_i), \quad (3.37a)$$

$$P(x_j = 1|h) = \sigma(W_j^T h + c_j) \quad (3.37b)$$

Prawdopodobieństwo aktywacji neuronów ukrytych w zależności od aktywacji neuronów widocznych określa się mianem *propagacji w górę* (3.37a), w odróżnieniu od *propagacji w dół* (3.37b).

Warianty RBM, Binarne wartości neuronów w modelu *Bernoulli-Bernoulli*

RBM prowadzą do zmniejszenia możliwości modelowania wartości rzeczywistych, które przeważają w rzeczywistych problemach. W tych przypadkach rozwiązaniem jest wariant *Gaussian-Bernoulli RBM*, w którym to neurony warstwy widocznej generują wartości rzeczywiste, a propagacja w dół w takim modelu korzysta z rozkładu Gaussa:

$$E(v, h) = -c^T x - b^T h - h^T W x \quad (1.38)$$

Powyższa zależność wynika z dodania komponentu $\frac{1}{2}x^T x$ do funkcji energetycznej:

$$E(v, h) = -c^T x - b^T h - h^T W x + \frac{1}{2}x^T x \quad (1.39)$$

Równania (1.38) i (1.39) mają bardziej skomplikowaną formę, gdy rozkład danych uczących nie jest standaryzowany.

Kolejnym, powszechnie stosowanym usprawnieniem modelu jest zamiana neuronów binarnych w warstwie ukrytej na neurony o aktywacji *NoisyReLU*, w takim przypadku propagacja w dół, czyli generowanie próbki ma postać:

$$h_j \sim \max(0, W_{ij}x + b_i + \mathcal{N}(0, 1)) \quad (1.40)$$

Uczenie RBM,

Pośrednim celem uczenia Ograniczonych Maszyn Boltzmanna jest zmaksymalizowanie prawdopodobieństwa wygenerowania przykładów ze zbioru uczącego. Funkcja kosztu, która spełnia te założenia jest ujemne log-prawdopodobieństwo:

$$\mathcal{L}(x) = -\log(p(x)) = \mathcal{F}(x) - \log\left(\sum_{\tilde{x}} p(\tilde{x})\right), \quad (3.41a)$$

$$\frac{\partial \mathcal{L}(x)}{\partial \Theta} = \frac{\partial \mathcal{F}(x)}{\partial \Theta} - \sum_{\tilde{x}} p(\tilde{x}) \frac{\partial \mathcal{F}(\tilde{x})}{\partial \Theta}, \quad (3.41b)$$

Pierwszy komponent gradientu funkcji kosztu względem parametrów Θ (3.41b) nazywany jest fazą pozytywną, ponieważ maksymalizuje prawdopodobieństwo wygenerowania przedstawionego przykładu x . Natomiast drugi element jest fazą negatywną, ponieważ minimalizuje prawdopodobieństwo pozostałych przykładów generowanych przez model i jest oznaczony jako \tilde{x} . Wyznaczenie pełnej sumy w fazie negatywnej jest niepraktyczne, jednak jako że jest to wartość oczekiwana wyrażenia $\frac{\partial \mathcal{F}(\tilde{x})}{\partial \Theta}$ to aproksymuje się ją podzbiorem \mathcal{M} :

$$\frac{\partial \mathcal{L}(x)}{\partial \Theta} \approx \frac{\partial \mathcal{F}(x)}{\partial \Theta} - \frac{1}{|\mathcal{M}|} \sum_{\tilde{x} \in \mathcal{M}} \frac{\partial \mathcal{F}(\tilde{x})}{\partial \Theta} \quad (1.42)$$

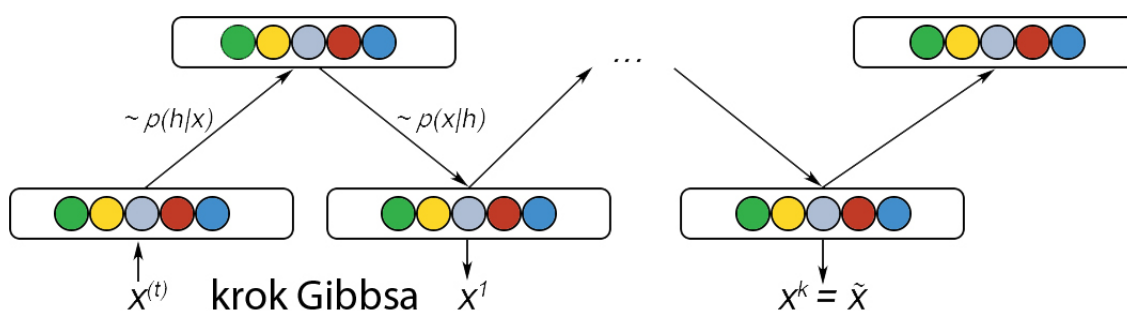
Próbki $\tilde{x} \in \mathcal{M}$, są określane jako *próbki negatywne* i są one generowane zgodnie z rozkładem wyznaczonym przez aktualny stan modelu.

Ostatnim brakującym elementem w procesie uczenia modelu RBM jest zdefiniowanie rozkładu związanego z aktualnym stanem modelu, może być to wykonane przez zastosowanie łańcucha Markowa oraz próbkowania Gibbsa [153]. Biorąc pod uwagę fakt, że obie warstwy są warunkowo niezależne, można wykonywać następujące kroki (rys. 1.37):

$$h^{(n+1)} \sim p(h^{(n+1)} | x^{(n)}), \quad (3.43a)$$

$$x^{(n+1)} \sim p(x^{(n+1)} | h^{(n+1)}) \quad (3.43b)$$

Gdzie $x^{(0)}$ jest wektorem losowym, a x^n dąży do $p(x)$ przy $n \rightarrow \infty$.



Rysunek 1.37: Zastosowanie łańcucha Markowa w algorytmie *Contrastive Divergence*

Osiągnięcie kryterium zbieżności łańcucha Markowa jest jednak czasochłonne, dlatego opublikowany w [153] algorytm *Contrastive Divergence (CD)* przyspiesza cały proces poprzez:

1. Zainicjowanie łańcucha Markowa aktualnym przykładem uczącym, tj. $x^{(0)} = x$, który wraz z postęпами w nauce staje się coraz bliższy rozkładowi $p(x)$.
2. Wykonywanie tylko k kroków Gibbsa, bez konieczności oczekiwania na osiągnięcia kryterium zbieżności. W rzeczywistości jednak już nawet $k = 1$ prowadzi do osiągnięcia dobrych rezultatów.

Modyfikacją algorytmu CD jest *Persistent Contrastive Divergence*, który korzysta z łańcucha Markowa posiadającego wewnętrzny stan aktualnej aproksymacji $p(x)$, dzięki czemu z każdym krokiem SGD ta aproksymacja jest dokładniejsza [307].

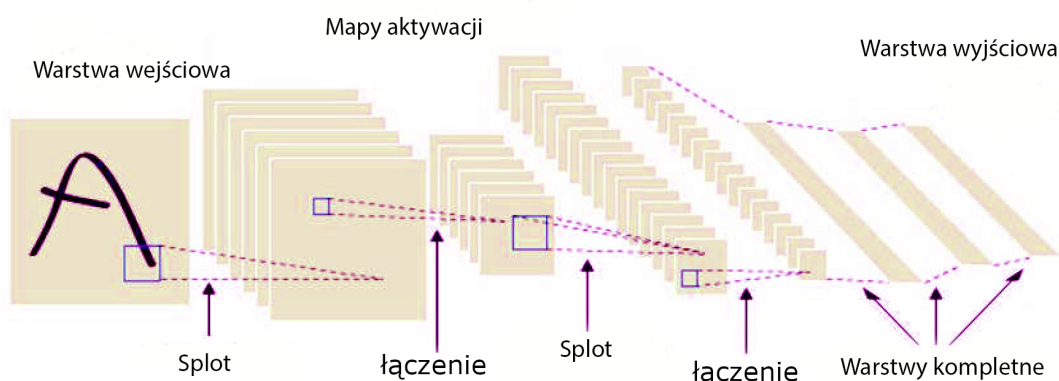
Zastosowanie, W praktyce Ograniczone Maszyny Boltzmanna najczęściej wykorzystane są do pretrenowania sieci wielowarstwowych (rys. 1.32), jednak mogą też tworzyć głębokie generatywne modele: *Deep Belief Network* oraz *Deep Boltzmann Machine*. Pierwszy z nich składa się z modelu RBM rozszerzonego o dodatkowe, zazwyczaj sigmoidalne, warstwy sztucznej sieci neuronowej. Natomiast drugi stanowi rozszerzony model RBM o wielu warstwach połączonych ze sobą w dwóch kierunkach.

1.4.7.3 Splotowe sieci neuronowe

Sieci splotowe stanowią odpowiedź na problem skalowania wektorów wejściowych w standardowych sieciach typu MLP. Problem ten jest szczególnie widoczny dla dziedziny przetwarzania obrazów. Przetwarzając nawet mały czarno-biały obraz o wymiarach 255x255, każdy neuron warstwy ukrytej musiałby posiadać 65 tysięcy wag. Jest to w

praktyce kłopotliwe. W pierwszej kolejności należy podłączyć wówczas neurony wyłącznie lokalnie, tj. do wyodrębnionego wycinka obrazu. Idea ta jest zaczerpnięta z zasady działania ludzkiego systemu wzrokowego, w którym to neurony aktywują się tylko, gdy zaistnieje zdarzenie w określonym fragmencie pola widzenia.

Następnym etapem jest zauważenie, że cechy, które są reprezentowane przez lokalne neurony mogą być efektywne również w całej strukturze obrazu. Wykorzystując ten fakt buduje się grupy neuronów, o wspólnych wagach, różniących się między sobą miejscem podłączenia do wejściowego obrazu (rys. 1.38). Wektor sygnałów wyjściowych z neuronów o tych samych wagach, można interpretować jako mapę aktywności danej cechy w całym obrazie. Cecha charakteryzowana przez jeden dzielony zestaw wag jest nazywana filtrem.



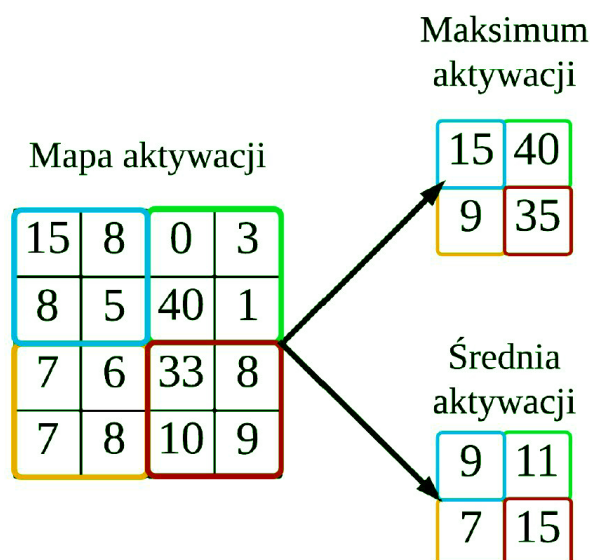
Rysunek 1.38: Architektura splotowej sieci neuronowej rozpoznającej litery [119]

Kompletna topologia (rys. 1.38), składa się z następujących warstw:

Warstwa splotowa, czy też *konwolucyjna*, jest zestawem opisanych powyżej map filtrów [272]. Z uwagi na to, że każda mapa jest dwuwymiarowa, na podobieństwo obrazów wejściowych, to warstwę splotową często interpretuje się jako trójwymiarową. Nazwa tej warstwy pochodzi od zamiany operacji mnożenia wektorów wykonywanej w klasycznym neuronie (1.38), na operację splotu (1.44), która lepiej sprawdza się przy wykrywaniu wzorców.

$$y_j = f(W_j * x + b_j) \quad (1.44)$$

Warstwa łącząca (ang. *pooling layer*) ma za zadanie zmniejszyć wymiar sieci przez ekstrakcję prostych statystyk aktywacji z lokalnego zakresu mapy neuronów. Najpowszechniej stosowane statystyki to maksimum aktywacji (max pooling) lub średnia aktywacja (ang. *mean pooling*) (rys. 1.39).



Rysunek 1.39: Warstwy łączące redukują wymiar danych poprzez ekstrakcję prostych statystyk z map aktywacji

Warstwa pełna jest stosowana w trakcie użycia końcowych faz przetwarzania. Wiąże się to z koniecznością znalezienia cech pozwalających na poprawne wnioskowanie. Uczenie sieci spłotowej nie różni się od nadzorowanego uczenia sieci MLP, choć spotykane są rozwiązania, w których najpierw wykonuje się wstępne uczenie filtrów warstwy spłotowej.

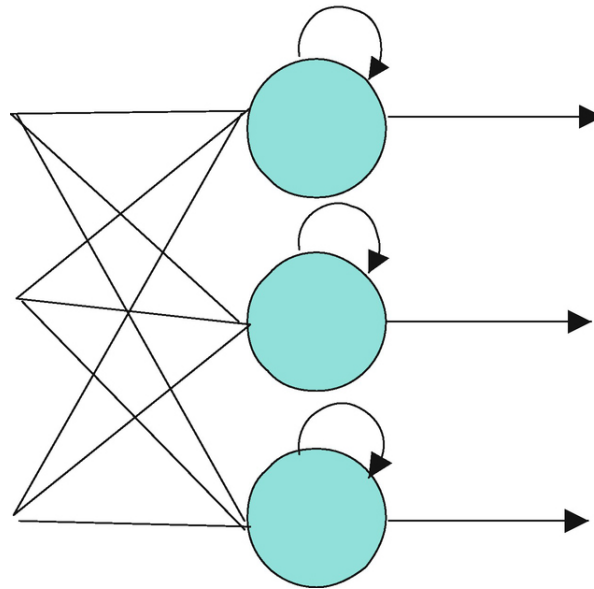
Pomimo iż ta topologia sieci wywodzi się z dziedziny przetwarzania obrazów, to znalazła również swoje zastosowanie w przetwarzaniu języka naturalnego, przetwarzaniu dźwięku oraz analizie szeregów czasowych cen [147].

1.4.7.4 Rekurencyjne sieci neuronowe

W najprostszej topologii (rys. 1.40) sieci rekurencyjne różnią się tylko sprzężeniem zwrotnym, czyli pętlą łączącą wyjście danego neuronu z jego wejściem. Pętla tego typu tworzy w modelu stan wewnętrzny, który pozwala przetwarzać i modelować ciąg wejść, przez co sieci rekurencyjne stosowane są do zadań o charakterze sekwencyjnym.

Obecność pętli w grafie sprawia również, że głębokość sieci staje się nieskończona. Z uwagi na fakt, iż wraz z głębokością rośnie pojemność sieci, to ma ona ogromne możliwości nauki, ale też sprawia duże problemy podczas uczenia. Ilość danych potrzebnych do nauczenia sieci rekurencyjnej sprawia, że są to ciągle modele przyszłości, ale ich potencjał jest przyczyną intensywnego rozwoju tej tematyki i już dzisiaj widać kilka obiecujących rezultatów. Jednym z ciekawszych rozwiązań jest wykorzystanie wielu warstw jednego z wariantów sieci rekurencyjnych: modelu *Long Short Term Memory (LSTM)* przez zespół naukowy Google. Zmniejszył on stopę błędu ich autorskiego systemu rozpoznawania mowy aż o połowę. Cała sieć została nauczona wykorzystując algorytm *Connectionist*

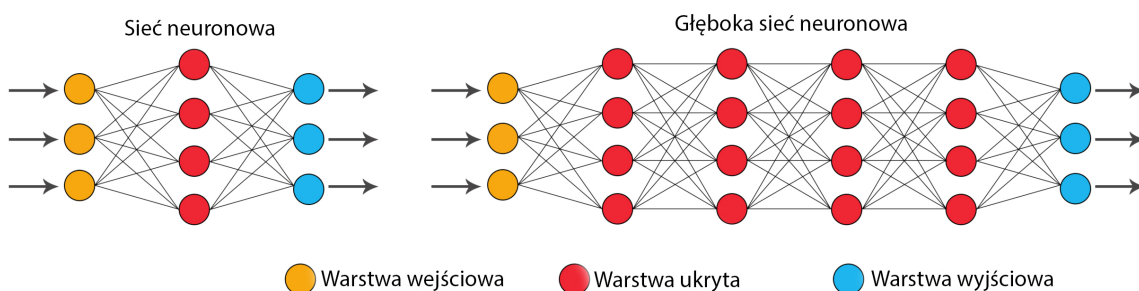
Temporal Classification (CTC) [235].



Rysunek 1.40: Sieci rekurencyjne różnią się pętlami łączącymi wyjście danego neuronu z jego wejściem

1.4.8 Głębokie uczenie

Głębokie uczenie, jako nowa gałąź sztucznej inteligencji, jest w istocie rzeczą metodą uczenia maszynowego wykorzystującą wielowarstwowe transformacje nieliniowe do modelowania abstrakcji wysokiego poziomu w zbiorach danych. Zgodnie z jego założeniami, kolejne poziomy stanowią poziomy funkcjonalne w porządku od najmniejszego do najbardziej abstrakcyjnego. Głębokie sieci neuronowe to najpopularniejsza grupa algorytmów głębokiego uczenia. rys. 1.41) [172].



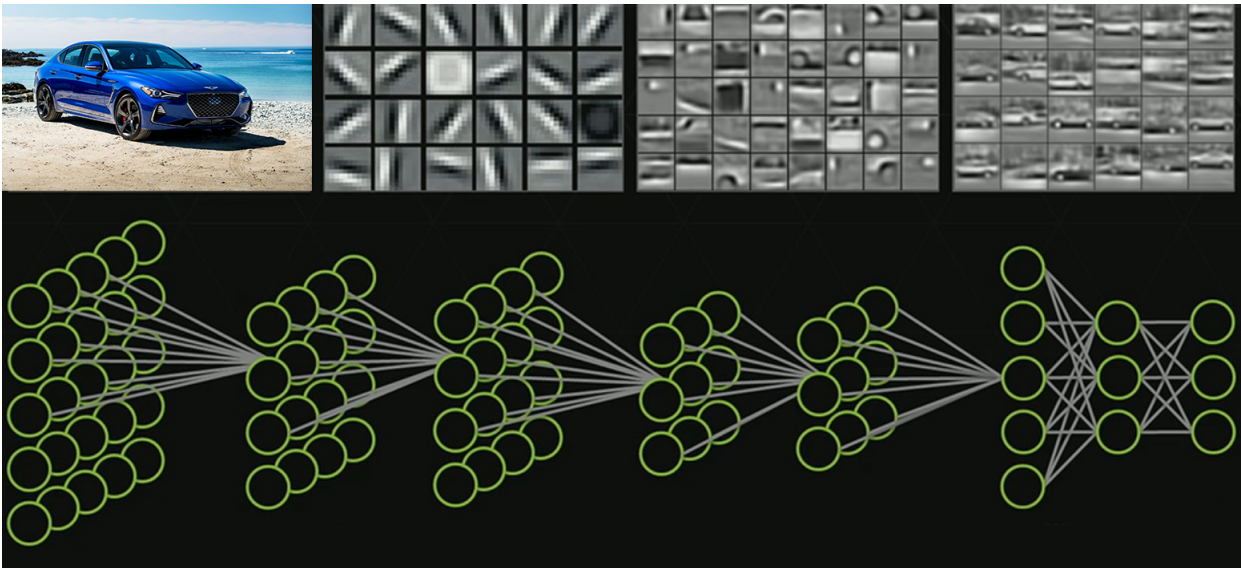
Rysunek 1.41: Głębokie sieci neuronowe są zbudowane z większej liczby warstw niż sieci klasyczne

Głębokość architektury sieci neuronowej jest definiowana jako długość najdłuższej ścieżki między neuronami wejściowymi i wyjściowymi. W topologii takiej sieci jest to bezpośrednio przenoszone na liczbę warstw. Formalnie, nie ma zatem ograniczeń co do

liczby warstw, które muszą być tworzyć taką sieć by uznać ją za sieć głęboką. Z tego powodu, można założyć, że sieć, w której skład wchodzi więcej niż dwie warstwy ukryte jest już siecią *głęboką*. Można oczekiwać, że wraz z rozwojem coraz większych sieci próg ten może się przesunąć [282], [173], [198], [298], [230]. Problemy, które są rozwiązywane przy pomocy metod uczenia maszynowego mają zwykle hierarchiczną strukturę danych:

- Istotą przetwarzania języka naturalnego jest hierarchiczna segmentacja, tj, podział na zdania, a te w dalszej kolejności na słowa.
- W przypadku rozpoznawania twarzy konieczna jest segmentacja obrazu, tj. ekstrakcja określonych cech takich, jak nos, usta, czy oczy, co w dalszej kolejności tworzy wektor deskryptorów, tj. możliwe jest opisanie ich jako zbiór prostych kształtów.
- W algorytmach przetwarzania mowy słowa dzieli się na sylaby, a te znowu na fonemy.

Hierarchie tego typu można scharakteryzować używając sieci jednowarstwowej, co jest dowiedzione poprzez tzw. *twierdzenie o uniwersalnej aproksymacji*, jednak w praktyce okazuje się, że płaska reprezentacja nie jest ani zwięzła, ani łatwa do nauczenia [231], [289], [288]. Koncepcja głębokich sieci neuronowych, która została zainspirowana działaniem mózgu również podlega potrzebie nauczenia swoich struktur danych, co w dalszej kolejności, znacznie ułatwia wnioskowanie. Model głębokich sieci zakłada istnienie większej liczby warstw ukrytych niż ma to miejsce w przypadku tzw. płytkie sieci neuronowych [310], [175]. Kolejne warstwy tak rozbudowanej topologii są w stanie rozpoznawać coraz bardziej złożone obiekty, odnosząc się do do danych abstrakcyjnych wyłonionych przez poprzednie warstwy (rys. 1.42). Nie znaczy to jednak, że dla wszystkich problemów sieci głębokie będą najlepsze. Okazuje się, że dla mniej skomplikowanych problemów lub problemów bez hierarchicznej struktury, sieci płytkie mogą dawać lepsze rezultaty.



Rysunek 1.42: Ekstrakcja i przetwarzanie cech sygnału dwuwymiarowego przez głęboką sieć neuronową

Należy zauważyć, że silnym atutem głębokich sieci neuronowych jest także automatyczna ekstrakcja cech sygnałów wielowymiarowych. W konwencjonalnych metodach uczenia maszynowego konieczne jest znalezienie metody wydobywającej najbardziej informatywne oraz nienadmiarowe cechy przetwarzanych danych i dopiero bazując na tych informacjach możliwe jest uczenie sieci. Głębokie sieci neuronowe potrafią się samodzielnie nauczyć jak wybierać najistotniejsze cechy danych bez udziału "operatora". Koncepcja głębokich sieci neuronowych powstała już na początku lat 70-tych, jednak są przyczyny, dla których ich rozwój w początkowej fazie był spowolniony:

- wraz ze wzrostem liczby warstw rośnie liczba parametrów modelu, które należy nauczyć. Ucząc wiele warstw przy użyciu małych wektorów danych dochodzi do tzw. przeuczenia sieci. W skrajnych okolicznościach SSN o zbyt dużej pojemności jest w stanie nauczyć się całego wektora uczącego „na pamięć”;
- zwiększając rozmiar sieci neuronowych oraz ilość danych uczących, zwiększa się także znacznie złożoność obliczeniowa i pamięciowa modelu, co w prostej linii prowadzi do zmożonego zapotrzebowania na moc obliczeniową maszyn, które przetwarzają tak duże zbiory danych;
- klasyczne podejście uczenie sieci, tj. przy użyciu metod gradientowych oraz algorytmu propagacji wstecznej, boryka się z problemem tzw. *zanikającego gradientu*, opisany w dalszej części pracy.

Podczas ostatniej dekady problemy te jednak tracą na znaczeniu ponieważ:

- rozwój dużych baz danych, dostęp do nich za sprawą szerokopasmowego Internetu umożliwiły bezpośredni dostęp do danych rozmaitego typu. W niektórych przypadkach problem się jednak odwrócił, tj. trudność stanowi eksploracja wielkich struktur danych, czyli tzw. *zglębianie danych (data mining)*, co stanowi osobną dziedzinę nauk inżynierskich. Nadal istnieją jednak specjalistyczne dziedziny, w których ciągle brakuje wystarczających ich ilości, aby stosować głębokie metody nauki;
- moc obliczeniowa procesorów oraz zasoby pamięci operacyjnej nieprzerwanie rośnie od kilku dekad, a wprowadzenie na rynek wydajnych procesorów graficznych (GPU) przyspieszyło procesy uczenia sieci neuronowych o rząd wielkości;
- prace Yanna LeCuna na temat *sieci splotowych* [200], [199] oraz pomysł Geoffreya Hintonna [180], [181] na *przetrenowanie* głębokich sieci neuronowych przy użyciu *Ograniczonych Maszyn Boltzmannna* można uznać za silny impuls rozwojowy dla tej dziedziny. Każdy rok przynosi nowe prace na ten temat, prowadząc tym samym do rozwoju metod uczenia adekwatnych dla większych, niż do tej pory, struktur.

1.5 Filtracja szeregów czasowych cen

Celem tej części pracy charakterystyka podstawowych filtrów cyfrowych. Jest to konieczne, bo modele tego typu znajdują zastosowanie w filtracji ekonomicznych szeregów czasowych. Przedstawione zostaną również skutki tych operacji cyfrowego przetwarzania sygnałów na portrety fazowe ich składowych trendowych oraz cyklicznych. W istocie rzeczy operacje arytmetyczne dokonywane na szeregach czasowych, czyli te związane są z ich analizą, prognozowaniem, modelowaniem, sterowaniem itp. można potraktować jako filtrację cyfrową. Wspomnieć należy, że zasadnicza większość operacji arytmetycznych w cyfrowym przetwarzaniu sygnałów sprowadza się do mnożenia i sumowania (mnożenia z akumulacją), co jest operacją celową optymalizowaną w ramach architektury tzw. procesorów sygnałowych. Filtry cyfrowe, poza elektroniką, ze szczególnym wskazaniem na przetwarzanie dźwięku i obrazu, znajdują zastosowanie m.in. w ekonomii do wygładzania szeregów czasowych, usuwania niepożądanych wahań (sezonowych, przypadkowych, wysoko- lub niskoczęstotliwościowych itp.), odsumiania sygnałów, prognozowania i modelowania procesów ekonomicznych [265], [174], [141], [334], [214], [179], [287], [237], [191], [276]. W dalszej części pracy zostanie przedstawiony zarys zasad i metodologia filtracji ze wskazaniem tych filtrów i technik cyfrowych, które znajdują szczególne zastosowanie w filtracji i odsumianiu szeregów czasowych. Należy zauważyć, że ekonomiczne szeregi czasowe są cyklami nieokresowymi, tzn. nie mają ściśle określonej długości oraz skali czasowej. Zastosowanie metodologii filtracji pozwala jednak wyłonić okresową, regularną naturę ekonomicznych szeregów czasowych. Jak dotąd, w tym obszarze badań za najbar-

dziej efektywne uznawano filtry Kalmana oraz Hodricka-Prescotta. W dalszej części pracy przedstawiona zostanie nowa metoda filtracji szeregów czasowych bazująca na koncepcji tzw. próbkowaniu oszczędnego, któremu to zostanie poświęcony odrębny podrozdział. Skupienie na tej metodzie jest uwarunkowane jej zastosowaniem przez autora niniejszej pracy w obszarze zaproponowanego nowego paradygmatu uczenia splotowych sieci neuronowych.

1.5.1 Wstęp do filtracji cyfrowej

W elektronice i innych naukach inżynierskich filtry cyfrowe odgrywają istotną rolę i to zarówno w postaci bezpośredniej jak i pośredniej. W praktyce mianowicie wiele powszechnie stosowanych operacji algebraicznych jest w istocie rzeczą realizacją filtrów cyfrowych w dziedzinie czasu, np. model ARMA jest literalnie realizacją filtru o nieskończonej odpowiedzi impulsowej w dziedzinie czasu. Operacje splotu, korelacji, autokorelacji też w istocie rzeczą można sprowadzić do takiej postaci, co jest uwarunkowane charakterystyką operacji arytmetycznych, które je opisują. Filtry cyfrowe są powszechnie stosowane w przetwarzaniu obrazu i dźwięku, np. w adaptacyjnych algorytmach redukcji szumu w telekomunikacji. Filtr cyfrowy F to operator transformacji (przetwarzania) danego (wejściowego sygnału) szeregu czasowego $\{y_t\}$ w inny (wyjściowy sygnał) szereg czasowy $\{z_t\}$. Transformacja taka określana jest mianem filtracji:

$$\{y_t\}, \rightarrow \boxed{F}, \rightarrow \{z_t\}. \quad (1.45)$$

W ogólności wszelkie przekształcenia szeregów czasowych związane z ich (szeregów) analizą, prognozowaniem, modelowaniem, sterowaniem itp. można przedstawić jako operację filtracji. Filtry cyfrowe są powszechnie stosowane m.in. w ekonomii do wygładzania szeregów czasowych, usuwania niepożądanych wahań (sezonowych, przypadkowych, wysoko- lub niskoczęstotliwościowych itp.), prognozowania i modelowania procesów ekonomicznych. Filtry można podzielić według rozmaitych kryteriów związanych z obszarem zastosowań, tj. celu (analiza i wygładzanie, prognozowanie), działania (dolnoprzepustowe, górnoprzepustowe i wszechprzepustowe, pasmowoprzepustowe i pasmowozaporowe), konstrukcji (liniowe i nielinowe, symetryczne i asymetryczne, rekursywne i nierekursywne, niezmiennie w czasie, adaptacyjne) i in. Do oceny działania filtrów można zastosować w obszarze czasu funkcję autokorelacji (ACF) lub autokorelacji cząstkowej (PACF). W dziedzinie częstotliwości natomiast stosuje się funkcję transformacji lub funkcję odpowiedzi częstotliwościowej. Robert J. Hodrick i Edward C. Prescott w roku 1997 [184] w wielokrotnie cytowanym artykule *Postwar U.S. Business Cycles: An Empirical Investigation* wykorzystali model filtru Kalmana [129] jako podstawę do skonstruowania własnego modelu filtracji, tj. filtru Hodricka-Prescotta (HP filter). Autorzy zaproponowali procedurę dekompozycji szeregu czasowego y_t (GNP, inflacji, stopy bezrobocia) na wygładzony zmienny trend g_t (składo-

wą wzrostu) oraz składową cykliczną c_t (stacjonarną resztę): $y_t = g_t + c_t$, gdzie nieznaną składową trendu jest równa: $g_t = 2g_{t-1} - g_{t-2} + \varepsilon_t$. Wygładzona składowa g_t wyliczana jest tak, aby jej suma kwadratów drugiej różnicy z szeregiem y_t nie była zbyt duża.

Problem optymalizacyjny został sformułowany w następujący sposób:

$$\min \left[\sum_{t=1}^T (y_t - g_t)^2 + \lambda \sum_{t=2}^{T-1} [(g_{t+1} - g_t) - (g_t - g_{t-1})]^2 \right] \quad (1.46)$$

ze względu na $\{g_t\}_{t=1}^T$. Miarą niezgodności c_t i ε_t w funkcji filtracji HP jest l . Filtr HP „przepuszcza” wahania o wyższych od wybranej (związanej z wyborem l) częstotliwościach, dlatego jest filtrem górnoprzepustowym. Filtr HP, jak każda praktyczna realizacja filtracji, nie jest filtrem „idealnym” i do jego wad zaliczyć należy:

- 1) wrażliwość na wydłużanie lub skracanie szeregów oraz na przekształcenia zmiennych
- 2) statystyczną inferencję (po uprzednim usunięciu składowych sezonowych z danych pierwotnych) i generowanie cykli pozornych,
- 3) nieumiejętne stosowanie filtru HP może generować cykle pozorne także dla szeregów niecyklicznych,
- 4) pozorną korelację szeregów poddanych przetwarzaniu filtrem HP. W praktyce filtr HP wyklucza trendy długofalowe, za to filtr Baxtera-Kinga (BK) służy wyeliminowaniu trendu deterministycznego lub stochastycznego wraz ze składową cykliczną. Według M. Baxtera i R.G. Kinga efektywna metoda tzw. detrendyzacji szeregów czasowych winna spełniać następujące warunki [335], [333]:
 - (a) filtr powinien wyeksponować składowe cykliczne w pożądanym zakresie częstotliwości bez zakłócania ich charakterystyk,
 - (b) filtr nie powinien prowadzić do zaistnienia przesunięć fazowych (i punktów zwrotnych) badanych szeregów,
 - (c) filtr powinien być w możliwie dużym stopniu aproksymować charakterystykę filtru „idealnego”,
 - (d) filtr powinien osłabiać składową trendową,
 - (e) uzyskana w procesie filtracji składowa cykliczna nie powinna zależeć od długości filtrowanego szeregu
 - (f) procedura filtracji powinna być operatywna.

Uwzględniając powyższe oczekiwania autorzy zaproponowali cyfrowy filtr pasmowy (liniowy, skończony rzędu K , z symetrycznymi wagami i dwustronną średnią ruchomą) będący aproksymacją filtru „idealnego”. Pozwala on określić

z badanego szeregu požądane pasmo (częstotliwości lub okresowości). W dziedzinie czasu filtr ten przekształca szereg wejściowy x_t jak średnia ruchoma, wyznaczając wyjściowy szereg y_t :

$$y_t = \sum_{j=-K}^K (a_j B^j x_{t+j}), \quad (1.47)$$

gdzie B oznacza operator przesunięcia ($B^j x_t = x_{t-j}$). W dziedzinie częstotliwości filtr jest opisywany transformatą Fouriera (spektrum) szeregu wejściowego i wyjściowego (operacja, która w dziedzinie czasu jest mnożeniem jest splotem w dziedzinie częstotliwości). Wartości B^j wyliczane są w wyniku minimalizacji funkcji strat jako różnicy między filtrem idealnym $A(\omega)$ i filtrem zastosowanym $B(\omega)$:

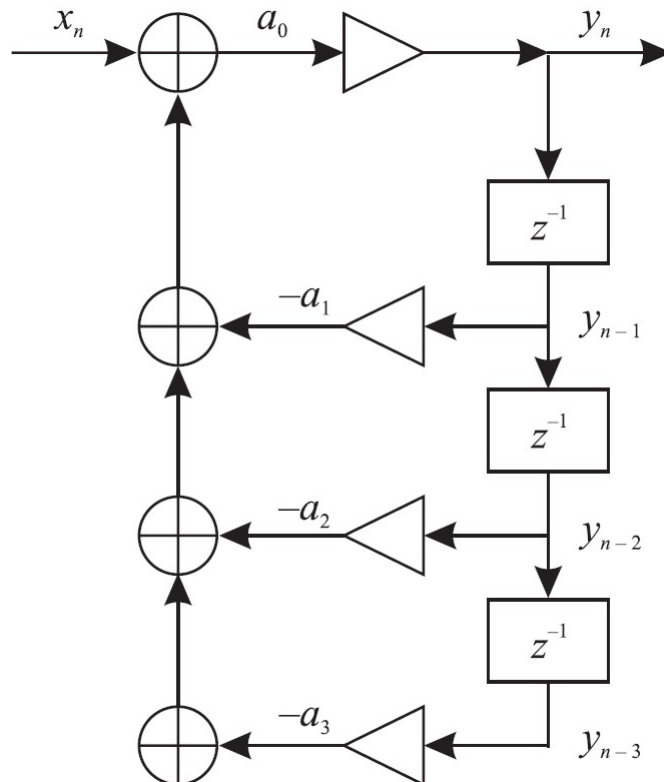
$$(1.48)$$

i gdzie $A(\omega)$ to funkcja transferowa filtru „idealnego”, natomiast $B(\omega)$ - filtru zastosowanego. Filtr Baxtera-Kinga podobnie jak inne cyfrowe filtry pasmowe jest poddawany krytyce ponieważ nie izoluje wzajemnie cykli szeregów z zintegrowaną składową trendową. Pierwsze różnice są „przepuszczane” przez filtr BK, w wyniku czego spektrum przefiltrowanego szeregu jest zależne od trendu nieprzefiltrowanego szeregu pierwotnego. Oprócz tego, gdy spektrum pierwotnego szeregu osiąga maksimum dla zerowej częstotliwości, filtr BK (podobnie jak i filtr HP) wykazuje istnienie, pozornej w istocie, cykliczności. R.G. King i M. Watson opisali to zjawisko jako „typowy przebieg spektralny stóp wzrostu”. Symetryczność filtru BK wyklucza wyłonienie pasm częstotliwości z początku i z końca pierwotnego szeregu czasowego; jego symetryczność nie gwarantuje też braku przesunięć fazowych oraz liniowej charakterystyki fazowej przefiltrowanego szeregu. Filtr BK może, podobnie jak to miało miejsce w przypadku filtru HP, generować cykle pozorne. W przeciwieństwie do filtrów o skończonej odpowiedzi impulsowej (ang. *Finite Impulse Response/FIR*), które istnieją wyłącznie w postaci cyfrowej, filtry o nieskończonej odpowiedzi impulsowej (NOI), określane angielskim skrótem IIR (Infinite Impulse Filter), są w istocie cyfrowymi odpowiednikami analogowych filtrów, tj. budowanych przy użyciu rezystorów, cewek, kondensatorów i wzmacniaczy operacyjnych. Metody projektowania, modelowania i realizacji filtrów analogowych są przedmiotem badań od wielu lat, a ich cyfrowe odpowiednik bazują na tzw. prototypach analogowych. Cechą odróżniającą filtry IIR od FIR jest ich odmienna topologia, to jest te pierwsze posiadają również sprzężenie zwrotne. Do wyznaczenia próbek wyjścia brane są nie tylko próbki sygnału wejściowego, ale także poprzednie wyniki filtracji (wartości wyjściowe). Filtr rzędu N używa do filtracji: bieżącej próbki sygnału, N poprzednich próbek sygnału oraz N poprzednich wyników filtracji. Filtry cyfrowe Butterwortha należą do klasy filtrów cyfrowych rekursywnych o nieskończonej

odpowiedzi impulsowej (IIR - *Infinite Impulse Response* - ang.). Do zalet cyfrowych filtrów NOI/IIR zalicza się to, że układ lub algorytm realizujący filtrację jest prostszy w rozumieniu liczby operacji arytmetycznych do wykonania oraz że w procesie projektowania cyfrowych filtrów NOI/IIR „wykorzystuje się wiedzę o filtrach analogowych”. Natomiast wadą tych filtrów jest trudność w otrzymaniu liniowej charakterystyki fazowej oraz fakt, że jego odpowiedź impulsowa „może zawierać nieskończoną liczbę próbek”, a układ może nie być stabilny. Nieliniowa faza eliminuje możliwość użycia takiego modelu w wielu praktycznych okolicznościach, tj. takich, w których jej liniowość jest warunkiem koniecznym. Ponadto projektant cyfrowych filtrów NOI/IIR „zwykle nie ma wpływu na kształt charakterystyki fazowej filtrów NOI”. Struktura blokowa filtru cyfrowego dla rzędu opóźnień $N = 3$ z linią opóźnienia tylko na wyjściu (blok z^{-1} oznacza układ opóźniający sygnał wejściowy o jedną obserwację) został przedstawiony na rys. 1.43. Zaprezentowany układ dyskretny realizuje zależność, która w dziedzinie czasu można przedstawić jako:

$$y(n) = x(n) - \sum_{j=1}^N a_j y(n-j). \quad (1.49)$$

Filtr Butterwortha jest filtrem dolnoprzepustowym o rzędzie = N . Kluczowym jest fakt, że w przypadku idealnych filtrów dolnoprzepustowych zachodzi konieczność odtworzenia charakterystyki amplitudowej sygnału, bez zwracania większej uwagi na charakterystykę fazową. Filtry te są stosowane m.in. do przetwarzania sygnałów akustycznych. Wynika to z faktu, że ludzki narząd słuchu nie odróżnia np. sygnałów różniących się tylko widmem fazowym. Zwiększanie liczby współczynników filtru, tj. jego rzędu powoduje, że charakterystyka amplitudowa filtru zbliża się do charakterystyki amplitudowej idealnego filtru dolnoprzepustowego.



Rysunek 1.43: Schemat blokowy filtru cyfrowego („tylko” NOI/IIR, AR - *Autoregressive* - ang.) dla $N = 3$ z linią opóźniającą tylko na wyjściu

Źródło: opracowanie własne na podstawie T.P. Zieliński, *Cyfrowe przetwarzanie sygnałów...*, s. 264.

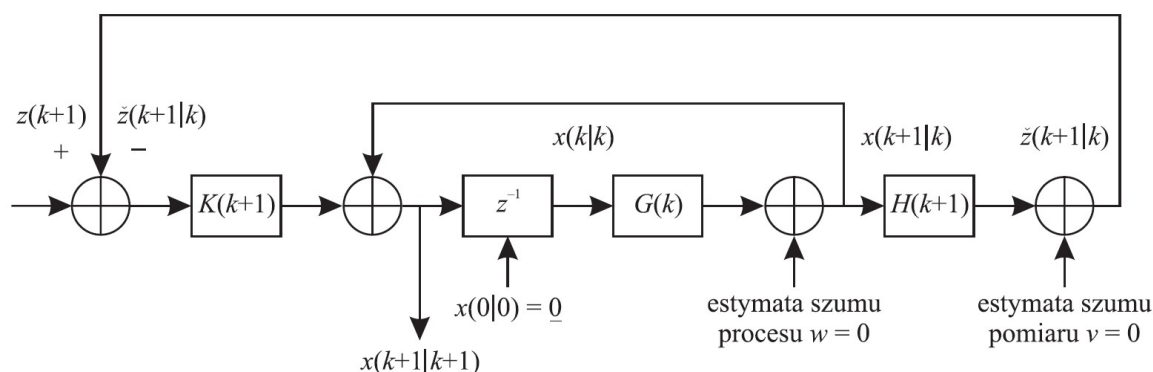
Filtr Kalmana (nazwa pochodzi od nazwiska jego autora, tj. R.E. Kalmana) [71] jako rekursywny, liniowy, adaptacyjny (a ponadto dyskretny, zmienny w czasie oraz mający skończoną wymiarowość) miał początkowo zastosowania techniczne i stopniowo został zastosowany w naukach ekonomicznych i w statystyce. Filtr ten filtruje szeregi czasowe (sygnały jednowymiarowe) bardziej na podstawie raczej ich własności statystycznych niż spektralnych. Filtr Kalmana w modelach przestrzeni stanów dla szeregów czasowych pełni rolę podobną do metody najmniejszych kwadratów w modelach regresji. Standardowa filtracja z pomocą rekursywnej, adaptacyjnej estymacji zakładanego (nieobserwowalnego) wektora x na podstawie obserwacji z i zakłóceń pomiarowych v sprowadza problem obliczeniowy poziomu minimalizacji błędów sumy kwadratów związanego z błędem (szumem) estymaty wektora $x : z = Hx + v$, gdzie na wektor x składa się N zmiennych, a na wektor $z - M$ zmiennych. Z tego powodu jakość oszacowania (estymacji) wektora x oceniana jest jako $\min(\hat{v}_1^2 + \hat{v}_2^2 + \dots + \hat{v}_M^2 = \min \sum_{i=1}^M \hat{v}_i^2$. Błąd pomiarowy pomiarowa w przypadku filtru Kalmana jest inny, minimalizowana jest bowiem wartość oczekiwana funkcji jakości estymacji (estymata wariancji): $\min\{E[(x - \hat{x})^T \cdot (x - \hat{x})]\}$. Natomiast w przypadku estymacji RLS (*Recursive Least Squares* - ang.) suma kwadratów błędów (szumów) pomiaru.

Schemat blokowy na rysunku 1.44 przedstawia procedurę estymacji filtru Kalmana.

Z formalnego punktu widzenia filtracja z wykorzystaniem filtru Kalmana (estymacja minimalno-średniokwadratowa, MMS, *Minimum Mean Squares* - ang.) istotnie różni się od filtracji z wykorzystaniem filtru RLS:

$$RLS : \begin{cases} x(k+1) = F(k) \cdot x(k) + w(k) \leftarrow \text{model procesu,} \\ z(k+1) = H(k+1) \cdot x(k+1) + v(k+1) \leftarrow \text{pomiar} \end{cases} \quad (1.50)$$

Pierwsze z równań opisuje obserwację procesu, drugie zaś - jego pomiar. Algorytm filtru Kalmana można w sposób uproszczony zapisać w następujący sposób:



Rysunek 1.44: Schemat blokowy filtru Kalmana

Źródło: opracowanie własne na podstawie [129].

- 1) poprzednia estymata stanu procesu,
- 2) prognoza nowego stanu
- 3) prognoza nowego pomiaru,
- 4) nowy pomiar,
- 5) błąd prognozy pomiaru,
- 6) korekta estymaty stanu,
- 7) powrót do punktu pierwszego.

W przypadku filtru Kalmana (FK), gdy szum ma charakter gaussowski, można udowodnić, że filtr Kalmana jest najlepszym filtrem. Gdy szum nie ma charakteru gaussowskiego filtr Kalmana jest najlepszym filtrem liniowym. Nawet gdy jest on stały w czasie, szum zaś stacjonarny, a współczynniki filtru są zmienne w czasie (cecha filtrów adaptacyjnych).

W ekonomii tego rodzaju filtry znajdują zastosowanie w wygładzaniu i filtrowaniu przebiegów czasowych, prognozowaniu, w modelowaniu nieliniowym, estymacji parametrów modeli ARMA, w uogólnionej metodzie najmniejszych kwadratów (GLS - *generalised least squares method* - ang.) oraz identyfikacji układów liniowych, w sztucznych sieciach neuronowych, w synchronizacji układów chaotycznych i innych.

Podstawowym problemem w filtrowaniu ekonomicznych szeregów czasowych jest to, że w większości przypadków nie można, w sposób niebudzący wątpliwości, ustalić typu ich stacjonarności. Fakt ten powoduje, że nie można zastosować „poprawnej” a priori metody filtracji. Możliwym, w danej sytuacji, rozwiązaniem wydaje się ocena efektywności filtracji poprzez zastosowanie dużej liczby różnych filtrów i porównanie ich działania.

1.5.2 Próbkowanie oszczędne

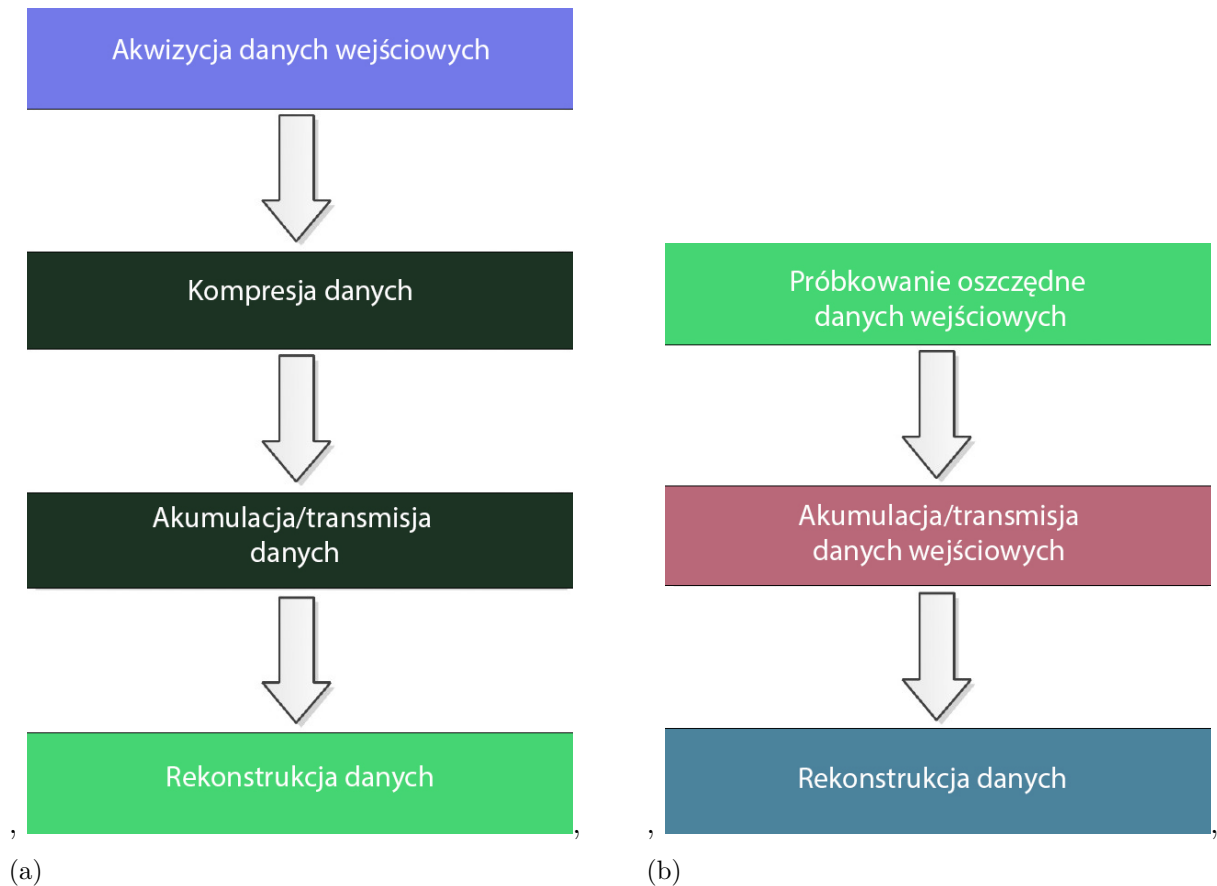
Próbkowanie oszczędne, *ang. compressed sensing* jest nowym protokołem akwizycji danych. Standardowe podejście do realizacji procesu akwizycji danych polega na "kompletnym" pomiarze sygnałów z częstotliwością określoną przez twierdzenie o próbkowaniu. Z pewnego punktu widzenia, o którym będzie mowa w dalszej części tej pracy, dane często są nadmiarowe. Konieczna jest zatem ich późniejsza kompresja, zazwyczaj o charakterze stratnym (np. powszechnie stosowany algorytm kompresji obrazów JPEG). Operacja ta jest konieczna ze względów technicznych i ekonomicznych, bowiem dopiero zmniejszone zbiory danych można transportować lub składować. Należy zauważyć, że coraz większy udział w rynku mają urządzenia mobilne, a te nie nakładają aż tak wysokich, jak urządzenia stacjonarne, wymogów w kwestii jakości wyświetlania. Rosnący udział mają serwisy internetowe, społecznościowe i inne, a tego typu systemy nie pozwalają na umieszczanie w swoich zasobach danych nieskompresowanych. Kompresja o charakterze stratnym powoduje, że zaraz po dokonaniu pomiaru część danych jest trwale eliminowana. Próbkowanie oszczędne to protokół akwizycji danych, który definiuje potrzebną do poprawnej rekonstrukcji, ilość danych pomiarowych. Metoda ta skłania się ku pomiarowi jedynie "istotnych składowych sygnału", pozostałą część danych klasyfikując jako dane nadmiarowe [165], [19], [166]. Zastosowanie takiego schematu decydowania o istotności danych możliwe jest jednak przy spełnieniu dodatkowych warunków: wyboru odpowiedniej transformacji, w której to bazie można orzec o rzadkości sygnału badanego oraz niekoherencji w procesie pomiaru.

Cyfrowe przetwarzanie danych czasu ciągłego (analogowych) wymaga ucyfrowienia sygnału analogowego. Sygnał ciągły reprezentowany jest przez nieskończoną liczbę chwil czasowych, a do reprezentacji wartości sygnału wymagana jest nieskończona ilość bitów. Sygnał czasu ciągłego jest zatem ciągły zarówno w osi wartości jak i argumentu. Z uwagi na oczywiste ograniczenia systemów dyskretnych nie jesteśmy w stanie w sposób

cyfrowy opisać takiego sygnału. Konieczna jest zatem konwersja analogowo cyfrowa, która zamienia sygnał czasu ciągłego w sygnał dyskretny na drodze trzech zasadniczych operacji: próbkowania, kwantyzacji oraz kodowania. W tych okolicznościach rodzi się pytanie, jakie są graniczne wartości parametrów tych trzech operacji by możliwe było odtworzenie sygnału ciągłego z jego dyskretnej reprezentacji, tj. ciągu bitów. Odpowiedź na to pytanie przynosi twierdzenie o próbkowaniu. Precyzuje ono, że sygnał można odtworzyć, jeśli ma on ograniczone pasmo i próbkowanie wykonywane jest z odpowiednio wysoką częstotliwością, tj. próbki na etapie procesu akwizycji danych pobierane były dostatecznie często. Klasyczne podejście do tego procesu wymaga pomiaru wszystkich danych (z co najmniej częstotliwością Nyquista). Jeśli badany sygnał w swoim widmie częstotliwości nie zawiera częstotliwości większych niż pewne B (pasmo określone jest przez naturalny rozmiar sygnału), to można go perfekcyjnie odtworzyć przy próbkowaniu z częstotliwością co najmniej dwukrotnie większą niż B , tj. z największą częstotliwością obecną w reprezentacji częstotliwościowej sygnału. Spełnienie tego warunku gwarantuje tzw. perfekcyjną rekonstrukcję, jednak kosztem użycia bardzo dużej ilości danych pomiarowych. Wymaganie takiej liczby niezależnych pomiarów jest niezbędne tylko w przypadku sygnałów nie posiadających żadnego uporządkowania (odbieranych przez człowieka jako szum). Okazuje się jednak, że wiele użytecznych sygnałów jest w pewien sposób uporządkowana, bądź też posiada pewne dodatkowe właściwości pozwalające na ograniczenie ilości danych pomiarowych. Oznacza to, że w klasycznym podejściu liczba pomiarów niezbędna do poprawnej rekonstrukcji jest określona przez naturalny rozmiar danych zdeterminowany przez częstotliwość próbkowania. W ten sposób pozyskane dane pomiarowe są zazwyczaj silnie nadmiarowe. Argumentem potwierdzającym tę tezę jest powszechność stosowania algorytmów kompresji multimedialnej. Strategia ta umożliwia znaczną redukcję wolumenów danych bez uszczerbku na jakości rekonstrukcji [177]. Najpopularniejsza metoda kompresji obrazów, tj. JPEG swe działanie opiera na transformacji obrazów do postaci, w której możliwe jest orzeczenie o nadmiarowości części danych i w dalszej kolejności możliwe jest ich usunięcie. Gdyby w określonych okolicznościach, tj. w bazie pewnego przekształcenia ortogonalnego nie można było stwierdzić ich rzadkości, to kompresja w ogóle nie byłaby możliwa. Reasumując, w klasycznym podejściu do budowania bazy danych pomiarowych (czyli de facto zrobienia zdjęcia) zachodzi konieczność pomiarów wszystkich składowych sygnału. Dopiero w dalszej kolejności możliwe jest wykluczenie tych, których amplituda jest niska, czyli wykluczenie składowych, które niosą informacje, które bez większej straty, można usunąć. Problem zaczyna zyskiwać na znaczeniu w przypadku, gdy budowanie bazy pomiarowej, czyli zdobywanie informacji o modelu wiąże się wydłużeniem czasu pomiaru, a to może mieć wpływ na elementy środowiska lub ilustrowany obiekt. Przykładem zagadnienia tego typu jest proces obrazowania medycznego szeroko stosowany w diagnostyce

medycznej.

W wielu procedurach diagnostyki medycznej stosuje się czynniki stanowiące zagrożenie dla zdrowia i życia człowieka. Problem nabiera na znaczeniu np. w przypadku promieniowania rentgenowskiego. Możliwość skrócenia czasu akwizycji danych, to zmniejszenie ilości niezbędnych do wykonania pomiarów (w tym wypadku chodzi o zmniejszenie dawki promieniowania) a tym samym znaczne zmniejszenie szkodliwości badania. Zastosowanie próbkowania oszczędnego pozwala tutaj na znaczne zmniejszenie wymaganych pomiarów, a tym samym zachowanie blisko 100 procentowej dokładności rekonstruowanego obrazu diagnostycznego. W standardowym podejściu zastosowania próbkowania oszczędnego nie jest jednak możliwe, gdyż proces pomiaru jest adaptacyjny (zależy od danych). Oznacza to, że przed rozpoczęciem procesu akwizycji danych (np. wykonaniem fotografii) nie wiemy, które składowe będą istotne, musimy zmierzyć wszystkie, a dopiero później usuwamy te, których amplituda jest mała. Istotą próbkowania oszczędnego jest określenie protokołu pomiaru, który nie będzie zależał od danych i nie będzie potrzebował do przeprowadzenia rekonstrukcji wszystkich, również małych, składowych sygnału. Rozwiązaniem jest stworzenie protokołu, który nie zależy od danych i pozwala na odzyskanie wszystkich ważnych składowych sygnału, bazując na relatywnie małej liczbie pomiarów w stosunku do rzeczywistego rozmiaru sygnału wejściowego. Próbkowanie oszczędne unika mierzenia wszystkich składowych. W praktyce realizowany jest pomiar stosunkowo niewielkiej ilości danych (w stosunku do naturalnego rozmiaru sygnału), a co równie istotne, jest to proces niezależny od danych. Rysunek 1.45 przedstawia porównanie wymiar nakładu pracy w przypadku metody próbkowania oszczędnego w porównaniu do standardowego podejścia. W przypadku klasycznego podejścia (rys. 1.45a) duży nakład pracy jest konieczny w procesie akwizycji oraz kompresji danych. Wykonuje się dużą liczbę pomiarów, a następnie przeprowadzana jest kompresja danych, która ma za zadanie zredukować ich wolumen. Proces rekonstrukcji jest tutaj relatywnie mało skomplikowany. W przypadku próbkowania oszczędnego (rys. 1.45b) proces budowania bazy pomiarowej nie jest pracochłonny. Duży nakład pracy natomiast wymagany jest jednak w procesie rekonstrukcji. Proces ten jest bardzo złożony i opiera się na przetworzeniu niewielkiej ilości danych na temat analizowanego sygnału. W niniejszej pracy proces ten został zastosowany jako mechanizm odszumiania szeregów czasowych cen traktując je jako zubożoną reprezentację wymagającą wyspecjalizowanej rekonstrukcji przy pomocy metodologii rekonstrukcji sygnałów próbkowanych oszczędnie, tj. rzadkich. Proces rekonstrukcji sygnału próbkowanego oszczędnie jest skomplikowany ponieważ posiadana jest stosunkowo niewielka ilość informacji wstępnej nt. rekonstruowanego sygnału oraz w dalszej kolejności konieczne jest zastosowanie skomplikowanych algorytmów zapewniających poprawną rekonstrukcję sygnału.



Rysunek 1.45: Porównanie algorytmów próbkowania oszczędnego (b) do podejścia klasycznego (a)

1.5.3 Warunki zastosowania próbkowania oszczędnego

Rekonstrukcja sygnału próbkowanego oszczędnie jest możliwa po spełnieniu następujących warunków:

- własności strukturalnych danych, tj. ich rzadkości,
- ograniczeń związanych z organizacją procesu akwizycji danych, które musi spełnić proces budowania bazy pomiarowej. Jest to tzw. niekoherencja pomiaru.

Rzadkość,

Pojęcie rzadkości odnosi się do własności sygnału. Wiąże się ono z koniecznością reprezentacji sygnału w zwartej formie. W praktyce oznacza to, że istnieje baza pewnego przekształcenia, w której sygnał jest reprezentowany przez niewielką liczbę składowych różnych od zera. Oznacza to konieczność określenia bazy, w której sygnał będzie rzadki (może to być np. baza, w której sygnał odpowiada dokładnie jednemu z wektorów bazowych). Jednak takie przedstawienie problemu oznacza, że wracamy do sytuacji, w której proces

akwizycji zależy od sygnału. Nowy protokół musi oznaczać okoliczność, w której w jednej z powszechnie używanych baz (np. transformacji kosinusowej, fourierowskiej, falkowej, itp.) sygnał ma niewielką ilość niezerowych składowych.

Sygnały nazywane mianem rzadkich to sygnały, których duża część składowych ma wartość równą 0. Sygnał x o długości determinowanej przez parametr n jest s -rzadki, jeżeli s jego składowych ma wartości różne od zera, a $n - s$ składowych ma wartości równe zero. Mamy wówczas formalne podstawy do orzeczenia o rzadkości sygnału, tj.:

$$s \ll n \quad (1.51)$$

gdzie:

$s, n \in N^+$ Natomiast w przypadku tzw. sygnałów prawie rzadkich (tj. kompresowalnych),

istotna jest liczba składowych takich sygnałów ma wartości bliskie zero, zatem ich pominięcie nie ma większego wpływu na dokładność reprezentacji sygnału. Z formalnego punktu widzenia, sygnał x jest sygnałem prawie rzadkim, jeśli istotna część jego składowych ma wartość bliską 0. Podobnie, ma to miejsce w przypadku sygnału rzadkiego możemy dla sygnału o długości n wyznaczyć s składowych, których wartości są znaczące w porównaniu z pozostałymi składowymi. Po przeprowadzeniu procesu sortowania składowych sygnału, ze względu na wartość bezwzględną, to prawie s -rzadki sygnał spełnia relację:

$$\sum_{i=s+1}^n |x_i| \ll \sum_{i=1}^s |x_i| \quad (1.52)$$

gdzie:

$s, n \in N^+$

x_i - składowe sygnału.

Spełnienie tego warunku oznacza, że znaczna część składowych sygnału ma wartości równe 0, co pozwala na ich wykluczenia, prowadząc tym samym do uproszczenia problemu.

Na ilustracji 1.46 przedstawiono trzy sygnały: a) sygnał pozbawionym rzadkości, b) sygnał rzadkim, c) sygnał prawie rzadkim. Należy zauważyć, że wiele sygnałów w swojej natywnej postaci nie wykazuje jednak, pożądanym tutaj, cech rzadkości. Oznacza to, że jeżeli celem jest zastosowanie próbkowania oszczędnego, konieczna będzie jest ich reprezentacja w innej bazie, gdyż rzadkość może się odkrywać dopiero po zastosowaniu właściwej bazy dla danego sygnału [177]. Na ilustracji nr 1.46 przedstawiono przykład sygnału, który w swojej naturalnej postaci (ciąg pikseli) nie wykazuje cech rzadkości, a te ukazują się po transformacji do bazy przekształcenia Fouriera. Obraz w swojej pierwotnej postaci (rys. 1.46a) nie jest rzadki. Zrealizowanie dwuwymiarowej transformaty

Fouriera tego obrazu ujawnia fakt, że wartości bezwzględnych składowych Fouriera w znacznej większości mają wartości bliskie 0 (rys. 1.46c pokazuje te wartości posortowane w porządku rosnącym). Odrzucenie około 85% składowych (około 85% najmniejszych wartości zostało wyzerowanych) i wykonanie transformaty odwrotnej prowadzi do wyznaczenia obrazu, którego jakość jest zbliżona do obrazu, który nie był poddany kompresji multimedialnej (rys. 1.46b). Spełnienie warunku rzadkości jest konieczne ale nie jest wystarczające. Celem jest pozyskanie informacji i wszystkich składowych z jedynie skończonego ich podzbioru. Oznacza to, że informacja ta powinna być pozyskana z s pomiarów (gdzie s oznacza rzadkość sygnału). Niestety, jak się okazuje, jest to niemożliwe, gdyż musielibyśmy z góry wiedzieć, które składowe należałoby wskazać jako te kluczowe [152]. Oznacza to, że warunkiem zapewniającym poprawną rekonstrukcję jest warunek wykonania większej liczby pomiarów niż określa to rzadkość sygnału, ale liczba ta może być dużo mniejsza od rozmiaru sygnału:

$$s < k \ll n \tag{1.53}$$

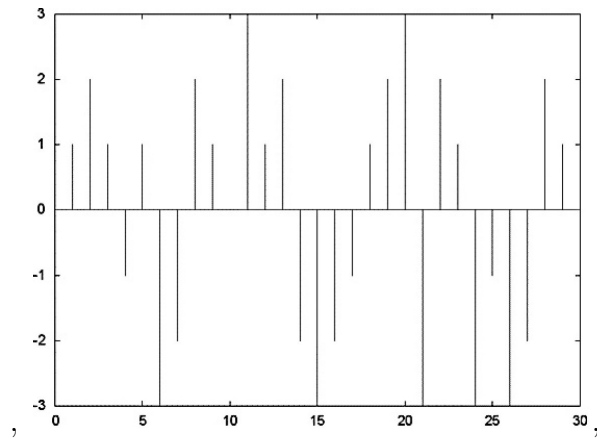
gdzie:

$$s, k, n \in N^+$$

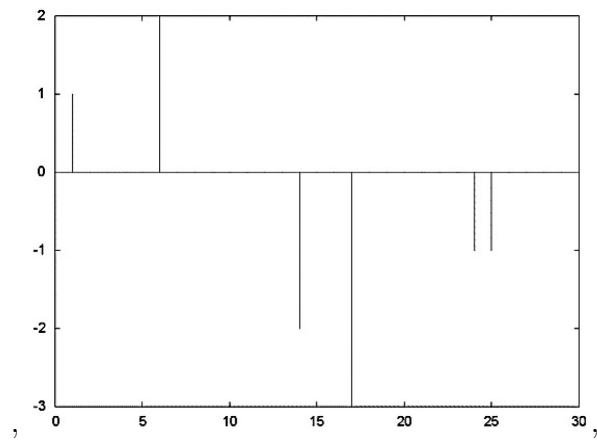
s - rzadkość sygnału,

k - liczba niezbędnych pomiarów,

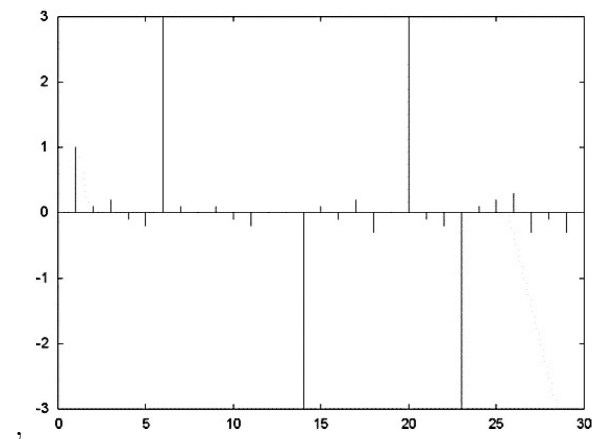
n - rozmiar sygnału.



(a)



(b)



(c)

Rysunek 1.46: a) sygnał o wartościach niezerowych w przedziale $[-3; 3]$, b) sygnał rzadki, c) sygnał prawie rzadki

Niekoherencja pomiaru,

Jak wcześniej wspomniano, rzadkość staje się warunkiem koniecznym, ale niewystarczającym do przeprowadzenia, wymaganej dla próbkowania oszczędnego, akwizycji danych. Poza tym warunkiem, konieczne jest spełnienie warunku dotyczącego metodologii prze-

przewodzenia pomiaru. Sam proces akwizycji danych musi być właściwie uporządkowany [151], tj. nie pomijać istotnych składowych sygnału. W przypadku sygnałów o charakterze rzadkim jest to kluczowe (w jego reprezentacji niewielka liczba składowych niesie istotne informacje i pominięcie choćby części z nich może prowadzić do niepowodzenia procesu rekonstrukcji sygnału). Można jednak temu zapobiec na drodze spełnienia warunku niekoherencji pomiaru [194]. Pomiar sygnału wiąże się z koniecznością porównania sygnału ze wzorcem. Ten, konieczny tutaj, punkt odniesienia jest konieczny z uwagi na wymagany tutaj układ wzorców zwany bazą pomiarową. Pomiar oznacza mapowanie (rzutowanie) sygnału na bazę pomiarową. Na ilustracji nr 1.46 rozmiar bazy pomiarowej wynosi $n = w2$, a w oznacza rozdzielczość obrazu. Baza pomiarowa jest zdefiniowana w następujący sposób:

$$\Phi = \{\varphi_i\}_{i=1,\dots,n} \quad (1.54)$$

gdzie:

Φ - baza pomiarowa,

φ_i - składowe bazy pomiarowej.

Rezultat pojedynczego pomiaru oznacza rzutowanie sygnału na i -tą składową bazy pomiarowej:

$$b_i = f \cdot \varphi_i \quad (1.55)$$

gdzie:

b_i - wynik pomiaru i -tej składowej sygnału,

f - sygnał,

φ_i - i -ta składowa bazy pomiarowej.

Problem ten w postaci macierzowo-wektorowej przyjmuje formę:

$$b = \Phi f \quad (1.56)$$

gdzie:

b - rezultat pomiarów sygnału,

f - sygnał,

Φ - baza pomiarowa, gdzie wiersze macierzy oznaczają wektory bazowe.

Formalnie, sygnał można przedstawić, korzystając z jego rozwinięcia w bazie, w której jest on rzadki:

$$f = \sum_{i=1}^n x_i \Psi_i \quad (1.57)$$

gdzie:

Ψ_i - i -ta składowa bazy sygnału,

x_i - i -ta składowa sygnału,

f - sygnał.

W notacji macierzowej można to przedstawić w następującej formie:

$$f = \psi x \quad (1.58)$$

gdzie:

ψ - macierz, której kolumny to kolejne wektory bazy w której sygnał jest rzadki,
 x - wektor składowych sygnału x ,
 f - sygnał.

Proces pomiaru w notacji macierzowej sprowadza się do mnożenia wektora f przez macierz pomiarową, tj.:

$$y = \Phi f = \Phi \psi x = Ax \quad (1.59)$$

gdzie:

Φ - baza pomiarowa,
 ψ - baza sygnału, w której sygnał jest rzadki,
 x - wektor składowych sygnału x ,
 f - sygnał.
 A - macierz pomiarowa.

Z równania nr 1.59 wynika, że w procesie akwizycji sygnału będzie mowa o dwóch bazach, tj. bazie sygnału (ψ), w której sygnał jest rzadki oraz bazie pomiarowej (Φ) wykorzystywanej w celach akwizycji danych. Ograniczenie się do jednej z nich nie jest satysfakcjonujące bo musielibyśmy wykonać liczbę pomiarów rzędu n (gdzie n oznacza ilość składowych sygnału). Oznacza to, że każdy pomiar z wykorzystaniem bazy sygnału dostarcza informacji tylko o jednej składowej sygnału. Wiąże się to z faktem dużej zależności pomiędzy bazą sygnału i bazą pomiarową (w tym przypadku zależność jest maksymalna, bo stosujemy te same bazy). Oznacza to, że konieczne jest wykorzystanie innej bazy, której stopień zależności z bazą sygnału jest jak najmniejszy.

Miarą tego rodzaju zależności jest koherencja. Może zostać ona scharakteryzowana w następujący sposób:

$$\mu(\Phi, \psi) = \sqrt{n} \cdot \max_{1 \leq k, j \leq n} \left| \left\langle \varphi_k, \Psi_j \right\rangle \right| \quad (1.60)$$

gdzie:

Φ - baza pomiarowa,
 Ψ - baza sygnału, w której sygnał jest rzadki,
 n - rozmiar baz,
 φ_i - składowe bazy pomiarowej,
 ψ_i - wektor bazowy sygnału.

Miarą koherencji pomiędzy dwoma bazami jest jak największa korelacja pomiędzy ich dwoma dowolnymi wektorami bazowymi. Oznacza ona największą zależność pomiędzy nimi. Założywszy, że obie bazy są bazami ortonormalnymi, możemy łatwo udowodnić, że koherencja przyjmuje wartości z zakresu $[1/\sqrt{n}, 1]$. Zważając na to, że iloczyn skalarny dwóch wektorów bazowych może przyjąć wartość maksymalną równą 1 (długości wektorów są równe jeden i cosinus kąta pomiędzy nimi również jest równy 1). Natomiast wartość minimalna maksimum iloczynu skalarnego dwóch wektorów bazowych wynosi $1/n$. Wartość ta jest osiągana wtedy, gdy wszystkie składowe wektora bazowego jednej z baz mają wartość $1/n$, a drugi wektor bazowy ma wartość jednej ze składowych równą 1, a wartości pozostałych składowych są równe 0. Ma to miejsce w sytuacji, gdy jeden wektor bazowy ma taką samą wartość składowych, a drugi posiada wartość 1 na jednej ze składowych i poza tym wartości 0 [273].

Koherencja jednak winna być jak najmniejsza. W takich okolicznościach bowiem pojedynczy pomiar testuje wartości wielu składowych sygnału [149]. Prowadzi to do uzyskania pełnej informacji o sygnale przy niewielkiej ilości pomiarów. Spełnienie tego warunku gwarantuje wierną rekonstrukcję sygnału (spełniając jednocześnie pozostałe, wymagane warunki takie, jak rzadkość).

Zakładając, że macierz pomiarowa jest macierzą bazową sygnału, sygnał reprezentowany jest jak na rysunku 1.46a przez wektor o poniższej postaci:

$$x^T = [1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, -2, 0, 0, -3, 0, 0, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0] \quad (1.61)$$

Jeśli zastosujemy jako jeden z wierszy macierzy pomiarowej (wektor bazowy sygnału x) wektor o postaci:

$$a_1 = [1, 0] \quad (1.62)$$

wtedy

$$a_1 x = 1 \quad (1.63)$$

W ten sposób zdobywamy dokładną wiedzę jedynie o pierwszym elemencie wektora x [150]. Uzyskanie wiedzy o wszystkich składowych wektora x możliwe jest po dokonaniu tylu pomiarów, ile składowych posiada wektor x . Liczba ta jednak powinna być jak najmniejsza. Z tego powodu konieczne staje się wybranie innych wektorów pomiarowych, takich które mają jak najmniejszą koherencję z wektorami naturalnej bazy sygnału. Wektorem spełniającym te wymagania jest wektor o następujących składowych:

$$a_2 = [0.18, 0.18, 0.18, \dots, 0.18] \in R^{30} \quad (1.64)$$

gdzie:

$0.18 \approx 1/\sqrt{30}$ - wektor x składa się z 30 składowych.

Użycie wektora bazowego a_2 prowadzi do zdobycia informacji o pewnej kombinacji składowych wektora sygnału, które są niezerowe. W powyższym przykładzie wektor x posiada 6 niezerowych składowych. Przeprowadzenie sekwencji takich pomiarów prowadzi do określenia koniecznej liczby istotnych składowych sygnału. **Konieczna liczba pomiarów,**

W literaturze wskazano, że do zastosowania próbkowania oszczędnego konieczne jest określenie liczby składowych sygnału, poziomu jego rzadkości oraz koherencji (zależnej od stosowanych baz). Wartości te można scharakteryzować w następujący sposób:

$$k \geq C \cdot \mu^2(\Phi, \psi) \cdot s \cdot \log(n) \quad (1.65)$$

gdzie:

C - pewna stała dodatnia,

μ - koherencja pomiędzy bazami Φ oraz ψ ,

s - rzadkość sygnału,

n - liczba składowych sygnału.

Oznacza to, że koherencja posiada znaczny wpływ na liczbę pomiarów, jakie są konieczne by możliwa była poprawna rekonstrukcja sygnału. Pozostałymi czynnikami pozostają te, na które nie mamy wpływu, tj. rzadkość sygnału oraz jego naturalny rozmiar.

Własność RIP,

Jak wspomniano powyżej, kluczowym elementem dla skuteczności działania metod próbkowania oszczędnego jest dobór właściwej, dla danego przypadku, bazy pomiarowej. Proces ten związany z optymalną akwizycją danych (od doboru baz zależy wygląd macierzy pomiarowej), a przeprowadza się go tak by koherencja pomiędzy bazą sygnału (w której jest on rzadki) a bazą stosowaną w pomiarze była jak najmniejsza. Metodologia doboru bazy pomiarowej opierająca się na koherencji nie jest praktyczna. Umożliwia ona jednak sprawdzenie, czy dana macierz pomiarowa ma odpowiednio niską koherencję. Jest ona miarą, która pozwala sprawdzić, czy dana macierz pomiarowa będzie odpowiednia, ale nie określa metodologii jej zapewnienia [55]. Dowiedziono, że sposobem pozwalającym na dobranie macierzy pomiaru, która będzie posiadać niewielką koherencję z bazą sygnału badanego jest spełnienie własności ograniczonej izometrii (Restricted Isometry Property (RIP)). Dla jej określenia konieczne jest wprowadzenie stałej izometrii δ_s , określonej dla danej rzadkości s , jako najmniejszej wartości spełniającej poniższy warunek:

$$(1 - \delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\|x\|_2^2 \quad (1.66)$$

gdzie:

$x \in R^n$ - badany sygnał (rzadki),

$A \in R^{m \times n}$ - macierz pomiaru,
 s - rzadkość sygnału x ,
 δ_s - stała izometrii macierzy A .

Spełnienie powyższej własności gwarantuje poprawną rekonstrukcję sygnału rzadkiego. Spełnienie przez macierz A własności RIP ma miejsce, jeżeli wartość δ_s jest istotnie różna od 1, wówczas też macierz pomiaru zachowuje długości wektorów, co z kolei implikuje, że wszelkie dokonane pomiary nie będą prowadzić do powstawania wektorów zerowych. Oznacza to, że możliwa będzie rekonstrukcja sygnału, gdyż w procesie pomiaru nie będzie tracona informacja o nim.

Cechą jaką musi charakteryzować macierz, która zachowuje odległości jest ortogonalność. Jeżeli macierz A w praktyce ma zachowywać się jak macierz ortogonalna, dowolny podzbiór jej kolumn musi być w przybliżeniu ortogonalny.

W rzeczywistości oznacza to, że możliwe jest wybranie bazy pomiarowej gwarantującej niską koherencję bez wcześniejszej znajomości bazy sygnału. Cechę tę mają macierze losowe o różnych rozkładach, np. rozkładzie Gaussa.

Rekonstrukcja sygnału,

Proces pomiaru danych można zaprezentować przy pomocy operacji macierzowych, gdzie macierzą pomiaru oddziałujemy na wektor reprezentujący sygnał. Przeprowadzenie takiej operacji prowadzi do określenia wektora reprezentującego badany sygnał w rzutowaniu na wzorce określone w macierzy pomiaru. Formalnie, proces ten sprowadza się do zrealizowania następujących kroków:

$$y = Ax \tag{1.67}$$

gdzie:

x - badany sygnał,
 A - macierz pomiaru,
 y - wektor pomiaru.

Proces rekonstrukcji sygnału, w przypadku gdy macierz A jest macierzą kwadratową jest operacją prostą i wymaga jedynie przemnożenia wektora y przez macierz odwrotną do A . W metodach próbkowania oszczędnej macierz A nie jest jednak macierzą kwadratową. Liczba zrealizowanych pomiarów jest zazwyczaj znacznie mniejsza od liczby składowych sygnału badanego. Macierz A ma wymiary prostokąta, gdzie jej liczba wierszy odpowiada rozmiarowi wektora y , a więc liczbie dokonanych pomiarów.

W metodach próbkowania oszczędnej liczba pomiarów jest wielokrotnie mniejsza od rozmiaru sygnału. Rekonstrukcja sygnału w takich okolicznościach prowadzi do

określenia wielu możliwych rozwiązań (pozwala jedynie na wyznaczenie zależności pomiędzy zmiennymi). Z tego powodu, zasadne jest wprowadzenie dodatkowych warunków, które zapewniają jednoznaczność rozwiązań. Jednym z nich jest włączenie faktu, że badany sygnał ma być sygnałem rzadkim, a co za tym idzie znaczna część jego składowych powinna być równa 0. Sprowadza to problem do następującej postaci:

$$\min_x \|x\|_{l_0}, \left|_{Ax=y} \right. . \quad (1.68)$$

Ten problem obliczeniowy sprowadza się do rozpatrzenia wszystkich rozwiązań $Ax = y$, a następnie wybrania tego, które minimalizuje liczbę składowych niezerowych. Jest to problem natury optymalizacyjnej, gdzie norma l_0 oznacza liczbę niezerowych elementów w wektorze.

Oznacza to niestety, że jest on problemem NP-zupełnym i w praktyce nie jest możliwe jego rozwiązanie w sensownym czasie. Z tego powodu zaleca się zastąpienie normy l_0 inną normą. Zazwyczaj stosuje się w tych okolicznościach normę l_2 :

$$\|x\|_{l_2} = \sqrt{\sum_i x_i^2}. \quad (1.69)$$

Jej użycie jednak nie daje zadowalających wyników. Faworyzuje ona istnienie wielu składowych sygnału o niewielkich wartościach, co w metodach próbkowania nie jest korzystne, bo konieczne jest odszukanie niewielkiej liczby składowych o istotnych wartościach. Z tego powodu, konieczne jest zastosowanie jeszcze innej normy [155]. W literaturze wskazano zasadność zastosowania normy l_1 :

$$\|x\|_{l_1} = \sum_i |x_i|. \quad (1.70)$$

Ostatecznie, problem rekonstrukcji danych w metodach próbkowania oszczędnego jest problemem optymalizacyjnym, który można wyrazić w następujący sposób:

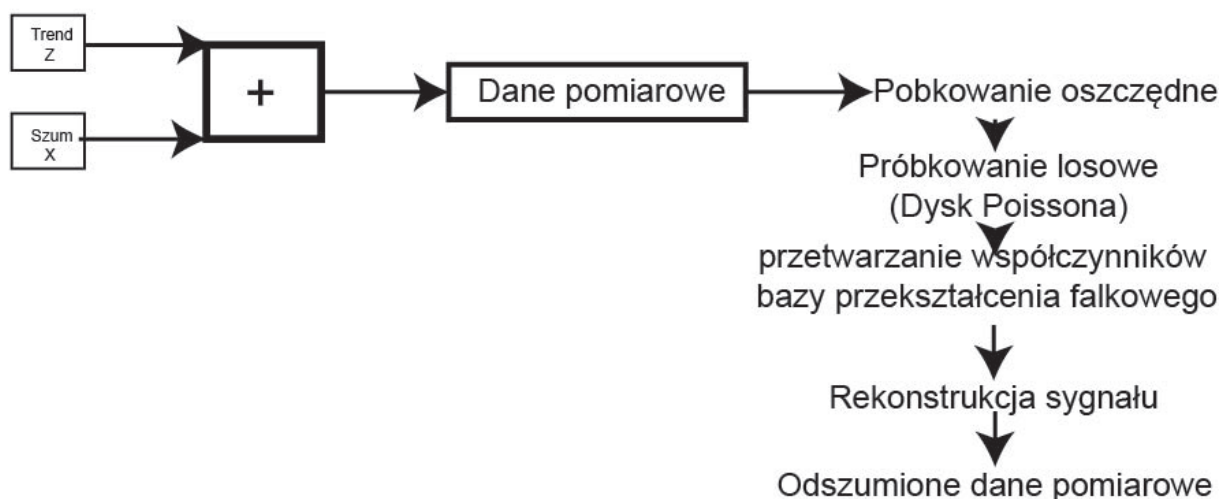
$$\min_x \|x\|_{l_1}, \left|_{Ax=y} \right. . \quad (1.71)$$

Jest to problem programowania liniowego o wielomianowej złożoności obliczeniowej.

Rozdział 2.

Proponowany algorytm prognozowania cen odtłuszczonego mleka w proszku

Uczenie maszynowe okazało się obiecującą techniką przewyższania niedogodności zidentyfikowanych w obszarze ekonometrii. Dowiedziono również, że techniki uczenia głębokiego, w tym splotowe sieci neuronowe (CNN), są skuteczne w identyfikowaniu, klasyfikowaniu i przewidywaniu wzorców obserwowanych w szeregach czasowych. Należy zauważyć, że metody CNN mogą uwzględniać immanentną złożoność danych, np. czasoprzestrzenność, chaotyczność i niestacjonarność, tylko w przypadkach, gdy są zaprojektowane i dopasowane do określonego zbioru danych i konkretnego zastosowania. Aby CNN działały, trzeba je zainicjować dużą liczbą etykiet [294].



Rysunek 2.1: Odszumianie danych pomiarowych

Metodologia głębokiego uczenia opiera się na paradygmatach uczenia maszynowego i zwykle składa się z dwóch głównych etapów: rekurencyjnych i splotowych sieci neuronowych. Do tej pory powszechnie uważano, że metodologia CNN wymaga użycia dwuwymiarowych obrazów wejściowych jako wektora wejściowego. Wymóg ten sprawia, że korzystanie z CNN w prognozowaniu rynków finansowych jest dość trudne. Aby rozwiązać ten problem, macierze jednowymiarowe, które są naturalnym sposobem reprezentacji wektorów danych szeregów czasowych cen OMP, muszą zostać sprowadzone do postaci macierzy dwuwymiarowych [274]. Celem niniejszej pracy jest przedstawienie metodologii uczenia splotowych sieci neuronowych wektorami danych odszumianych przy użyciu protokołu rekonstrukcji sygnałów rzadkich próbkowania oszczędnego.

W zaproponowanym algorytmie dane wejściowe są wstępnie przetwarzane i znormalizowane, co ma wpływ na dalsze etapy algorytmu. Następnym krokiem jest operacja jednowymiarowego splotu i wskazanie kluczowych cech (np. wolumenu, ceny wysokiej i niskiej oraz ceny zamknięcia) jako danych wejściowych.

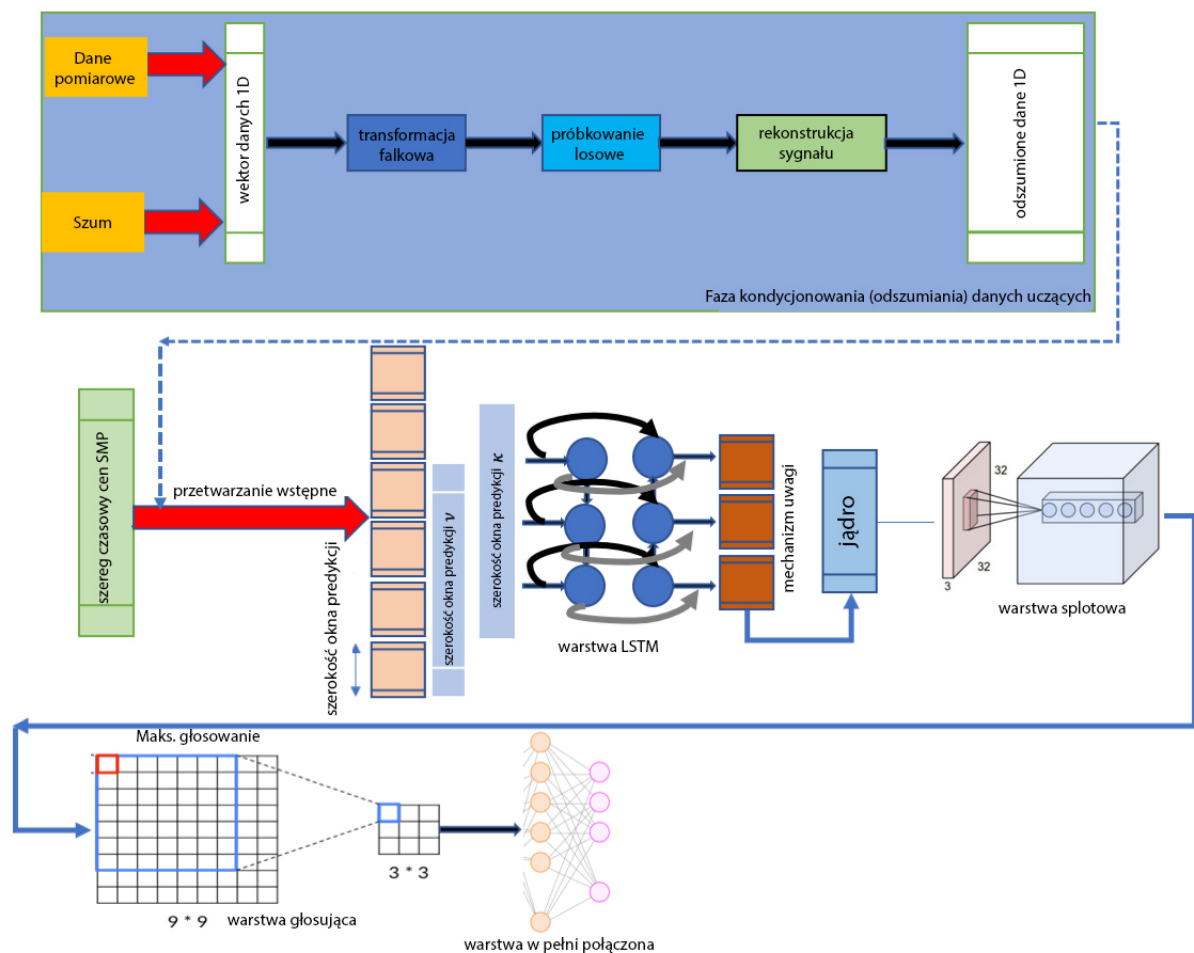
Splotowa sieć neuronowa jest przykładem sprzężonej sieci neuronowej. Można uważać liczne podobieństwa między typową typologią SSN a splotową siecią neuronową (CNN). Sieć tego typu zawiera również warstwy wejściowe, ukryte i wyjściowe. Różnica pomiędzy topologią typowej sieci neuronowej a siecią splotową polega na włączeniu do modelu tej drugiej dodatkowo warstw splotowych i łączących. Ponadto liczba ukrytych warstw w CNN jest zwykle wyższa niż w typowej SSN. Im większa liczba ukrytych warstw, tym większa zdolność wyodrębniania i rozpoznawania danych wejściowych. Sieci splotowe znalazły zastosowanie w dziedzinie przetwarzania obrazu, widzenia komputerowego, rozpoznawania cech ludzkich i klasyfikacji obrazów. Udowodniono również, jak skuteczna może być metodologia głębokiego uczenia w wielu obszarach, takich jak: rozpoznawanie wzorców, rozpoznawanie mowy i głosu, systemy wspomagające podejmowanie decyzji i in. [190, 291, 309, 210, 201, 293].

Dane dotyczące cen OMP oraz dane trendów Google mogą być traktowane jako dane wejściowe dla rekurencyjnych sieci neuronowych (RNN). Metoda uczenia zaproponowana w niniejszej pracy opiera się na następujących etapach: zbiór danych uczących jest w pierwszej kolejności odsumiany z udziałem algorytmu rekonstrukcji sygnałów z niekompletnych pomiarów (próbkowanych oszczędnie), a następnie odsumiony zbiór jest stosowany do uczenia sieci splotowej. Tak zmodyfikowany paradygmat uczenia połączony jest z procesem normalizacji, który uwzględnia fluktuacje w przedziałach cenowych zachodzące między różnymi akcjami i okresami. Połączenie tych cech okazało się kluczowym czynnikiem dla skutecznego skalowania dużych zbiorów danych.

Z danymi tymi związane są liczne zadania, m.in. prognozowanie trendów cenowych, regresja przewidywanej wartości, np. zdolność do wykrywania nieoczekiwanych zdarzeń prowadzących do skoków cen, a także zmienności akcji. Odpowiednie rozpoznanie tych aspektów jest bardzo pomocne w podejmowaniu właściwych decyzji inwestycyjnych. W praktyce, skuteczność działania tej klasy algorytmów sprowadza się zazwyczaj do poddania ich próbie danych za pomocą wyspecjalizowanych procedur, takich jak Open-High-Low-Close [305], co pozwala na utrzymanie reprezentatywnej liczby próbek w zakładanym okresie. OHLC nie zachowuje jednak wszystkich informacji o mikrostrukturze rynków.

Sieci CNN można wykorzystać do rozwiązywania problemu predykcji, ponieważ mogą one posłużyć się wszystkimi informacjami zagnieżdżonymi w danych, częściowo ze względu na ich zdolność do dokładniejszego ekstrahowania i wykorzystywania powta-

rzających się wzorców zaobserwowanych pomiędzy krokami czasowymi. Zakładając, że zbiory danych wejściowych są odsumione, mogą wówczas zostać one podane na wejścia modułów sieci pamięci długoterminowej (LSTM), co ma na celu wykorzystanie zbiorów wektorów szeregów czasowych o określonej szerokości macierzy danych wejściowych, by na koniec wytworzyć nową sekwencję kodową. W dalszej kolejności wygenerowana sekwencja kodowa uruchomi tzw. mechanizm uwagi (ang. attention mechanism) w celu poprawy jego atrybutów czasowych. Kolejny etap algorytmu obejmuje przetwarzanie przez moduł splotowych sieci neuronowych (CNN), który wydobywa cechy przestrzenne z wyjściowego zbioru danych. Liczba zmiennych lub nadmiarowych atrybutów jest zmniejszana przez warstwę maksymalnego łączenia (Max Pooling), patrz rysunek 2.2.



Rysunek 2.2: Schemat ideowy zastosowanego algorytmu predykcyjnego

W ostatnim etapie procedury w pełni połączona warstwa z liniową funkcją aktywacji generuje sygnał wyjściowy sieci, co jest równoważne z wygenerowaniem prognozy tendencji ceny OMP.

2.1 Wstępne odsumianie: kondycjonowanie danych uczących

Ze względu na złożoność natury szumu, procedury jego usuwania powinny bazować na jego dokładnym rozpoznaniu i być realizowane przy użyciu specjalistycznych technik. W niniejszej pracy przedstawiono nowatorską, ulepszoną metodę odsumiania opartą na metodologii rekonstrukcji danych próbkowanych oszczędnie. Wspomniana wcześniej rzadkość sygnału wiąże się z koniecznością aby sygnał był rzadki w jakiejś dziedzinie. Ponadto niespójność sygnału, która jest osiągnięta poprzez spełnienie właściwości izometrycznej, musi zaistnieć aby spełnić wystarczające wymagania dla rzadkich sygnałów. Niedokreślony zestaw równań liniowych ma więcej niewiadomych niż równań i generalnie ma nieskończoną liczbę rozwiązań. W ten sposób powstaje układ równań $y = Dx$, w którym chcemy znaleźć rozwiązanie dla x . Aby obliczyć rozwiązanie tego rodzaju układu, należy narzucić więcej ograniczeń lub warunków (np. gładkość). W przypadku metod próbkowania oszczędnego dodaje się ograniczenie rzadkości, akceptując tylko rozwiązania z minimalną liczbą niezerowych współczynników. Nie każdy rodzaj niedookreślonego układu równań liniowych może mieć rzadkie rozwiązanie.

Metodologia próbkowania oszczędnego bazuje na istnieniu i identyfikacji korzyści z różnego rodzaju nadmiarowości w różnych dziedzinach. Założenie to pozwala na zmniejszenie liczby współczynników, co prowadzi do przyspieszenia akwizycji pomiarów. Początkowo zaproponowano normę L2 jako główną operację wykonywaną w celu minimalizowania energii w układzie. Ta procedura, dość prosta z obliczeniowego punktu widzenia, zwykle daje słabe wyniki w większości praktycznych przypadków, dla których nieznanne współczynniki mają energię niezerową. Każda cena historycznych danych giełdowych może zawierać skończoną ilość szumu. Odszumianie metodą „pogoni za bazą” było często sugerowane [73] zamiast metodologii programowania liniowego. Ta technika odsumiania zachowuje rzadkość sygnału w obecności szumu i może być rozwiązana szybciej niż dokładny problem liniowy. W przypadku odsumiania szeregów czasowych najpierw transformujemy zaszumione szeregi czasowe do dziedziny transformacji zachowując rzadkość:

$$\phi = \Psi \times (x + z)$$

gdzie z oznacza szum addytywny. Zmienna ϕ jest próbkowana oszczędnie według schematu określonego przez macierz $M_{m \times n}$, w której M jest zmienną stałą oraz stabilną i niespójną z transformacją macierzową:

$$\Psi_y = M \times \phi = M \times \Psi \times (x + z)$$

oraz $M_{m \times n} \times \Psi_{n \times n}$ która oznacza macierz próbkowania oszczędnego A .

Stosownie do $y = A \times x$, chcemy odzyskać sygnał pierwotny (oryginalny) z sekwencji zaszumionych próbek. Procedura rekonstrukcji sygnału z sekwencji próbek (czę-

ściowo obciążonych szumem, zatem niekompletnych) odbywa się według następującego schematu:

Tabela 2.1: Odszumianie szeregu czasowego oparte na metodologii próbkowania oszczędnego.

1. Dokonaj transformacji do dziedziny przekształcenia ortogonalnego, np. przekształcenia falkowego gdzie X oznacza sygnał oryginalny a Z szum, według następującego wzoru:

$$\phi = \Psi \times (X + Z).$$

2. Z $M \times N$ wymiarowej macierzy obserwacji M która jest stabilna

a jej współczynniki nie są skorelowane z bazą transformacji Ψ , użyj M do wyznaczenia ϕ

by wygenerować wektor obserwacji $Y = M \times \phi = M \times \Psi \times (X + Z)$.

3. Zrekonstruuuj sygnał X' z sygnału Y

Procedura rekonstrukcji CS ma na celu zrekonstruować sygnał przy użyciu minimalnej liczby losowych odczytów. Ową procedurę rekonstrukcji można interpretować jako wyodrębnianie sygnału z zaszumionej obserwacji w sposób iteracyjny. W prezentowanym algorytmie obserwuje się (wykrywa) nieznaną zaszumioną szereg czasowy przez ograniczoną liczbę funkcjonałów liniowych w losowej projekcji, a następnie oryginalne szeregi czasowe są rekonstruowane przy użyciu wektora obserwacji i istniejących procedur rekonstrukcji, takich jak minimalizacja w normie L1 [168]. Poza zastosowaniami takimi, jak obrazowanie medyczne czy telekomunikacja, próbkowanie oszczędne może znaleźć zastosowanie w dziedzinie odszumiania dużych zbiorów danych. Wstępne próby wykorzystania tej metodologii w dziedzinie danych zasugerował [292, 151, 311]. Formalnie, w bazie właściwego dla danego przypadku przekształcenia, rzadkie sygnały można interpretować w następujący sposób: Wektor $X \in R^n$ jest wyrażony w bazie właściwego przekształcenia ortogonalnego

$\theta = [\theta_1, \theta_2, \dots, \theta_n]$ w następujący sposób:

$$X = \sum_{i=1}^n B_i \theta_i$$

gdzie B_i oznacza i -ty współczynnik wektora X a $B_i = \langle X, \theta_i \rangle$. W ten sposób, X składa się z θB , gdzie θ jest $n \times n$ -elementową macierzą, w której $\theta_1, \theta_2, \dots, \theta_n$ to kolumny. Rzadkość X oznacza, że w macierzy istnienie B_i zerowych elementów.

Odszumianie wymaga przeprowadzenia następujących kroków:

1. Rzadka reprezentacja.

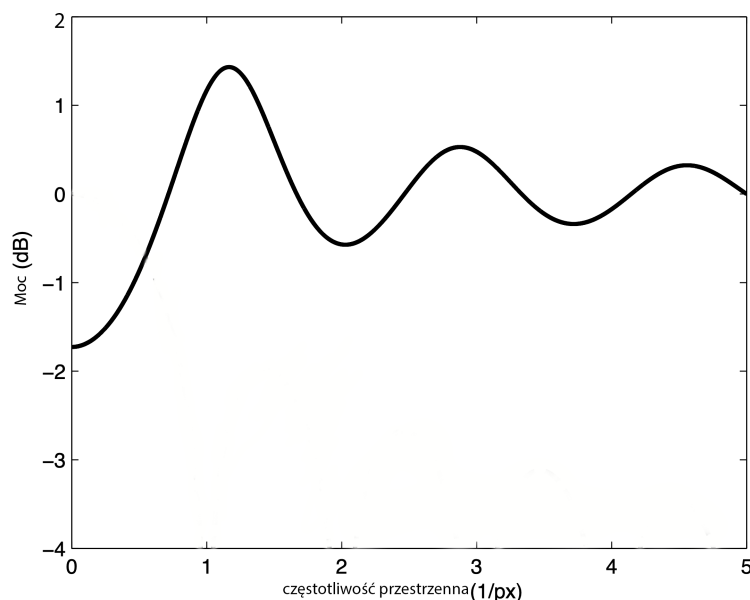
Zakładając, że $X \in R^n$ jest w rzadkiej bazie pewnego przekształcenia ortogonalnego θ , współczynniki rzadkości B mogą być wyrażone jako $B = \theta^T X$.

2. Próbkowanie losowe.

$m \times n$, ($m < n$) wymiarowa macierz obserwacji ϑ jest zdefiniowana w następujący sposób. Oznacza to, że wektor obserwacji Y jest wyrażony przy użyciu macierzy próbkowania losowego ϑ i rzadka reprezentacja sygnału jest określona w następujący sposób.

$$Y = \vartheta \theta^T X$$

W niniejszej pracy próbkowanie losowe jest realizowane według modelu próbkowania dysku Poissona, patrz rysunek 2.3 oraz 2.1.



Rysunek 2.3: Wartość średnia mocy widma procesu jednowymiarowego dysku Poissona. Proces ma związek z widmem szumu niebieskiego

Rekonstrukcja sygnału Podstawowym celem procesu rekonstrukcji jest wyodrębnienie sygnału X z sygnału Y , który jest próbkowany oszczędnie. Operacja ta przebiega według następującego schematu:

$$\min \|\theta^T X\|_0, \text{ such that, } Y = \vartheta \theta^T X.$$

Powyższe równanie nie jest odpowiednie dla trudnych problemów niedeterministycznych o wykładniczym czasie. Użycie normy L0 powoduje przedstawienie zadania jako problemu NP-zupełnego i w praktyce nie jest możliwe jego rozwiązanie w sensownym czasie. Standardowo stosowaną w tego typu zagadnieniach normą jest norma L2, jednak jej użycie nie

przynosi dobrych wyników. Norma ta "faworyzuje" istnienie wielu składowych sygnału o niewielkich wartościach, co jest sprzeczne z ideą metod próbkowania oszczędnego, gdzie konieczne jest odszukanie niewielkiej liczby składowych o istotnych wartościach. Niezbędne jest zatem zastosowanie innej normy. Z tego powodu norma L0 jest zastąpiona normą L1 według następującego schematu:

$$\min \|\theta^T X\|_0, \text{ gdzie, } Y = \vartheta\theta^T X.$$

Zakładając, że sygnał X jest obarczony szumem, zagadnienie minimalizacji powinno zostać sformułowane według poniższego schematu:

$$\min \|\theta^T X\|_0, \text{ gdzie, } \|\vartheta\theta^T X - Y\|_2 < \xi.$$

Powyższy problem optymalizacji można rozwiązać za pomocą algorytmu poszukiwania dopasowania ortogonalnego (OMP). Sygnał X jest rzadki w odpowiedniej domenie ortogonalnej. Szereg czasowy cen OMP jest przedstawiony na rysunku 2.6. Algorytmy odsumiania oparte na CS zakładają, że natywne próbki szeregów czasowych składają się z wartości trendu oraz szumu. Głównym celem metody jest wyeliminowanie szumu. Proces odsumiania zmniejsza wymiary pierwotnych szeregów czasowych. Warto zauważyć, że tradycyjne filtrowanie oparte na bazie Fouriera jest znacznie bardziej wymagające pod względem parametrów początkowych, takich jak częstotliwości graniczne pasma przepustowego oraz zaporowego. Usuwanie szumów oparte na CS jest bardziej elastyczne, ponieważ wystarczy wybrać odpowiednią ortogonalną bazę i częstotliwość próbkowania.

2.2 Przygotowanie danych wejściowych

Wstępną analizę przeprowadzono na cenach OMP. Analizie poddano dane w różnych przedziałach czasowych, jednak numeryczne wyniki umieszczone w niniejszej rozprawie wyliczono analizując dane z okresie 2018-2019 roku, podczas gdy zbiór uczący reprezentował szerszy zakres czas. Zbiór danych opublikowany przez Global Dairy Trade zawiera rekordy transakcji OMP w od 2000 do 2020 roku. Aby model prognozowania oparty na CNN działał, konieczna jest wstępna standaryzacja danych. W niniejszej pracy wykorzystano skalę standardową. W tym zakresie standardowe obliczenia skalarne wykonywane są w sposób przedstawiony poniżej:

$$S(x_i) = \frac{x_i - \frac{1}{N} \sum_{i=1}^N x_i}{\sqrt{\frac{1}{N} \sum_{i=1}^N [x_i - (\frac{1}{N} \sum_{i=1}^N x_i)]^2}}$$

gdzie x_i oznacza wartość w szeregu czasowym cen OMP w pewnym okresie czasu (np. co tydzień).

Dane wejściowe mają być okienkowane przy użyciu okna o określonej szerokości, która jest oznaczona jako α . W niniejszej pracy $\alpha = 4$, co oznacza, że mamy do czynienia z 4 wartościami na miesiąc. W ten sposób wektor $x_{t-1}, x_t, \dots, x_{t+n}$ jest przekształcany w $S(x_{t-1}, S(x_t), \dots, S(x_{t+n}))$.

2.3 Komponent LSTM

Sieci pamięci długoterminowej (LSTM) to sztuczne, powtarzające się sieci neuronowe, które zostały pierwotnie przedstawione przez Hochreitera i Schmidhubera [183], a rozszerzone przez Gravesa [291]. W topologii sieci neuronowych przedstawionej w niniejszej pracy, komponent LSTM składa się z trzech warstw: warstwy wejściowej, warstwy jednostkowej LSTM i warstwy wyjściowej, posiadających własną funkcję „od końca do końca”. Schemat działa przy użyciu β jako liczby warstw LSTM, co jest jego kontrolowanym parametrem.

Dla danej sekwencji czasowej T wektor wejściowy $S(x_{t-1}, S(x_t), \dots, S(x_{t+n}))$ zostanie wprowadzony do modułów LSTM w celu uzyskania wyczerpujących informacji z poprzednich i przyszłych kroków. Komponent LSTM składa się z trzech bramek: wejściowej, wyjściowej, zapomnianej. Bramka wejściowa $input_t$ kontroluje zakres, w jakim nowa wartość wkracza do jednostki. Bramka zapomnianej f_t decyduje o tym, które wartości mają zostać zachowane lub odrzucone. Bramka wyjściowa z kolei określa wartość sygnału wyjściowego jednostki LSTM.

Operacje te można wyrazić następująco:

$$\begin{aligned}
 input_t &= sig(U_i x_t + S_i h_{t-1} + c_i) \\
 f_t &= sig(U_f x_t + S_f h_{t-1} + c_f) \\
 output &= sig(U_o x_t + S_o h_{t-1} + c_o) \\
 f_t &= sig(U_f x_t + S_f h_{t-1} + c_f) \\
 d_t &= f_t \circ d_{t-1} + i_t \circ tanh(U_c x_t + S_c h_{t-1} + c_d) \\
 e_t &= output_t \circ tanh(d_t)
 \end{aligned}$$

gdzie x_t oznacza wektor wejściowy. X_t jest podawany na wejście jednostki LSTM, h_t jest jego wektorem wyjściowym. U, S^* i c^* są z kolei odpowiednio ważonymi macierzami i parametrami wektora odchylenia. Ponadto wartości początkowe są oznaczone jako $d_0 = 0$ i $e_0 = 0$. Operator \circ odnosi się do iloczynu elementarnego. sig i $tanh$ to funkcje sigmoidalne i hiperboliczne funkcje styżne. Produkt komponentu LSTM jest wysyłany do mechanizmu uwagi.

2.4 Mechanizm uwagi

Wynikowa sekwencja jednostki LSTM zasila mechanizm uwagi (ang. attention mechanism, MU). Ze względu na dużą wymiarowość wejściowych szeregów czasowych, wynik będzie w dużym stopniu zależny od liczby jednostek przetwarzania LSTM, które są zmniejszane. Aby skuteczniej pobierać odpowiednie informacje z ciągu kodowego oraz wyodrębnić istotne cechy czasowe i przestrzenne, do topologii sieci zostanie dodany MU. Blok ten można uznać za mechanizm walidacji rozkładu prawdopodobieństwa wag.

MU, ustawiając różne wagi prawdopodobieństwa uwagi, decyduje o dostrzeżeniu pewnych zmian lub tendencji, które mogą wystąpić w zbiorze danych uczących.

Wyraża się to następująco:

$$\begin{aligned}f_{ij} &= \tanh(W_1 e_i + W_1 e_i + b_\alpha) \\f_{ij} &= \tanh(W_1 e_i + W_1 e_i + b_\alpha) \\a_{ij} &= \text{softmax}(g_{ij}) = \frac{\exp(g_{ij})}{\sum_j (g_{ij})}, \\K_i &= \sum_j (a_{ij} k_j)\end{aligned}$$

gdzie g_{ij} wiąże i -tą wartość i j -tą wartość. W, b_α, a_{ij}, K_i odnoszą się do parametru odchylenia wagi, wartości wag uwagi i -tej i j -tej. Wartości są korygowane poprzez wykorzystanie funkcji softmax, t. determinowany jest ostateczny stan wyjścia mechanizmu uwagi.

Wyjście bloku MU trafia bezpośrednio na wejście komponentu CNN.

2.5 Komponent CNN

Jak pokazano na rysunku 2.2, struktura CNN [201] obejmuje warstwę splotową, warstwę maksymalnego łączenia i jest uczona przy użyciu danych, które zostały uprzednio odsumione. Warstwy splotu i łączenia stanowią część jednostki CNN. Ten komponent jest uczony przy użyciu odsumionych danych 1D. Zależności przestrzenne między próbkami zestawu danych uczących są kluczowymi czynnikami, które określają dalsze wyniki prognozowania. W ten sposób CNN jest wykorzystywana do identyfikacji cech morfologicznych danych pochodzących z bloku MU.

CNN składa się z następujących warstw:

1. Pojedyncza warstwa splotu. Typowe sztuczne sieci neuronowe są trudne do przystosowania ze względu na ich immanentną topologię, tj. w pełni połączone neurony. CNN są lepsze od SSN, ponieważ umożliwiają połączenie każdego pojedynczego neuronu z sąsiednimi neuronami. Tego rodzaju sieci neuronowe wymagają zainicjowania przez wiele parametrów: typy filtrów, operatory i długości filtrów oraz pewną

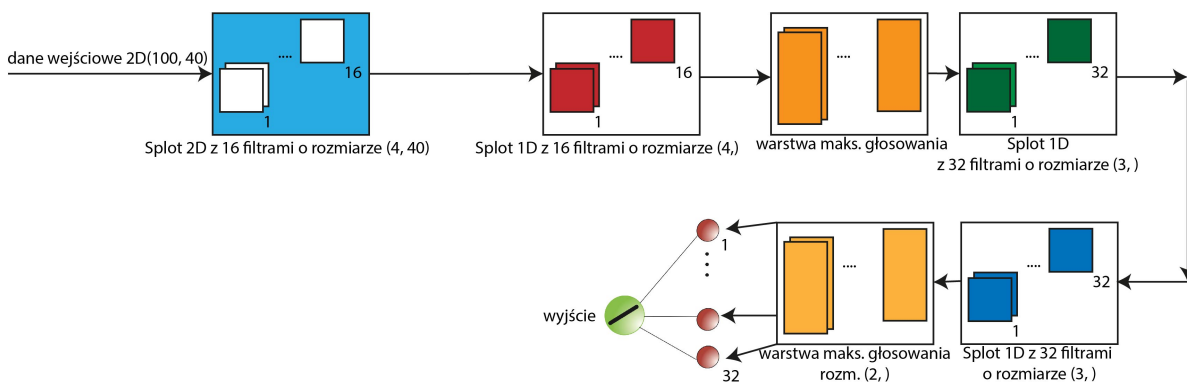
liczbę neuronów pola receptywnego. W ten sposób CNN jest trenowana przy użyciu wybranych cech, które są wykrywane i są charakterystyką problemu, przestrzenną lokalizacją i wagą opisującą dane wejściowe. Regulowane parametry sieci obejmują liczbę filtrów w warstwie splotu oznaczonej jako γ . Jego wartość jest w pełni regulowana.

2. Warstwa maksymalnego łączenia. Warstwa ta służy do zmniejszania parametryzacji topologii (np. wagi treningowe i filtry) oraz wyeliminowania niepotrzebnych funkcji. Ponadto warstwa ta może również wpływać na zbieżność sieci neuronowych. Może ona zostać użyta jako mechanizm redukcji problemu nadmiernego dopasowania. Warstwa łączenia określa maksymalną wartość spośród wartości osiągalnych przez filtr maksymalnego łączenia. W praktyce warstwa łączenia przypisuje wagi do odpowiedniego filtru, który jest wybierany na podstawie najwyższej wartości. Proces ten przebiega w następujący sposób:

$$y_j = \sum_{i=1}^n w_{i,j} x_i + b_1$$

gdzie y_j oznacza wyjście w pełni połączonej warstwy w j -tym neuronie, n odnosi się do długości danych wejściowych 1D (x), $w_{i,j}$ jest wagą neuronu między i -tą wartością wejściową a j -tym neuronem a b_1 odnosi się do odchylenia.

Po wykonaniu obliczeń wartości są wysyłane do połączonych komponentów w wyższej warstwie za pomocą funkcji aktywacji, aby zdecydować o ich wpływie na dalszą wydajność prognozowania, rys. 2.4.



Rysunek 2.4: Architektura splotowej sieci neuronowej w proponowanym rozwiązaniu

Źródło: opracowanie własne.

Funkcja aktywacji jest wyrażona w następujący sposób:

$$\overrightarrow{output}_j = l(y_j) = \max(0, y_i)$$

gdzie $\overrightarrow{output}_j$ oznacza wyjście, po którym następuje funkcja aktywacji.

Funkcję aktywacji można wyrazić za pomocą prostokątnej jednostki liniowej (ReLU) [272], która przekazuje na wyjście tylko wartości dodatnie. Tego typu funkcja okazała się skuteczna w przypadkach, gdy miało zostać usunięte zjawisko nadpasowania.

Formalnie można to wyrazić następująco:

$$\overrightarrow{output}_j = f(y_j) = \overrightarrow{output}_j.$$

Jednowymiarowe modele CNN Należy zauważyć, że pomimo iż pierwotnym obszarem zastosowań sieci splotowych były sygnały dwuwymiarowe, CNN mogą być wykorzystywane do modelowania problemów z prognozowaniem szeregów czasowych, które z naturalnych względów są przykładem danych jednowymiarowych. Jednowymiarowe szeregi czasowe to zbiory danych składające się z uporządkowanych czasowo pojedynczych obserwacji. Model natomiast musi posiadać wiedzę na podstawie serii wcześniejszych obserwacji, co w dalszej kolejności pozwoli na predykcję kolejnych wartości w sekwencji. W tej części pracy zostaną omówione aspekty związane z użyciem CNN do prognozowania cen OMP, tj. :

1. Przygotowanie danych
2. Model sieci splotowej (CNN)

Przygotowanie danych,

Zanim będzie możliwe modelowanie szeregu jednowymiarowego konieczne jest jego przygotowanie. Model CNN nauczy się funkcji, która odwzorowuje sekwencję wcześniejszych obserwacji jako dane wejściowe do obserwacji wyjściowej. W związku z tym sekwencja obserwacji musi zostać przekształcona w wiele przykładów, z których model może się nauczyć.

Rozważmy podaną sekwencję jednowymiarową:

1 [10, 20, 30, 40, 50, 60, 70, 80, 90]

Możemy podzielić sekwencję na wiele wzorców wejścia / wyjścia zwanych, wyłaniając próbki, w których trzy etapy czasowe są używane jako dane wejściowe, a jeden krok jest używany jako wynik dla jednostopniowej predykcji, której się uczy.

1	X,	y	
2	10,	20,	30 40
3	20,	30,	40 50
4	30,	40,	50 60
5	...		

Funkcja `split_sequence()` implementuje to zachowanie i podzieli daną sekwencję jednowymiarową na wiele próbek, w których każda z nich ma określoną liczbę kroków czasowych, a wyjście jest pojedynczym krokiem czasowym.

```
1 # podziel sekwencje jednoczynnikowa na probki
2 def split_sequence(sequence, n_steps):
3     X, y = list(), list()
4     for i in range(len(sequence)):
5         # znajdz koniec tego wzoru
6         end_ix = i + n_steps
7         # sprawdź, czy jesteśmy poza sekwencją
8         if end_ix > len(sequence)-1:
9             break
10    # zbierz wejściowe i wyjściowe części wzorca
11    seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
12    X.append(seq_x)
13    y.append(seq_y)
14    return array(X), array(y)
```

Funkcja ta uwzględniając zbiór testowy przedstawia się następująco:

```
1 # przygotowanie danych jednoczynnikowych
2
3 # podziel sekwencje jednoczynnikowa na probki
4 def split_sequence(sequence, n_steps):
5     X, y = list(), list()
6     for i in range(len(sequence)):
7         # find the end of this pattern
8         end_ix = i + n_steps
9         # sprawdź, czy jesteśmy poza sekwencją
10        if end_ix > len(sequence)-1:
11            break
12        # zbierz wejściowe i wyjściowe części wzorca
13        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
14        X.append(seq_x)
15        y.append(seq_y)
16        return array(X), array(y)
17
18 # zdefiniuj sekwencje wejściowa
19 raw_seq = [10, 20, 30, 40, 50, 60, 70, 80, 90]
```

```

20 # wybierz liczbe przedzialow czasowych
21 n_steps = 3
22 # podziel na probki
23 X, y = split_sequence(raw_seq, n_steps)
24 # podsumuj dane
25 for i in range(len(X)):
26     print(X[i], y[i])

```

Uruchomienie wzorca powoduje podzielenie serii jednowymiarowych na sześć próbek, przy czym każda próbka ma trzy wejściowe przedziały czasowe i jeden wyjściowy przedział czasowy.

```

1 [10 20 30] 40
2 [20 30 40] 50
3 [30 40 50] 60
4 [40 50 60] 70
5 [50 60 70] 80
6 [60 70 80] 90

```

Teraz, gdy wiemy, jak przygotować szereg jednowymiarowy do modelowania, przyjrzyjmy się metodom projektowania i przygotowania modelu CNN, który może nauczyć się mapowania danych wejściowych na wyniki.

Model sieci splotowej (CNN) Jednowymiarowy moduł CNN to model sieci splotowej, który ma spłotową warstwę ukrytą, która działa na sekwencji jednowymiarowej. W niektórych przypadkach może następować druga warstwa spłotu, na przykład bardzo długie sekwencje wejściowe, a następnie warstwa łącząca, której zadaniem jest przefiltrowanie danych wyjściowych warstwy spłotowej celem pozostawienia najbardziej istotnych elementów. Po warstwach spłotowych i łączenia znajduje się gęsta, w pełni połączona warstwa, która interpretuje cechy wyodrębnione przez spłotową część modelu. Warstwa spłaszczona jest używana między warstwami spłotowymi a warstwą gęstą, aby zredukować mapy obiektów do jednego jednowymiarowego wektora. Jednowymiarowy model CNN z jedną zmienną służący do prognozowania szeregów czasowych możemy zdefiniować w następujący sposób.

```

1 # zdefiniuj model
2 model = Sequential()
3 model.add(Conv1D(filters=64, kernel_size=2, activation='relu',
4                 , input_shape=(n_steps, n_features)))
5 model.add(MaxPooling1D(pool_size=2))
6 model.add(Flatten())

```

```

6 model.add(Dense(50, activation='relu'))
7 model.add(Dense(1))
8 model.compile(optimizer='adam', loss='MSE')

```

Kluczowy w definicji jest kształt danych wejściowych. Tego oczekuje model jako dane wejściowe dla każdej próbki, tj. odnosząc się do liczby kroków czasowych i liczby cech. Przetwarzaniu ulega seria jednowymiarowa, więc liczba cech jest równa jedności, tj. dla jednej zmiennej. Liczba kroków czasowych jako dane wejściowe, to liczba, którą wybraliśmy podczas przygotowywania naszego zbioru danych jako argument funkcji `split_sequence()`. Kształt wejściowy dla każdej próbki jest określony w `input_shape` argument dotyczący definicji pierwszej warstwy ukrytej. W praktyce, należy założyć, że liczba próbek jest znacząco większa od jedności, dlatego model będzie oczekiwał, że składnik wejściowy danych uczących będzie miał wymiary lub kształt:

```
1 [samples, timesteps, features]
```

Funkcja `split_sequence()` użyta w poprzedniej generuje wyjście w postaci X w kształcie `[samples, timesteps]`, dlatego z łatwością można je przekształcić dodając dodatkowy wymiar.

```

1 # reshape from [samples, timesteps] into [samples, timesteps,
      features]
2 n_features = 1
3 X = X.reshape((X.shape[0], X.shape[1], n_features))

```

CNN w rzeczywistości nie postrzega danych jako posiadających przedziały czasowe, zaamiast tego są traktowane jako sekwencja, na której można wykonywać splotowe operacje odczytu, tak samo, jak na obrazie jednowymiarowym.

W tym przykładzie definiujemy warstwę splotową z 64 mapami filtrów i wielkością jądra równą 2. Po niej następuje maksymalna warstwa łączenia i warstwa gęsta do interpretacji funkcji wejściowej. Określono warstwę wyjściową, która przewiduje pojedynczą wartość liczbową.

Model jest dopasowywany przy użyciu wydajnej wersji Adama stochastycznego modelu metody spadku gradientowego (lub metody gradientów sprzężonych) i jest optymalizowany przy użyciu średniej kwadratowej błędu lub funkcji straty „MSE”. Po zdefiniowaniu modelu możemy dopasować go do zestawu danych uczących.

```

1 # dopasuj model
2 model.fit(X, y, epochs=1000, verbose=0)

```

Po dopasowaniu modelu możemy go użyć do przeprowadzenia prognozy.

Możemy przewidzieć następną wartość w sekwencji, podając dane wejściowe:

```
1 [70, 80, 90]
```

Oczekujemy, że model będzie przewidywał według następującego schematu:

```
1 [100]
```

Model oczekuje, że kształt wejściowy będzie trójwymiarowy z próbkami, krokami czasu, cechami], dlatego przed wykonaniem prognozy musimy zmienić kształt pojedynczej próbki wejściowej.

```
1 # zaprezentuj predykcje
2 x_input = array([70, 80, 90])
3 x_input = x_input.reshape((1, n_steps, n_features))
4 yhat = model.predict(x_input, verbose=0)
```

Teraz możemy powiązać to wszystko razem i zademonstrować, jak opracować jednowymiarowy model CNN dla jednoczynnikowego prognozowania szeregów czasowych i dokonać pojedynczej prognozy.

```
1 # przyklad CNN dla jednej zmiennej
2 from numpy import array
3 from keras.models import Sequential
4 from keras.layers import Dense
5 from keras.layers import Flatten
6 from keras.layers.convolutional import Conv1D
7 from keras.layers.convolutional import MaxPooling1D
8
9 # podziel sekwencje jednoczynnikowa na probki
10 def split_sequence(sequence, n_steps):
11     X, y = list(), list()
12     for i in range(len(sequence)):
13         # znajdz koniec tego wzoru
14         end_ix = i + n_steps
15         # sprawdź, czy jesteśmy poza sekwencją
16         if end_ix > len(sequence)-1:
17             break
18         # zbierz wejściowe i wyjściowe części wzorca
19         seq_x, seq_y = sequence[i:end_ix], sequence[end_ix:]
20         X.append(seq_x)
21         y.append(seq_y)
22     return array(X), array(y)
23
```



```

24 # zdefiniuj sekwencje wejsciowa
25 raw_seq = [10, 20, 30, 40, 50, 60, 70, 80, 90]
26 # wybierz liczbe przedzialow czasowych
27 n_steps = 3
28 # podziel na probki
29 X, y = split_sequence(raw_seq, n_steps)
30 # zmien ksztalt z [probki, odcinki czasu] na [probki, odcinki
    czasu, funkcje]
31 n_features = 1
32 X = X.reshape((X.shape[0], X.shape[1], n_features))
33 # zdefiniuj model
34 model = Sequential()
35 model.add(Conv1D(filters=64, kernel_size=2, activation='relu',
    , input_shape=(n_steps, n_features)))
36 model.add(MaxPooling1D(pool_size=2))
37 model.add(Flatten())
38 model.add(Dense(50, activation='relu'))
39 model.add(Dense(1))
40 model.compile(optimizer='adam', loss='MSE')
41 # dopasuj model
42 model.fit(X, y, epochs=1000, verbose=0)
43 # zademonstruj predykcje
44 x_input = array([70, 80, 90])
45 x_input = x_input.reshape((1, n_steps, n_features))
46 yhat = model.predict(x_input, verbose=0)
47 print(yhat)

```

Wykonanie powyższego kodu ma na celu przygotowanie danych, dopasowanie modelu i przygotowanie prognozy. Z oczywistych względów, jakakolwiek zmiana powyżej użytych danych na inne może prowadzić, każdorazowo, do nieco różniących się od siebie wyników, co ma związek ze stochastycznym charakterem algorytmu. Z tego powodu, co przedstawiono w dalszej części pracy, wykonanie algorytmu ma charakter wielokrotny, a uzyskane wyniki są uśredniane.

2.6 Weryfikacja i testowanie skonstruowanego algorytmu predykcyjnego

W tej części pracy przedstawiono szczegóły eksperymentalnej części pracy, tj. opis typu danych, jego atrybuty oraz wskaźniki wydajności. W eksperymencie te same dane wejściowe zostały przetestowane przy użyciu kilku konkurencyjnych algorytmów prognozowania. Przedstawiona metodologia została skonfrontowana zarówno z dobrze znanymi od lat technikami (np. ARIMA [147]) jak i najnowocześniejszymi algorytmami, takimi jak: Artificial Neural Network (ANN) [272], Least Square Support Vector Regression (LSSVR) [271], ARIMA połączona z wcześniejszym odszumianiem przy pomocy metodologii próbkowania oszczędnego (CS-ARIMA), sieć neuronowa oparta na filtrowaniu Kalmana [71], wygładzanie wykładnicze [169] odszumiona sztuczna sieć neuronowa (ES-ANN), sztuczna sieć neuronowa wraz filtrowaniem Hodricka - Prescottta (HP-ANN) [141], sztuczna sieć neuronowa wraz z metodą odszumiania opartą na dyskretnej transformacji kosinusowej [131] (DCT-ANN), sztuczna sieć neuronowa wraz z metodą odszumiania opartą na dyskretnej transformacji falkowej [236] (WD -ANN), metoda maszyny wektorów podporowych o najmniejszych kwadratach wraz połączona z wcześniejszym odszumianiem przy pomocy metodologii próbkowania oszczędnego [179]. Testy przeprowadzono tak by wykazać wpływ zarówno wstępnej filtracji danych jak i modułu predykcyjnego na jakość predykcji.

Wskaźniki wydajności

Aby ocenić wydajność prognozy, przyjmowane są dwie metryki. Średnia kwadratowa błędów (RMSE) jest miarą powszechnie stosowaną w obszarze badania błędów estymacji. RMSE formalnie może zostać wyrażone w następujący sposób:

$$RMSE = \sqrt{\frac{1}{N}, \sum_{t=1}^N (\hat{x}(t) - x(t))^2},$$

gdzie: $x(t)$ oznacza bieżącą wartość, $\hat{x}(t)$ jest wartością przewidywaną, a N oznacza liczbę prognozowanych wartości w czasie t . Yao i Tan [126] opracowali wyrażoną w procentach miarę D_{stat} , której celem jest pomiar kierunkowej zmiany szeregów czasowych. Im wyższa wartość D_{stat} , tym lepiej model wychwytuje ruchy szeregów czasowych. Ową zdolność do przewidywania kierunku ruchu można oszacować za pomocą statystyki kierunkowej D_{stat} , która jest wyrażona w następujący sposób:

$$D_{stat} = \frac{1}{N}, \sum_{t=1}^N a_t \times 100,$$

gdzie: $a_t = 1$ dla $(x_{t-1} - x_t)(\hat{x}_{t+1} - x_t) \geq 0$ oraz $a_t = 0$ dla pozostałych przypadków.

Celem wyznaczenia statystycznie istotnych różnic w znaczeniu dokładności prognozowania pomiędzy różnymi technikami predykcyjnymi zastosowano statystykę Diebolda - Mariano (DM). W niniejszej pracy funkcja straty jest zastosowana w celu zilustrowania średniokwadratowego błędu predykcji (MSPE), a hipoteza zerowa opiera się na fakcie, że wartość MSPE testowanej metody nr 1 nie jest niższa niż wskazana drugą metodą. Statystyka Diebolda - Mariano (DM) jest wyrażona w następujący sposób:

$$S = \frac{\bar{g}}{(V_{\bar{g}}/M)^{\frac{1}{2}}}$$

gdzie $\bar{g} = (\sum_{t=1}^M g_t)/M$ ($g_t = (x_t - \hat{x}_{a,t})^2 - (x_t - \hat{x}_{b,t})^2$) i $\hat{x}g = \gamma_0 + 2$.

$\sum_{l=1}^{\infty} \gamma_l$ ($\gamma_l = cov(g_t, g_{t-l})$) $\hat{x}_{a,t}$ i $\hat{x}_{b,t}$, dla każdego węzła czasowego, a t , oznaczmy przewidywane wartości dla x_t .

W ten sposób test jednostronny jest stosowany do testowania statystyki S . Formalnie rzecz biorąc, wartości S oraz p mogą być użyte do oszacowania wyższości metody nr 1 nad metodą nr 2, patrz tabele poniżej.

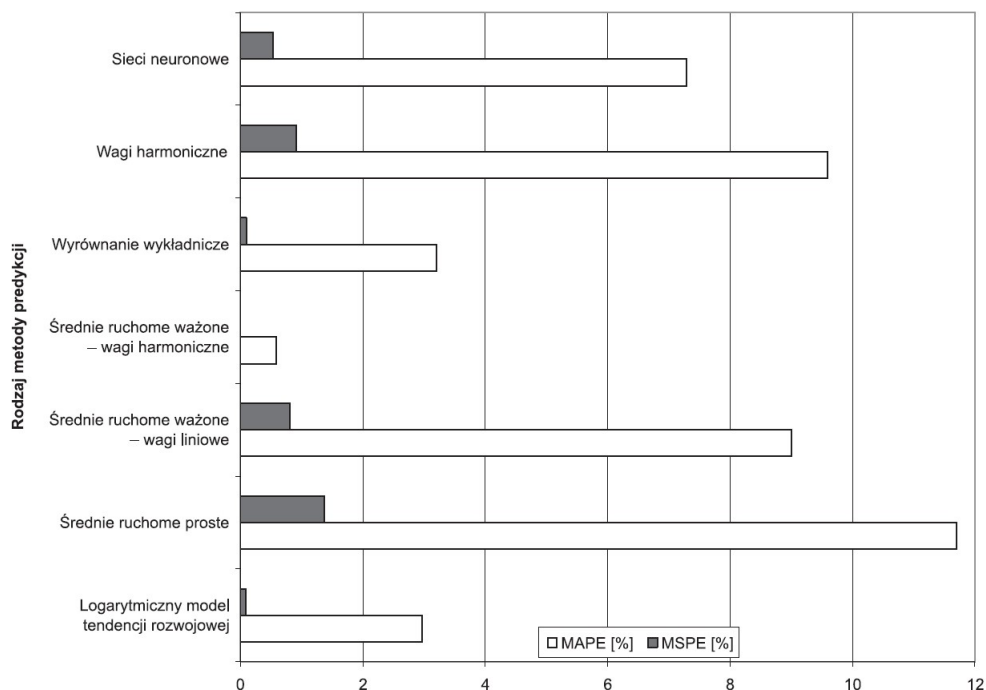
Testy wydajności

Na początek weryfikowana jest adekwatność przedstawionej metody w poprawie precyzji prognozowania. Z tego powodu kilka modeli hybrydowych (rozumianych jako połączenie odsumowania z udziałem metodologii próbkowania oszczędnego oraz dobrze znanych procedur prognozowania), np. najbardziej konwencjonalnej metody ARIMA oraz najbardziej popularnych algorytmów sztucznej inteligencji, takich jak LSSVR i ANN; oraz ich rozszerzenia oraz wersji hybrydowych, tj. Compressed- Sensing- Denoising- ARIMA, Compressed-Sensing-Denoising-Least-Square-Support-Vector-Regression and Compression-Sensing-Denoising-Artificial-Neural-Network. Ma to na celu znalezienie odpowiedzi, gdzie należy szukać przyczyn osiągnięcia określonych wyników predykcji ale też wykazanie jaka submetoda i w jakim stopniu, się do tego przyczyniła.

Główne cele wykorzystania ARIMA, LSSVR i ANN można podsumować z dwóch punktów widzenia. ARIMA może być postrzegana jako najbardziej typowy model regresji liniowej i jako metoda predykcyjna, była szeroko wykorzystywana jako punkt odniesienia w badaniach predykcyjnych. Z drugiej strony, LSSVR i ANN były powszechnie stosowane, szczególnie do przewidywania szeregów cenowych, jako najbardziej powszechne procedury sztucznej inteligencji. W związku z tym te dwa modele są tutaj wykorzystywane jako predyktory w testowanych architekturach hybrydowych. W dalszej kolejności badana jest przewaga stosowanej metody uczenia sieci, tj. Compressed-Sensing-Denoising-AI nad rozwiązaniami konkurencyjnymi. Ponadto, pięć innych, fundamentalnych technik odsumowania, w tym wygładzanie wykładnicze (ES), metoda Hodricka-Prescotta (HP), filtrowanie

Kalmana (KF), dyskretna transformata kosinusowa (DCT) i odszumianie falkowe (WD) jest użytych jako algorytmy referencyjne dla zaproponowanej metody odszumiania, co zostało zilustrowane wyznaczonymi wskaźnikami wydajności. W odszumianiu opartym na CS, jako podstawę transformacji rzadkiej zastosowano funkcję skalowania symletu 6, liczba próbek wynosiła pięćset, a liczba iteracji algorytmu poszukiwania ortogonalnego dopasowania (OMP) wynosiła 100. Współczynnik wygładzania algorytmu Exponential Smoothing wyniósł 0,2. Wartość wygładzenia Hodricka - Prescottta wyniosła 100. Filtrację Kalmana sparametryzowano następującymi wartościami: kowariancja pomiaru = 0,25; kowariancja procesu = 0,0004. Próg częstotliwości dyskretny transformaty kosinusowej wyniósł 100.

W przypadku dyskretny transformaty falkowej, symlet 6 został zaimplementowany jako podstawa falki. Poziomy rozkładu wyniosły 8, progi częstotliwości wyprowadzono na zasadzie miękkiego progu. W przypadku ARIMA parametry początkowe obliczono za pomocą Schwarz Criterion. Sieć neuronowa z wyprzedzeniem została zaimplementowana w oparciu o 7 ukrytych węzłów, 1 neuron wyjściowy i neurony wejściowe I , gdzie I oznacza kolejność opóźnień uzyskaną przez autokorelację i obliczenia korelacji częściowej i wynosiło sześć.



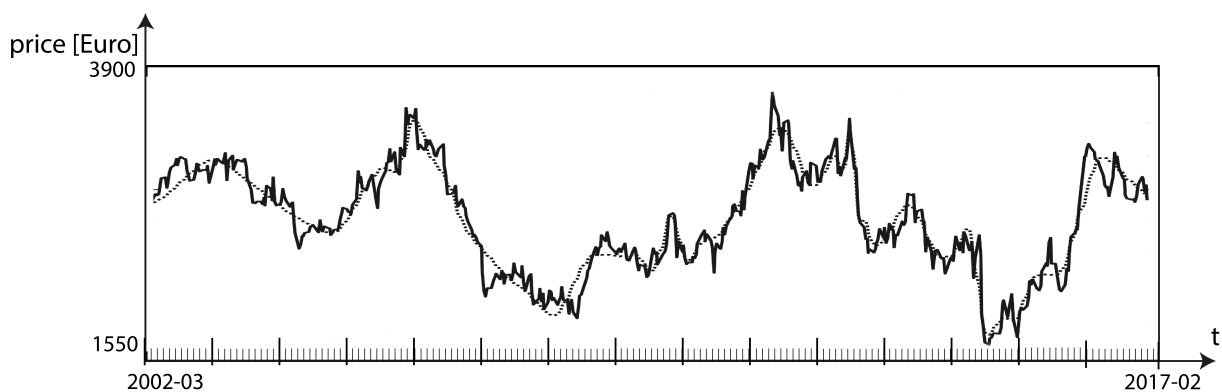
Rysunek 2.5: Kształtowanie się średniego absolutnego błędu procentowego (MAPE) oraz procentowego błędu średniokwadratowego (MSPE) dla analizowanych metod predykcji

Źródło: opracowanie własne na podstawie danych z tabeli 2.2.

Każda sztuczna sieć neuronowa została uruchomiona 10000 razy, wykorzystując podzbiór uczący. Wybrano funkcję jądra LSSVR, Gaussian RBF, i wykorzystano procedurę przeszukiwania siatki w celu uzyskania wartości parametrów γ i δ . Wszystkie próby zostały przeprowadzone w środowisku MathWorks Matlab.

2.7 Wnioski

Proponowana metoda prognozowania CNN z wbudowaną procedurą odszumiania pozwoliła na wyczyszczenie danych treningowych, patrz rysunek 2.6.



Rysunek 2.6: Odszumione dane przebiegu czasowego cen OMP (linia przerywana) oraz dane obciążone (linia ciągła)

Drugim etapem prób było prognozowanie odszumionych danych za pomocą dopasowanego algorytmu prognozowania (np. CNN, ANN, ARIMA lub LSSVR). W dalszej kolejności zweryfikowano zestaw procedur wzorcowych, w tym algorytmy samodzielne i algorytmy hybrydowe. Każdy test był powtarzany 100 razy w celu wygenerowania statystycznie wiarygodnych miar. Dla wszystkich wymienionych poniżej algorytmów prognozowania analizę statystyczną oceniono przy użyciu średniej i odchylenia standardowego (std).

Tabela 2.2: Wartości błędów predykcji dla wybranych metod prognozowania wartości cen OMP

Lp.	Rodzaj metody	Rodzaj błędu		
		MAE [tys. ton]	MAPE [%]	MSPE [%]
1	Logarytmiczny model tendencji rozwojowej	2877,62	2,82	0,09
2	średnie ruchome proste	12 349,91	12,62	1,41
3	średnie ruchome ważone - wagi liniowe	10 201,72	8,99	0,81
4	średnie ruchome ważone - wagi harmoniczne	620,58	0,57	0,00
5	Wyrównanie wykładnicze	3222,60	3,19	0,11
6	Wagi harmoniczne	8428,44	8,58	0,71
7	Sieci neuronowe	7433,21	7,26	0,51

Źródło: opracowanie własne.

W wyniku przeprowadzonych analiz testem t-studenta zauważa się, że nowo uzyskane wyniki przy użyciu autorskiego algorytmu nie różnią się istotnie od wyników uzyskanych we wcześniejszych badaniach. Wykazano, że wszystkie p wartości są mniejsze niż 0,01, co oznacza, że istnieje istotna przewaga prezentowanego rozwiązania autorskiego nad rozwiązaniami konkurencyjnymi. W pierwszej części niniejszej weryfikacji wyników porównano kilka podstawowych metod predykcyjnych z algorytmem opartym na sieciach neuronowych. Wyniki obliczeń zestawiono w tabeli 2.2, a kształtowanie się błędów procentowych *MAPE* i *MSPE* przedstawiono na rysunku 2.5. Analizując dane zamieszczone w tabeli 2.2 oraz na rysunku 2.5, można stwierdzić, że:

- najlepszą, spośród analizowanych, metod predykcji szeregu czasowego opisującego rozkład cen OMP jest metoda wykorzystująca średnie ruchome ważone z wagami harmonicznymi;
- sztuczne sieci neuronowe nie dają co prawda najlepszych wyników, bo charakteryzującą się wynikami klasyfikowanymi na czwartym miejscu wśród siedmiu testowanych metod.

Realizując drugi etap oceny celowości wykorzystania sieci neuronowej do prognozowania, opracowano tabelę 2.3 zawierającą punktową ocenę ww. metod predykcji. Analizując dane zamieszczone w tabelach 2.2 oraz 2.3, można stwierdzić, że najbardziej najlepszymi algorytmami predykcyjnymi w tym ujęciu są kolejno: średnie ruchome ważone wagami harmonicznymi, logarytmiczny model tendencji rozwojowej oraz średnie ruchome ważone wagami liniowymi. Punktacja zamieszczona w tabeli 2.3 odnosi się do uśrednionych miar MAE, MAPE oraz MSPE, które to wyliczono dla każdej z ujętych w tabeli metod. Sieci neuronowe zajmują w tym rankingu ostatnie miejsce, a to z uwagi na znaczne skomplikowanie algorytmu predykcji i w praktyce konieczność położenia dużego nacisku na selekcję

danych oraz, co zostanie przedstawione w dalszej części niniejszej pracy, dostosowanie modelu do specyfiki określonego obszaru zastosowań.

Tabela 2.3: Punktowa ocena wybranych metod predykcji

Lp.	Rodzaj metody	Ocena punktowa	Ocena punktowa 25 pkt = 100%
1	Logarytmiczny model tendencji rozwojowej	21	84%
2	średnie ruchome proste	19	76%
3	średnie ruchome ważone - wagi liniowe	20	80%
4	średnie ruchome ważone - wagi harmoniczne	23	92%
5	Wyrównanie wykładnicze	18	72%
6	Wagi harmoniczne	16	64%
7	Sieci neuronowe	16	64%

Źródło: opracowanie własne na podstawie wyników osiągniętych przez autora pracy.

W dalszej części przeprowadzoną analizę działania metod klasy the-state-of-art oraz metody proponowanej przez autora niniejszej pracy. W poniższych tabelach przedstawiono szczegółowo wyniki oceny odsumiania opartego na metodologii próbkowania oszczędnego. Ponadto, każda testowana metoda prognostyczna była uruchamiana zarówno jako pojedyncza, jak i hybrydowa, tj. z użyciem technik odsumiania. W przypadku pomiaru ilościowego obliczono również wskaźniki D_{stat} i RMSE. Tabele przedstawiają wyniki tych metryk, które zostały obliczone dla jedno- i pięciostopniowych przypadków predykcyjnych. Można zauważyć, że testowane algorytmy osiągają wyższe wskaźniki wydajności dla przypadków predykcji jednoetapowej niż predykcji pięciostopniowych.

W celu oceny dokładności predykcji kierunkowej obliczono wartości D_{stat} . Prezentowana metoda przewyższa wszystkich konkurentów, patrz tabele poniżej. Ponadto zastosowane procedury odsumiania poprawiły wskaźniki wydajności we wszystkich przypadkach. Obliczone metryki są lepsze w przypadku prognoz jedno- i pięcioetapowych. Osiągnięta poprawa jest mniej spektakularna w przypadku przewidywania pięciostopniowego niż w przypadku jednego kroku. Potwierdzono, że wszystkie metody predykcji oparte na sztucznej inteligencji przewyższają tradycyjne schematy prognozowania ARIMA. Ta metoda była zmuszona zmagać się z nieliniowościami, które często wpływają na złożone wzorce w danych o cenach OMP. Poprawa dokładności algorytmu uzyskanego przez kondycjonowanie wstępne w oparciu o CS znajduje odzwierciedlenie w wyliczonych metrykach. Wnioski są zgodne z wnioskami uzyskanymi z analiz RMSE i D_{stat} . Tabele

2.12 i 2.13 pokazują, że obliczone współczynniki RMSE dla przedstawionej metody, jak również dla wszystkich innych algorytmów hybrydowych, są niższe niż dla metod pojedynczych. Najniższą wartość uzyskano dla algorytmu przedstawionego przez autora. Najniższy współczynnik RMSE okazał się metodą CS-CNN. W porównaniu z konwencjonalnymi paradygmatami uczenia sieci, procedury głębokiego uczenia się, w połączeniu z odsumianiem (ang. Compressed Sensing Denoising, CSD), są lepsze pod względem zdolności do generalizowania. W niniejszej pracy, środek ciężkości jest położony na sieci pamięci długoterminowej (LSTM) [183], mechanizmom uwagi oraz głębokim sieciom splotowym (ang. Deep Convolutional Networks, DCN) [232]. Przewidywanie szeregów czasowych OMP jest realizowane poprzez wykorzystanie jednej ze struktur modelowania głębokiego uczenia, tzw. „Multiple Input, Multiple Output” (MIMO). Pojęcie to odnosi się do metody inicjalizacji opartej na wielowejściowych zestawach danych. W ten sposób metoda ta jest zdolna do wykorzystania wielu wektorów prognoz jako danych wyjściowych (np. prognozy OMP i masła). Metodologia CSD okazała się być idealną do rekonstrukcji sygnałów z ich rzadkiej reprezentacji. Jej zastosowanie prowadzi do uczenia z użyciem relewantnych danych uczących, co ma istotny wpływ dla modeli predykcyjnych, a tym samym poprawia wydajność algorytmu prognozowania. Ma to wyraźny związek z wykluczeniem z procesu uczenia danych obciążanych szumem i tym samym możliwości nauczenia sieci zachowania szeregów czasowych, które w rzeczywistości nie miało miejsca. Wyniki empiryczne wskazują, że topologia LSTM ma zdolność do uwzględnienia silnych nieliniowości i jest przydatna do obliczania prognoz niestacjonarnych szeregów czasowych cen OMP. Wskaźniki wydajności obliczono przy użyciu ustalonych kryteriów oceny (uśrednionej różnicy, MSE, RMSE i RMSLE) dla każdej procedury w celu oceny wydajności zarówno proponowanego przez autora niniejszej pracy rozwiązania jak i rozwiązań konkurencyjnych. Czwartą, określoną w celu i zakresie pracy, hipotezę badawczą mówiącą o niestacjonarności szeregów czasowych cen OMP zweryfikowano poprzez zastosowanie rozszerzonego testu Dickeya-Fullera. Celem weryfikacji piątej hipotezy dane poddano weryfikacji testem Kwiatkowskiego-Phillipsa-Schmidta-Shina z hipotezą zerową testu mówiącą, że analizowany szereg jest stacjonarny. Hipotezą alternatywną było stwierdzenie, że szereg nie jest stacjonarny. Poziom istotności, czyli prawdopodobieństwo, że odrzuciliśmy hipotezę zerową pomimo, że była prawdziwa, został określony na poziomie 5 %. Wartość statystyki okazała się większa od wartości krytycznej, co oznacza konieczność odrzucenia hipotezy zerowej i przyjęcie hipotezy alternatywnej wskazując tym samym, że czyli szereg nie jest stacjonarny.

W doświadczalnej części pracy wykazano, że z powodzeniem mogą być one stosowane również jako efektywne narzędzie predykcji cen surowców mleczarskich, zwłaszcza mleka w proszku. Poza tym, szczególny nacisk położono na zastosowanie, opartych na me-

metodologii tzw. próbkowania oszczędnego, metod odszumiania danych uczących. W dalszej części pracy udowodniono, że mechanizm rekonstrukcji sygnałów z sekwencji niepełnych pomiarów jest skuteczną metodą odfiltrowywania szumu z sekwencji uczących. Ponadto, wykazano również, że metody głębokiego uczenia oraz charakterystyczna dla spłotowych sieci neuronowych rozbudowana topologia uwzględniająca warstwy spłotowe oraz łączące, mechanizm uwagi osiągają przewagę nad "typowymi" sztucznymi sieciami neuronowymi, które charakteryzują się prostszą architekturą i tym samym mniejszą zdolnością do ekstrakcji oraz identyfikacji cech sygnałów, które w dalszej kolejności służą właściwej klasyfikacji, identyfikacji oraz predykcji [179], [169]. Oceny kilku metod oraz modeli prognostycznych w obszarze cen produktów mleczarskich można znaleźć w [132, 252, 144, 154, 164]. Udowodniono, że zastosowany paradygmat uczenia się schematu prognozowania CNN jest skutecznym narzędziem do poprawy wydajności predykcyjnej modelu poprzez zmniejszenie poziomu szumu obserwowanego w danych dotyczących ceny OMP. Właśnie z tych powodów ulepszona metodologia CNN opiera się wyłącznie na zasadzie struktury CS jako procedury odszumiania danych uczących. Testom poddano zarówno główne algorytmy predykcyjne jak i cząstkowe metody odszumiające. Predykcję przeprowadzono z udziałem techniki Compressed-Sensing- Denoising- AI (tj. Compressed- Sensing- Denoising-ANN i Compressed- Sensing- Denoising- LSSVR) z Compressed- Sensing- Denoising- ARIMA oraz porównując sztuczną inteligencję, sztuczne sieci neuronowe i regresję wektora wsparcia metodą najmniejszych kwadratów z autoregresyjną zintegrowaną średnią ruchomą. We wszystkich przypadkach p wartości są znacznie poniżej progu 0,1, co potwierdza, że sztuczne sieci neuronowe okazały się dokładniejsze niż zwykłe metody prognozowania, takie jak ARIMA w przypadku zbioru danych dotyczących cen OMP poniżej poziomu ufności 90%.

Tabela 2.4: Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 1 krok naprzód.

*Algorytm proponowany przez autora niniejszej pracy

Metoda predykcji z i bez wstępnego odsumiania	D_{stat}	N	M	SD	t(100)	p wartość
ARIMA	0.4342	100	0.4342	0.03	-1.495	0.160
ANN	0.4892	100	0.4892	0.02	-1.019	0.312
LSSVR	0.4765	100	0.4765	0.04	0.403	0.701
CS-ARIMA	0.5460	100	0.5460	0.02	-0.778	0.436
KF-ANN	0.6423	100	0.6423	0.02	-1.028	0.318
ES-ANN	0.6882	100	0.6882	0.03	-1.192	0.230
HP-ANN	0.6439	100	0.6439	0.02	-1.013	0.318
DCT-ANN	0.6101	100	0.6101	0.03	-1.211	0.243
WD-ANN	0.6599	100	0.6599	0.04	0.397	0.719
CS-ANN	0.6877	100	0.6877	0.02	-0.782	0.453
CS-LSSVR	0.6583	100	0.6583	0.02	-1.016	0.106
CS-CNN*	0.6911	100	0.6921	0.03	-1.191	0.039

Tabela 2.5: Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 2 kroki naprzód.

*Algorytm proponowany przez autora niniejszej pracy

Metoda predykcji z i bez wstępnego odsumiania	D_{stat}	N	M	SD	t(100)	p wartość
ARIMA	0.4366	100	0.4366	0.03	-1.377	0.161
ANN	0.4899	100	0.4899	0.02	-1.023	0.362
LSSVR	0.4701	100	0.4701	0.04	0.233	0.601
CS-ARIMA	0.5401	100	0.5401	0.02	-0.769	0.436
KF-ANN	0.6356	100	0.6356	0.01	-1.213	0.318
ES-ANN	0.6801	100	0.6801	0.02	-1.156	0.230
HP-ANN	0.6399	100	0.6399	0.02	-0.993	0.331
DCT-ANN	0.6082	100	0.6082	0.03	-1.143	0.241
WD-ANN	0.6523	100	0.6523	0.04	0.765	0.722
CS-ANN	0.6877	100	0.6877	0.02	-0.694	0.443
CS-LSSVR	0.6583	100	0.6583	0.02	-1.021	0.111
CS-CNN*	0.6744	100	0.6744	0.03	-1.013	0.029

Tabela 2.6: Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 3 kroki naprzód.

*Algorytm proponowany przez autora niniejszej pracy

Metoda predykcji z i bez wstępnego odsumiania	D_{stat}	N	M	SD	t(100)	p wartość
ARIMA	0.4323	100	0.4323	0.03	-1.219	0.161
ANN	0.4782	100	0.4782	0.03	-1.144	0.212
LSSVR	0.4655	100	0.4655	0.01	0.213	0.821
CS-ARIMA	0.5221	100	0.5221	0.02	-0.712	0.446
KF-ANN	0.6144	100	0.6144	0.01	-1.195	0.318
ES-ANN	0.6721	100	0.6721	0.02	-1.210	0.230
HP-ANN	0.6354	100	0.6354	0.03	-0.951	0.318
DCT-ANN	0.5981	100	0.5981	0.03	-1.128	0.243
WD-ANN	0.6499	100	0.6499	0.04	0.792	0.719
CS-ANN	0.6801	100	0.6801	0.02	-0.781	0.453
CS-LSSVR	0.6593	100	0.6593	0.02	-1.021	0.106
CS-CNN*	0.6744	100	0.6721	0.03	-1.123	0.031

Tabela 2.7: Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 4 kroki naprzód.

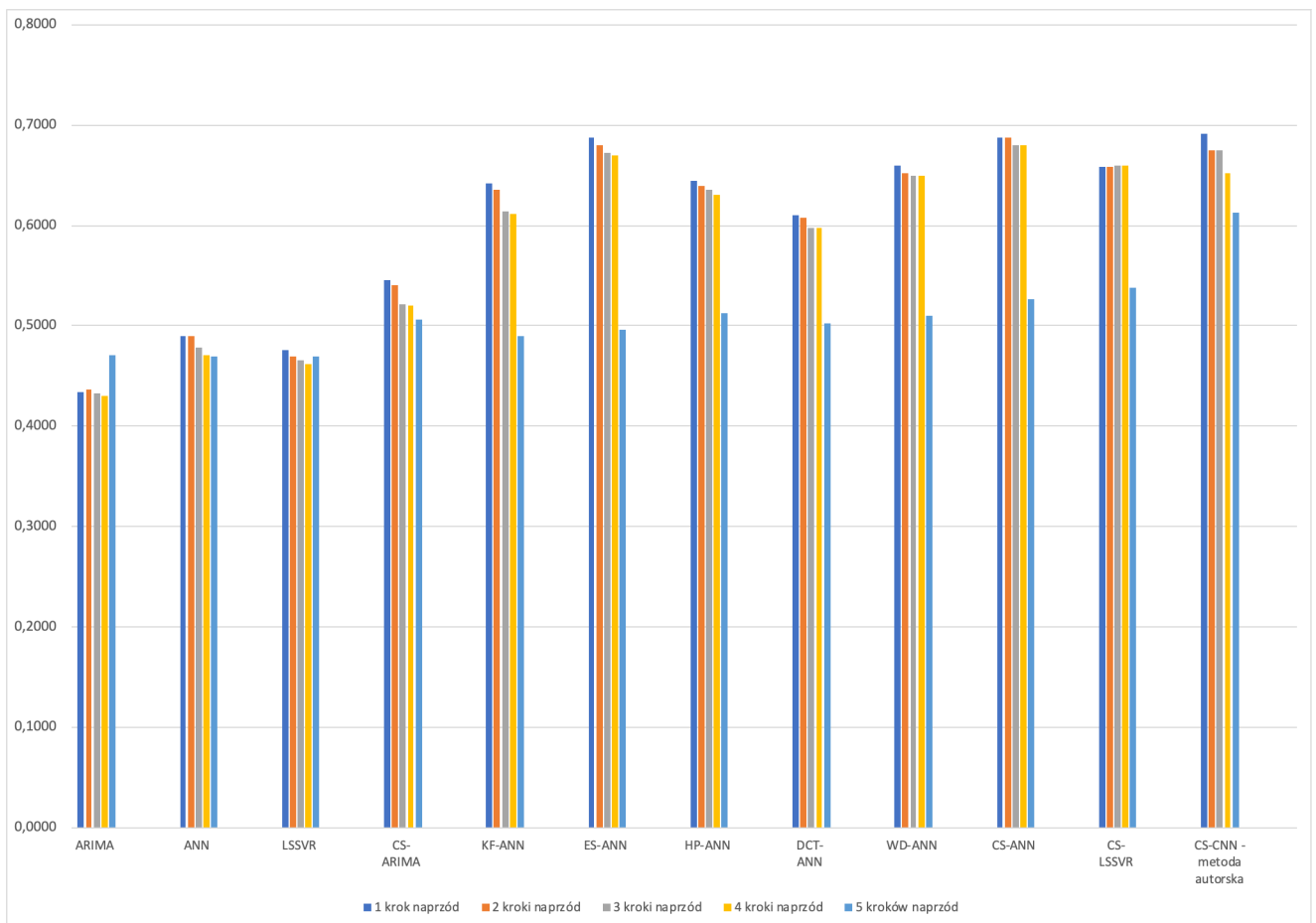
***Algorytm proponowany przez autora niniejszej pracy**

Metoda predykcji z i bez wstępnego odsumiania	D_{stat}	N	M	SD	t(100)	p wartość
ARIMA	0.4300	100	0.4300	0.03	-1.199	0.172
ANN	0.4710	100	0.4710	0.03	-1.141	0.232
LSSVR	0.4621	100	0.4621	0.01	0.115	0.798
CS-ARIMA	0.5200	100	0.5200	0.02	-0.734	0.446
KF-ANN	0.6121	100	0.6121	0.01	-1.234	0.318
ES-ANN	0.6702	100	0.6702	0.02	-1.191	0.194
HP-ANN	0.6301	100	0.6301	0.03	-0.348	0.347
DCT-ANN	0.5981	100	0.5981	0.03	-1.128	0.241
WD-ANN	0.6499	100	0.6499	0.04	0.792	0.716
CS-ANN	0.6801	100	0.6801	0.02	-0.781	0.451
CS-LSSVR	0.6593	100	0.6593	0.02	-1.021	0.75
CS-CNN*	0.6521	100	0.6521	0.01	-1.921	0.029

Tabela 2.8: Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 5 kroków naprzód.

*Algorytm proponowany przez autora niniejszej pracy

Metoda predykcji z i bez wstępnego odsumiania	D_{stat}	N	M	SD	t(100)	p wartość
ARIMA	0.4711	100	0.4711	0.03	-1.648	0.127
ANN	0.4695	100	0.4695	0.03	-1.389	0.169
LSSVR	0.4701	100	0.4701	0.03	-1.432	0.181
CS-ARIMA	0.5061	100	0.5061	0.03	-1.658	0.103
KF-ANN	0.4899	100	0.4899	0.03	-1.180	0.259
ES-ANN	0.4956	100	0.4956	0.03	-1.138	0.271
HP-ANN	0.5122	100	0.5122	0.02	-1.490	0.164
DCT-ANN	0.5023	100	0.5023	0.03	-1.463	0.148
WD-ANN	0.5103	100	0.5103	0.04	-1.234	0.215
CS-ANN	0.5263	100	0.5263	0.03	-1.069	0.309
CS-LSSVR	0.5381	100	0.5381	0.03	-1.211	0.238
CS-CNN*	0.6126	100	0.6126	0.02	-1.236	0.096



Rysunek 2.7: Wartość D_{stat} dla poddanych testom algorytmów predykcji od 1 do 5 kroków naprzód

Tabela 2.9: Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 1 krok naprzód.

***Algorytm proponowany przez autora niniejszej pracy**

Metoda predykcji z i bez wstępnego odsumiania	RMSE	N	M	SD	t(100)	p wartość
ARIMA	1.6450	100	1.6450	0.03	-1.641	0.110
ANN	1.1731	100	1.1731	0.03	-1.403	0.189
LSSVR	1.1433	100	1.1433	0.03	-1.438	0.158
CS-ARIMA	1.5122	100	1.5122	0.01	-1.655	0.124
KF-ANN	1.2287	100	1.2287	0.03	-1.196	0.236
ES-ANN	1.2887	100	1.2887	0.03	-1.150	0.250
HP-ANN	1.3543	100	1.3543	0.03	-1.488	0.145
DCT-ANN	1.3210	100	1.3210	0.04	-1.481	0.170
WD-ANN	1.2132	100	1.2132	0.02	-1.231	0.213
CS-ANN	1.1193	100	1.1193	0.03	-1.067	0.290
CS-LSSVR	1.1999	100	1.1999	0.03	-1.223	0.230
CS-CNN*	1.0544	100	1.0544	0.03	-1.220	0.087

Tabela 2.10: Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 2 kroki naprzód.

***Algorytm proponowany przez autora niniejszej pracy**

Metoda predykcji z i bez wstępnego odsumiania	RMSE	N	M	SD	t(100)	p wartość
ARIMA	1.6912	100	1.6912	0.03	-1.201	0.112
ANN	1.1923	100	1.1923	0.01	-1.293	0.171
LSSVR	1.1621	100	1.1621	0.02	-0.948	0.148
CS-ARIMA	1.7232	100	1.7232	0.03	-1.310	0.119
KF-ANN	1.5283	100	1.5283	0.03	-1.176	0.219
ES-ANN	1.3888	100	1.3888	0.03	-1.132	0.211
HP-ANN	1.4544	100	1.4544	0.03	-1.461	0.122
DCT-ANN	1.4214	100	1.4214	0.04	-1.477	0.143
WD-ANN	1.3131	100	1.3131	0.02	-1.233	0.265
CS-ANN	1.2191	100	1.2191	0.01	-1.062	0.265
CS-LSSVR	1.2003	100	1.2003	0.04	-1.203	0.235
CS-CNN*	1.0544	100	1.0514	0.03	-1.219	0.087

Tabela 2.11: Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 3 kroki naprzód.

***Algorytm proponowany przez autora niniejszej pracy**

Metoda predykcji z i bez wstępnego odsumiania	RMSE	N	M	SD	t(100)	p wartość
ARIMA	1.7133	100	1.7133	0.02	-1.152	0.121
ANN	1.2311	100	1.2311	0.01	-1.101	0.165
LSSVR	1.2211	100	1.2211	0.02	-0.921	0.129
CS-ARIMA	1.8112	100	1.8112	0.03	-1.310	0.122
KF-ANN	1.6411	100	1.6411	0.03	-1.176	0.221
ES-ANN	1.5912	100	1.5912	0.03	-1.132	0.198
HP-ANN	1.6123	100	1.6123	0.03	-1.461	0.154
DCT-ANN	1.5512	100	1.5512	0.02	-1.477	0.139
WD-ANN	1.5623	100	1.5623	0.01	-1.233	0.232
CS-ANN	1.4488	100	1.4488	0.03	-1.062	0.212
CS-LSSVR	1.5223	100	1.5223	0.02	-1.203	0.236
CS-CNN*	1.1021	100	1.1021	0.03	-1.028	0.074

Tabela 2.12: Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 4 kroki naprzód.

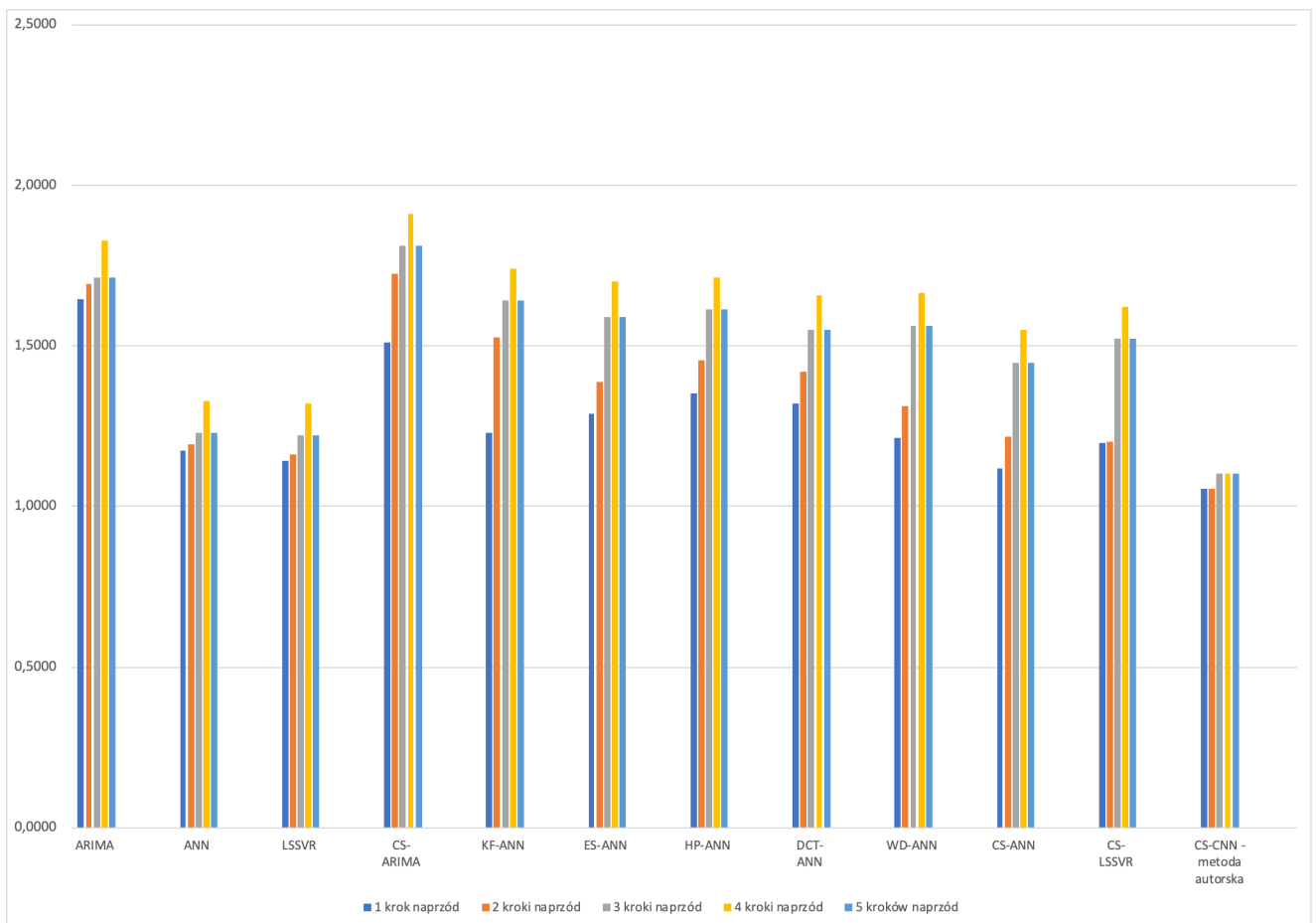
***Algorytm proponowany przez autora niniejszej pracy**

Metoda predykcji z i bez wstępnego odsumiania	RMSE	N	M	SD	t(100)	p wartość
ARIMA	1.8266	100	1.8266	0.02	-1.142	0.110
ANN	1.3287	100	1.3287	0.01	-1.101	0.141
LSSVR	1.3214	100	1.3214	0.02	-0.921	0.134
CS-ARIMA	1.9126	100	1.9126	0.03	-1.310	0.188
KF-ANN	1.7420	100	1.7420	0.03	-1.176	0.277
ES-ANN	1.6999	100	1.6999	0.03	-1.132	0.173
HP-ANN	1.7141	100	1.7141	0.03	-1.461	0.123
DCT-ANN	1.6563	100	1.6563	0.02	-1.477	0.128
WD-ANN	1.6647	100	1.6647	0.01	-1.233	0.274
CS-ANN	1.5491	100	1.5491	0.03	-1.062	0.219
CS-LSSVR	1.6210	100	1.6210	0.02	-1.203	0.210
CS-CNN*	1.1012	100	1.1061	0.03	-1.028	0.074

Tabela 2.13: Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 5 kroków do przodu.

***Algorytm proponowany przez autora niniejszej pracy**

Metoda predykcji z i bez wstępnego odsumiania	RMSE	N	M	SD	t(100)	p wartość
ARIMA	2.7899	100	2.7899	0.03	-1.629	0.119
ANN	2.7722	100	2.7722	0.03	-1.392	0.189
LSSVR	2.7891	100	2.7891	0.02	-1.426	0.158
CS-ARIMA	2.6732	100	2.6732	0.04	-1.675	0.101
KF-ANN	2.6427	100	2.6427	0.03	-1.184	0.257
ES-ANN	2.6611	100	2.6611	0.03	-1.155	0.253
HP-ANN	2.6373	100	2.6373	0.03	-1.501	0.156
DCT-ANN	2.6431	100	2.6431	0.03	-1.469	0.154
WD-ANN	2.6199	100	2.6199	0.03	-1.243	0.234
CS-ANN	2.6540	100	2.6540	0.02	-1.073	0.288
CS-LSSVR	2.6770	100	2.6770	0.03	-1.216	0.235
CS-CNN*	2.1699	100	2.1699	0.03	-1.222	0.094



Rysunek 2.8: Wartość RMSE dla poddanych testom algorytmów predykcji od 1 do 5 kroków naprzód

Streszczenie pracy

Głównym celem pracy było zaprojektowanie, implementacja oraz wykazanie przewagi nad rozwiązaniami konkurencyjnymi nowego hybrydowego algorytmu służącego do przewidywania cen odtłuszczonego mleka w proszku. Ponadto, za cel postawiono sobie przedstawienie teoretycznego podłoża rynków rolnych, a w szczególności rynku mleka. Szeregi czasowe cen odtłuszczonego mleka w proszku cechują się nieliniowością, niestacjonarnością oraz dużą zmiennością dynamiczną mającą swe podłoże w zjawiskach gospodarczych, które są charakterystyczne dla specyficznego rynku rolnego, jakim jest rynek mleka. Połączenie analizy technicznej, sztucznych sieci neuronowych pozwoliły na osiągnięcie dokładniejszej predykcji od wcześniej stosowanych metod obliczeniowych i w dalszej kolejności, zaprojektowany algorytm może posłużyć jako wyspecjalizowane narzędzie, które jego użytkownikom może zapewnić osiągnięcie stabilniejszej sytuacji ekonomicznej i docelowo, większych zysków.

Cena odtłuszczonego mleka w proszku zależy zasadniczo od podaży i popytu. Wśród czynników mających kluczowy wpływ na dynamikę zmian krzywych popytu i podaży wymienić można zarówno czynniki ekonomiczne takie, jak wzrost gospodarczy, koniunktura oraz siła nabywcza pieniądza, jak i mechanizmy charakterystyczne dla rynków rolnych, w tym silne mechanizmy interwencyjne. Podejmowane przez producentów rolnych decyzje inwestycyjne bardzo często nie są poddane wcześniejszej analizie ryzyka ekonomicznego, a są wynikiem obserwowanego przez nich chwilowego ożywienia gospodarczego w obszarze swojej aktywności. Aspekty te determinują licznosc pogłowia krów mlecznych. Czynniki te w dużej mierze prowadzą do zachwiania naturalnych zasad konkurencji wolnorynkowej i w istocie mają wpływ zarówno na podaż jak i późniejszy popyt. Pozostałymi czynnikami cenotwórczymi w tym obszarze są aspekty polityczne, tj. przepisy, ograniczenia albo preferencje celne, embarga, ale też czynniki psychologiczne (moda na określone produkty, zmiany trendów konsumenckich), wywoływanie nacisku przez organizacje ekologiczne, których celem może być zniechęcenie do konsumpcji produktów pochodzenia zwierzęcego, w tym mleka i jego przetworów. Istotną presję na wartość rynkową surowców mleczarskich wywołują również zdarzenia nieregularne, tj. występowanie czynników nagłych, jak np. epidemie, choroby bydła mlecznego. Czynniki te prowadzą do dużej zmienności rynku mleka i masła, a szeregi czasowe ich cen charakteryzują się złożoną nieliniowością, zmiennością dynamiczną i wysoką nieregularnością. Z kolei fluktuacje cen tych produktów mają istotny wpływ na gospodarkę żywnościową i społeczeństwo, a te czynniki należy uwzględnić w planowaniu polityki i reguł działania podmiotów gospodarczych, co w dalszej kolejności jest czynnikiem zapewniającym stabilny rozwój gospodarczy i społeczny. Najistotniejszymi czynnikami determinującymi cenę mleka w

skupie jest, zmienna w czasie, wartość rynkowa jego składników, tj. tłuszczu i białka. Istotność tych komponentów, jako surowców produkcyjnych, oddziałuje zarówno na globalne gospodarki jak i rynek żywności oraz przetwórców żywności. Pierwszą część pracy poświęcono analizie teoretycznej istoty i uwarunkowania kształtowania cen na rynku mleka. W szczególności uwagę poświęcono aspektom związanym z rynkowym mechanizmem kształtowania cen wskazując przy tym na istnienie, charakterystycznych dla rynku mleka osobliwości go determinujących. Uwarunkowania związane z prowadzeniem działalności gospodarczej w obszarze rolnictwa stanowią istotne czynniki kształtujące specyfikę rynków rolnych, a co z tym idzie, model klasycznej teorii cen, nie znajduje tutaj zastosowania. Zmienność skali produkcji zachodząca pod wpływem rozmaitych czynników jest cechą charakterystyczną dla tego rynku i zachodzi zarówno w aspekcie kierunkowym jak i wielkościowym (ilościowym). Tematem dalszej części pracy są aspekty związane z produkcją, przetwórstwem oraz handlem krajowym i międzynarodowym. Aspekty te stwarzają bowiem obiektywne trudności dla konwencjonalnych algorytmów predykcyjnych, wskazując tym samym na konieczność zastosowania metod o charakterze samo-organizującym. W ten sposób zidentyfikowano obiektywne trudności stanowiące przeszkody dla do tej pory stosowanych algorytmów predykcyjnych. Uzasadniono tym samym prawdziwość hipotezy badawczej stanowiącej o możliwości konstrukcji systemów prognozowania cen surowców mleczarskich przy pomocy algorytmów sztucznej inteligencji w agrobiznesie, w szczególności uwzględniając potrzebę zastosowania nowatorskiego podejścia do wyeliminowania immanentnego szumu obciążającego próbki szeregów czasowych cen mleka w proszku. Uzasadniono również konieczność zastosowania sieci głębokich oraz zmodyfikowanego paradygmatu uczenia głębokiego. Dzięki zastosowaniu technik optymalizacji algorytmów sztucznej inteligencji poprzez zastosowanie odszumiania danych trenujących osiągnięto poprawę działania również w zakresie predykcji w porównaniu z konwencjonalnymi sieciami splotowymi, które stanowiły prototypowy model predyktora użytego w części konstrukcyjnej pracy. Na uwagę zasługuje fakt, że po raz pierwszy w obszarze predykcji cen zastosowano splotowe (konwolucyjne) sieci neuronowe, dotychczas stosowane jedynie w obszarze klasyfikacji obrazów i dźwięków. Dzięki zastosowaniu koncepcji sieci splotowych wraz z wstępnym odszumianiem danych uczących wyeliminowano ograniczenia konwencjonalnych systemów predykcyjnych.

Istotną część pracy poświęcono również narzędziom analizy szeregów czasowych, by na jej koniec starać się przybliżyć problematykę zarówno sieci neuronowych jak i filtracji cyfrowej, które to znalazły zastosowanie w kolejnym rozdziale. Kolejna część niniejszej pracy ma charakter konstrukcyjno-doświadczalny i jest próbą opisu i scharakteryzowania, proponowanego przez autora niniejszej pracy, własnego algorytmu predykcyjnego. W części konstrukcyjnej pracy przedstawiono szczegóły zaprojektowanego i zaimplementowanego

w środowisku Matlab Mathworks algorytmu predykcyjnego. W rozdziale trzecim pracy przedstawiono badania wyniki badań doświadczalnych, prezentując jednocześnie zarówno użyte w pracy miary oceny wydajności algorytmów predykcji, jak i porównano przy ich użyciu proponowany algorytm z czołowymi rozwiązaniami światowymi w zakresie predykcji szeregów czasowych.

Główne etapy analityczne pracy obejmowały:

1. scharakteryzowanie rynku mleka,
2. identyfikację czynników wpływu na zmienność cen odtłuszczonego mleka w proszku,
3. analizę algorytmów predykcji szeregów czasowych cen,
4. identyfikację czynników ograniczających dokładność algorytmów predykcji,
5. optymalizację algorytmów sztucznej inteligencji poprzez stosowanie odsumiania danych trenujących,
6. optymalizację algorytmów sztucznej inteligencji poprzez zastosowanie metod głębokiego uczenia
7. opracowanie hybrydowego algorytmu do przewidywania cen mleka w proszku z zastosowaniem wskaźników analizy technicznej oraz splotowych SSN,
8. opracowanie nowego paradygmatu uczenia głębokiego przy użyciu metod rekonstrukcji sygnałów próbkowanych oszczędnie,
9. zaprojektowanie i zbudowanie eksperckiego systemu informatycznego do ograniczenia ryzyka inwestycyjnego oraz ułatwienia inwestorom podejmowania decyzji strategicznych w przedsiębiorstwach wytwarzających, sprzedających oraz przetwarzających jako półprodukt odtłuszczone mleko w proszku;
10. użycie sieci splotowych jako narzędzia predykcyjnego oraz nowe podejście do odsumiania zbiorów danych, tj. stworzenie nowego paradygmatu głębokiego uczenia SSN poprzez wykorzystanie opartych na metodologii próbkowania oszczędnego (ang. compressed sensing, CS) procedur odsumiania.

Osiągnięte wyniki pozwalają na stwierdzenie, że istnieje możliwość adaptacji niestosowanych do tej pory metod modele splotowych sieci neuronowych w predykcji cen na rynku OMP. Wykazano, że modele splotowych sieci neuronowych można zastosować do prognozowania szeregów czasowych OMP. W przeciwieństwie do innych metod uczenia maszynowego, sieci CNN są zdolne do obsługi nienadzorowanych funkcji uczenia się na

podstawie danych sekwencyjnych. Ponadto, udowodniono, że są one zdolne do analizowania danych wielowymiarowych i mogą bezpośrednio tworzyć wektory, które są użyteczne do prognozowania wieloetapowego.

Sieci splotowe w ujęciu pojedynczym, tj. jako pojedyncza technika (bez wstępnej filtracji) działały dobrze, a osiągnięte z ich udziałem wyniki przewyższały zarówno modele konwencjonalne jak i tradycyjne sieci neuronowe. Jak wykazano w części doświadczalnej pracy, algorytmy głębokiego uczenia się, w połączeniu z odsumianiem CSD, są lepsze od metod ANN pod względem zdolności do generalizowania. Warto podkreślić, że prognozowanie z użyciem tradycyjnych sieci neuronowych nie prowadzi do lepszych wyników wyrażonych miarami MAE, MAPE, czy MSPE niż z użyciem metody średnich ruchomych, ważonych – wag harmonicznym oraz wyrównania wykładniczego. Zastosowane autorskie hybrydowe algorytmy predykcji pozwoliły na rozwiązanie problemów predykcyjnych, które należą do zakresu tzw. trudnych problemów z przewidywaniem sekwencji. Wykazano również, że z uwagi na ścisłą zależność skuteczności działania algorytmów uczenia maszynowego od trafności zestawów szkoleniowych CNN, włączony algorytm odsumiania pozwolił na osiągnięcie lepszych wyników. Metodologia odsumiania danych uczących czerpiąca z idei rekonstrukcji sygnałów rzadkich może skutecznie zwiększyć zdolność prognozowania, ponieważ wszystkie testowane algorytmy predykcyjne oparte na odsumianiu CS osiągnęły lepsze wyniki niż ich samodzielne odpowiedniki, tj. z pominięciem filtrowania oparte na CS. Warto podkreślić, że zadowalające rezultaty osiągnięto dzięki zastosowaniu wszystkich testowanych powyżej modele hybrydowych. Przedstawiona metoda predykcji wskazała na jej skuteczność w wyrażaniu nieliniowych wzorców ukrytych w cenie OMP. Zaproponowany hybrydowy algorytm osiągnął najlepsze wyniki zarówno pod względem poziomu, jak i pod względem prognoz kierunkowych, wskazując na jego wszechstronność i stabilność.

Liczne zmienne czynniki prowadzące do dużej zmienności rynku mleka i masła i złożonej nieliniowości, zmienności dynamicznej i wysokiej nieregularności sprawiły obiektywne trudności dla większości testowanych algorytmów predykcyjnych uzasadniając tym samym konieczność zastosowania prezentowanego przez autora niniejszej pracy rozwiązania. Wykazano również, że z uwagi na niemożność modelowania wartości szeregów czasowych cen przez liniowe modele predykcyjne oraz aktywny, wciąż zmieniający się w swej naturze i dynamice wpływ czynników zewnętrznych, konieczne było zastosowania odmiennych mechanizmów opisujących dynamikę zmian oraz pozwalających na wiarygodne wyznaczenie przyszłych wartości, na podstawie przeszłych wartości z uwzględnieniem dotychczas pomijanych, zewnętrznych czynników wpływu. Udowodniono również przewagę sztucznych sieci neuronowe, które nabywają zdolność przewidywania w wyniku procesu uczenia nad rozwiązaniami konwencjonalnymi. Sieci neuronowe pozwoliły na

możliwość budowy modeli odwzorowujących złożone zależności pomiędzy danymi wejściowymi i wyjściowymi dla zjawisk, których struktura, prawa działania czy zależności przyczynowe nie zostały poznane w stopniu wystarczającym na zaprojektowanie efektywnych modeli matematycznych. W części doświadczalnej zaobserwowano również, że sieci neuronowe (zarówno tradycyjne jak i splotowe) są zdolne do odkrywania w zbiorze danych związków niewykrywalnych nawet przez tradycyjne metody statystyczne. Rozbudowana, w stosunku do tradycyjnych sieci neuronowych, topologia sieci splotowych pozwoliła na bardziej szczegółową ekstrakcję cech sygnałów i w dalszej kolejności lepsze wyniki predykcji. Wykazano również, że nie bez znaczenia jest relewantność danych uczących oraz ich odszumienie. Źródłami szumu są czynniki prowadzące do pojawienia się w szeregu czasowym wartości, które prowadzą do krótkookresowych skoków wartości, co utrudnia działanie sieciom neuronowym i w dalszej kolejności predykcję. Z powodu występowania szumu w wektorach danych, odszumianie szeregów czasowych cen jest czynnością konieczną z uwagi na wymagania stawiane przez algorytmy predykcyjne wobec „czystości” i „relewantności” danych wejściowych.

Autor pracy wyraża nadzieję, że opracowana przez niego metodologia będzie w przyszłości podstawa do zaoferowania gotowego narzędzia predykcyjnego, które stanie się efektywnym narzędziem inwestorskim i będzie prowadzić do ograniczenia ryzyka finansowego [296], [7]. Ostatnie wydarzenia na arenie międzynarodowej, tj. wybuch pandemii COVID-19 oraz agresja zbrojna Rosji na Ukrainie spowodowały konieczność zmian w polityce planowania ale również zastosowania nowych metod analizy procesów zachodzących na rynkach gospodarczych. Wywołane tymi czynnikami gwałtowne zmiany i duża dynamika czynników cenotwórczych oraz cen zmuszają planistów i decydentów do stosowania najnowocześniejszych technologii predykcji oraz analizy rynków [82]. Ostatnie kilkanaście miesięcy wskazuje na wzrost zainteresowania środowiska naukowego i inwestorskiego metodami uczenia maszynowego, a w szczególności technikami głębokiego uczenia. Pandemia COVID-19 miała drastyczny wpływ na światową gospodarkę oraz rynki akcji, do czego przyczyniły się różne poziomy zarządzanych blokad warunkowanych zmienną liczbą dziennych zakażeń [115]. Zdarzenia te z powodu ich nieregularności miały i mają wpływ na rynki akcji. Oznacza to, że stosowanie metod głębokiego uczenia oraz sieci neuronowych do wieloetapowego prognozowania cen akcji przed i podczas COVID-19 zyskuje na znaczeniu i przewadze nad rozwiązaniami konkurencyjnymi. Autor pracy dostrzega konieczność rozszerzenia zakresu swoich badań naukowych na inne obszary gospodarki. Istnieje bowiem możliwość rozszerzenia tematyki zawartej w niniejszej pracy do analizy innych szeregów czasowych cen, np. predykcji cen: masła, oleju napędowego, prawa do emisji dwutlenku węgla i innych, które cechują się dużą nieprzewidywalnością oraz złożonością [275], [236], [179], [169], [277], [287].

Bibliografia

Pozycje zwarte:

- [1] Augustyńska-Grzymek I, Goraj L., Jarka S, Pokrzywa T., Skarżyńska A., *Metodyka liczenia nadwyżki bezpośredniej i zasady klasyfikacji gospodarstw rolniczych*, *FAPA*, Warszawa, **2000**.
- [2] Baczyński, D., Parol M., Piotrowski P. *Sztuczna Inteligencja w praktyce. Laboratorium*, *Oficyna Wydawnicza Politechniki Warszawskiej*, Warszawa, **2006**.
- [3] Baumol W.J. (red.), *Ekonomia polityczna kapitalizmu*, *PWN*, Warszawa, **1969**.
- [4] Becker G.A., Theory of the Allocation of Time, *Economic Journal*, **1965**.
- [5] Begg D., Fischer S., Dornbusch R., *Economics*, *Mc Graw – Hill*, London, **1984**.
- [6] Begg D., Fischer S., Dornbusch R., *Mikroekonomia*, *PWE*, Warszawa, s. 196-204, **2003**.
- [7] Biegański, M. *Hedging i nowoczesne usługi finansowe*. *Wyd. AE*, Poznań, **2001**.
- [8] Binswanger H. Ch., *Spirala wzrostu, pieniądz, energia i kreatywność w dynamice procesów rynkowych*, *Wyd. ZYSK i S-KA*, Poznań, **2011**.
- [9] Blaug M. *Teoria ekonomii*, *PWN*, Warszawa, **1994**.
- [10] Boczar K., Szelażek T., Wala F., *Spółdzielczość wiejska w okresie przełomu*, *Fundacja F. Eberta w Polsce*, Warszawa, **1993**.
- [11] Bowden E. V., Bowden J. H., *Ekonomia*, *Wyd. Fundacja Innowacja*, Warszawa, **2002**.
- [12] Bożyk P., Misala J., Puławski M., *Międzynarodowe stosunki ekonomiczne*, *wyd.2 PWE*, Warszawa, **2002**.
- [13] Brockwell, P., Davis R., *Introduction to Time Series and Forecasting*, Second Edition, *Springer-Verlag New York, Inc.*, **2002**.
- [14] Brodziński M. G., *Spółdzielczość obsługująca wieś i rolnictwo w okresie przekształceń ustrojowych*, *Wyższa Szkoła Ekonomiczna*, Warszawa, **2005**.
- [15] Caban W. (red.), *Ekonomia*, t. II, *Podstawy makroekonomii*, *Wyd. „Absolwent“*, Łódź, **1993**.

- [16] Cieślak, M. (ed.), Prognozowanie gospodarcze: metody i zastosowanie, Economic forecasts: methods and applications, *Wydawnictwo Naukowe PWN*, Warszawa, **2005**.
- [17] Czarny B., Bartkowiak R., Podstawy ekonomii, Warszawa, **2000**.
- [18] Dach Z., Podstawy mikroekonomii, *Wyd. Synaba*, Kraków, **2001**.
- [19] Davenport, M., The Fundamentals of Compressive Sensing, *SigView*, April 12, **2013**.
- [20] Deaton A.S., J. Mullbauer, Economics and Consumer Behaviour, *Cambridge University Press*, **1980**.
- [21] Domaradzki R., Zastosowanie sieci neuronowych do generowania strategii decyzyjnych (inwestycyjnych) na przykładzie Giełdy Papierów Wartościowych, *Rozprawa Doktorska*, Kraków, **2007**.
- [22] Duch, W., Korbicz, J., Rutkowski L., Tadeusiewicz R. Sieci Neuronowe, Tom 6 Biocybernetyka i Inżynieria Biomedyczna 2000, *Polskie Towarzystwo Sieci Neuronowych, Akademia Oficyna Wydawnicza EXIT*, PAN, Warszawa **2000**.
- [23] Encyklopedia agrobiznesu, *Fundacja Innowacja*, Warszawa, s. 240-241, **1998**.
- [24] Encyklopedia ekonomiczno-rolnicza, *PWRiL*, Warszawa, s. 323-324, **1984**.
- [25] Ferenc J.: Zarys ekonomiki i organizacji rolnictwa. Cz. 2., *WSRP*, Siedlce, **1997**.
- [26] Friedman M., Capitalism and Freedom, *University of Chicago Press*, Chicago, **1962**.
- [27] Friedman M., Theory of the Consumption Function. Princeton, Nj: Princeton University. Press for the NBER, **1957**.
- [28] Gajda, J. B., Prognozowanie i symulacja a decyzje gospodarcze, *Wydawnictwo C.H. Beck*, Warszawa, **2001**.
- [29] Gately E., Sieci neuronowe: prognozowanie finansowe i projektowanie systemów transakcyjnych, *WIG Press*, **1999**.
- [30] Goraj L, Mańko S.: Systemy monitorowania sytuacji ekonomicznej i produkcyjnej gospodarstw rolnych, w: *Rachunkowość rolnicza*, Difin, Warszawa, **2004**.
- [31] Gornowicz M., Polskie mleczarstwo w aspekcie konkurencyjności na jednolitym rynku UE, *Wydawnictwo Uniwersytetu Warmińsko-Mazurskiego*, **2003**.
- [32] Górską-Warsewicz H., Świątkowska M., Krajewski K., Marketing żywności, *Oficyna Wolters Kluwer Business*, Warszawa, **2013**.

- [33] Grabowski S., Specjalizacja i skala produkcji w rolnictwie, w: *Ekonomika rolnictwa - zarys teorii*, (red.) A. Woś i F. Tomczak, *PWRiL*, Warszawa, s. 187, **1997**.
- [34] Grabowski T., *Podstawy teorii ekonomii*, *Wyd.. Adam Marszałek*, Toruń, **2000**.
- [35] Green H.A.J., *Consumer Theory*. London: Penguin, **1971**.
- [36] Grochowski Z., Grabowski S., Koszty i opłacalność produkcji rolnej, [w:] *Ekonomika rolnictwa-zarys teorii*, (red.) A. Woś i F. Tomczak, *PWRiL*, Warszawa, s. 244-264, **1983**.
- [37] Haykin S., *Neural Network. A comprehensive foundation*, Pearson Education, **2005**.
- [38] Horodecka A., *Ewolucja celów polityki gospodarczej. Rola zmian otoczenia*, *PWN*, Warszawa, **2008**.
- [39] Jajuga, K., *Zarządzanie ryzykiem*, *Wydawnictwo Naukowe PWN*, Warszawa, **2007**.
- [40] Kaczmarek T. T., *Zarządzanie ryzykiem. Ujęcie interdyscyplinarne*, *DIFIN*, Warszawa, **2005**.
- [41] Kamerschen D.R., McKenzie R.B., Nardienelli C., *Ekonomia*, *Fundacja Gospodarcza NSZZ Solidarność*, Gdańsk, **1991**.
- [42] Karpuś P., Węclawski J. (red.), *Rynek finansowy. Szanse i Zagrożenia rozwoju. Instrumenty i strategie rynku finansowego*, t. 1, *Wyd. UMCS w Lublinie*, Lublin, **2005**.
- [43] Kaźmierczak M., Sznajder M., Sznajder E., *Polski sektor mleczarski 2017*, *Horyzont Mateusz Sznajder*, Przeźmierowo, **2017**.
- [44] Kiełtyka L., *Inteligentny System Prognozowania. Zasady funkcjonowania. Zastosowania*, *Wydawnictwo Politechniki Częstochowskiej*, Częstochowa, **2000**.
- [45] Klimczak B., *Mikroekonomia*, *Wyd.. AE we Wrocławiu*, Wrocław, **1998**.
- [46] Komański A., *Mikroekonomiczne aspekty roli współczesnego państwa*, *Ekonomia*, nr 5-6, **1992**.
- [47] Korbicz J. A., Obuchowicz D., *Sztuczne sieci neuronowe, podstawy i zastosowania*, *Akademicka Oficyna Wydawnicza*, Warszawa, **1994**.
- [48] Kosiński R. *Sztuczne sieci neuronowe, dynamika nieliniowa i chaos*. *Wydawnictwa Naukowo-Techniczne*, Warszawa, **2002**.

- [49] Kośmiecki E., Teoria konkurencji ekonomicznej. Próba oceny stanu badań i koncepcji, *RPE i S*, nr 1, **1988**.
- [50] Kraszewska M., Neuronowe modele decyzyjne w inwestowaniu w instrumenty pochodne na indeks PMP Giełdy Papierów Wartościowych, *Rozprawa Doktorska*, Kraków, **2012**.
- [51] Kucharski M., Rączkowski S., Wierzbicki J., Pieniądz i kredyt w kapitalizmie, *PWN*, Warszawa, **1973**.
- [52] Kudliński R., Strategia wielkich korporacji, *PWN*, Warszawa, **1972**.
- [53] Laidler D., Estrin S., Wstęp do mikroekonomii, *Gebethner i Ska*, Warszawa, **1991**.
- [54] Lambin J. J., Strategiczne zarządzanie marketingowe, *PWN*, Warszawa, **2001**.
- [55] Lange, K.: Optimization, Springer Texts in Statistics. *Springer*, New York, **2004**.
- [56] Levacic R., Economic Policy-Making, Sussex, **1987**.
- [57] Linder S.B., An Essay on Trade and Transformation, *Almqvist and Wiksells Boktryckeri AB*, Uppsala, **1961**.
- [58] Lipowski A., Mechanizm rynkowy w gospodarce rynkowej. Podstawy teoretyczne, perspektywy, dylematy, *PWN*, Warszawa, **1989**.
- [59] Luderer D., Nollau V., Vettters K., Mathematical Formulas for Economists, *Verlag F. Vahlen*, Munchen, **2010**.
- [60] Łuczka-Bakuła W. (red.), Mikroekonomia – wybrane zagadnienia, Wydawnictwo AR im. A. Cieszkowskiego, Poznań, s. 64-66, **2007**.
- [61] MacDonald J.M., Law J., Mosheim R., Consolidation in U.S. Dairy Farming United States Department of Agriculture (USDA), *Economic Research Report*, No. 274: 1–55, **2020**.
- [62] Malak-Rawlikowska A., Ekonomiczne i organizacyjne skutki wprowadzenia systemu regulacji produkcji mleka w wybranych krajach Unii Europejskiej i w Polsce, *Wydawnictwo SGGW*, Warszawa, **2005**.
- [63] Mankiw Gregory N., Taylor Mark P., Makroekonomia, *PWE*, Warszawa, **2009**.
- [64] Manteuffel R., Ekonomika i organizacja gospodarstwa rolniczego, *PWRiL*, Warszawa, **1984**.

- [65] Masters T., Sieci neuronowe w praktyce, Programowanie w języku C++, *Wydawnictwa Naukowo-Techniczne*, Warszawa, **1993**.
- [66] Mazurek-Łopacińska K., Zachowania nabywców i ich konsekwencje marketingowe, *PWE*, Warszawa, **2003**.
- [67] Mierzwa D., W poszukiwaniu nowego modelu spółdzielczości rolniczej, *Wydawnictwo Akademii Rolniczej*, Wrocław, **2005**.
- [68] Milewski R., Kwiatkowski E., Podstawy ekonomii, *PWN Warszawa*, **2003**.
- [69] Misala J. (red.), Makroekonomia gospodarki otwartej, *Politechnika Radomska*, Radom, **2006**.
- [70] Misala J., Pluciński E.M., Handel wewnątrzgałęziowy między Polską a UE, *SGH*, Warszawa, **2000**.
- [71] Mohinder S. Grewal, Angus P. Andrews, Kalman Filtering: Theory and Practice Using Matlab, *John Wiley and Sons*, **2001**.
- [72] Nahotko, S. Ryzyko ekonomiczne w działalności gospodarczej. *Oficyna Wydawnicza Ośrodka Postępu Organizacyjnego Sp. z o.o.*, Bydgoszcz, **2001**.
- [73] Nasiłowski M., System rynkowy. Podstawy mikro- i makroekonomii, *Wyd. Keytext*, Warszawa, **1998**.
- [74] Nowak E., Rachunek kosztów jako instrument zarządzania przedsiębiorstwem, w: Rachunek kosztów. Modele i zastosowania, E. Nowak, M. Wierzbiński, *PWE*, Warszawa, s. 13-18, **2010**.
- [75] Parzonko A., Bórawski P., Competitiveness of Polish dairy farms in the European Union, *Agricultural Economics*, 66: 168–174, **2020**.
- [76] Parzonko A., Globalne, regionalne i lokalne zróżnicowanie produkcji i przetwórstwa mleka, *Wydawnictwo SGGW*, **2013**.
- [77] Parzonko A., Inequalities in economic potential and the level of direct payments in dairy farms from selected EU countries, *Social Inequalities and Economic Growth*, 61: 43–54, **2020**.
- [78] Parzonko A., Inequalities in economic potential and the level of direct payments in dairy farms from selected EU countries, *Social Inequalities and Economic Growth*, 61: 43–54, **2020**.

- [79] Parzonko A., Bórawski P., Competitiveness of Polish dairy farms in the European Union, *Agricultural Economics-Zemledska Ekonomika*, 66(4), 168–174, **2020**.
- [80] Pesaran, B.; Pesaran, M.H. *Time Series Econometrics Using Microfit 5.0*; Oxford University Press: New York, NY, USA, 2009.
- [81] Piekara A., Samorządność, samorząd, rozwój, *Wydawnictwa Uniwersytetu Warszawskiego*, Warszawa, **2000**.
- [82] Popovics P.A., Analysis of economic issues relating to the dairy sector, with emphasis on price transmission, *Applied Studies in Agribusiness and Commerce (AP-STRACT)*, 2: 61–70, **2008**.
- [83] Porter M.E., Strategia konkurencji: Metody analizy sektorów i konkurentów, *PWE*, Warszawa, **1992**.
- [84] Ramirez C.A., Patel M., Blok K., From fluid milk to milk powder: Energy use and energy efficiency in the European dairy industry, *Energy*, 31: 1984–2004, **2006**.
- [85] Rembeza J., Seremak-Bulge J., Asymmetry in price transmission on milk market and its products, *Problems of Agricultural Economics*, 3: 110–123, **2006**.
- [86] Rezitis A.N., Stavropoulos K.S., Modeling beef supply response and price volatility under CAP reforms: The case of Greece, *Food Policy*, 35: 163–174, **2010**.
- [87] Rogall H., Ekonomia zrównoważonego rozwoju. Teoria i praktyka, *Wyd.. ZYSK I S-ka*, Poznań, **2010**.
- [88] Roman M., Milk market integration between Poland and the EU countries, **Agri-culture**, 10: 1–17, **2020**.
- [89] Roman M., Spatial integration of the milk market in Poland, *Sustainability*, 12: 1–15, **2020**.
- [90] Rojek R. Zastosowanie sztucznych sieci neuronowych i logiki rozmytej w automatyce, *Oficyna Wydawnicza*, Opole, **2000**.
- [91] Runowski H., Rachunek ekonomiczny w gospodarstwie rolnym, w: Poradnik dla rolnika w zakresie prowadzenia rachunkowości rolnej i zarządzania gospodarstwem rolnym, *SGGW*, Warszawa, **2003**.
- [92] Rymarczyk J., Handel zagraniczny organizacja i technika, *PWE*, Warszawa, **1996**.

- [93] Sarnecki J., Przetwórstwo i rynek mleka w Polsce w ujęciu przestrzennym, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej - PIB*, **2004**.
- [94] Schmitz M., Biofuel's impact on food prices should be put into perspective, *European Voice*, **2013**.
- [95] Setianti C., Ekowati T., Setiadi A., Market integration of fresh milk in dairy business area of Pamijahan District and Cisarua District, Bogor Regency – Indonesia, *Journal of the Indonesian Tropical Animal Agriculture*, 42: 42–47, **2017**.
- [96] Simo D., Muta L., Buleca J., Assessment of milk production competitiveness of the Slovak Republic within the EU-27 countries, *Agricultural Economics – Czech*, 62: 482–492.
- [97] Skarżyńska A., Abramczuk Ł., Analysis of the results of dairy farms of the largest milk producers in the European Union, *Agriculture and Food*, 5: 102–112, **2017**.
- [98] Skarżyńska A., Jabłoński K., Milk prices in the European Union by 2015 and projection of the profitability of milk production in Poland in 2020, *AGROFOR International Journal*, 1: 52–59, **2016**.
- [99] Sloman J., Mikroczyk I., Podstawy ekonomii, *PWE*, Warszawa, **2001**.
- [100] Stewart H., Blayney D.P., Retail dairy prices fluctuate with the farm value of milk. *Agricultural and Resource Economics Review*, 40: 201–217, **2011**.
- [101] Stock, J.H.; Watson, M.W. *Business Cycle Fluctuations in US Macroeconomic Time Series; Handbook of Macroeconomics*, NBER Working Paper Series, Elsevier, Amsterdam, Netherlands; No. 6528; 1998.
- [102] Stokes J.R., Tozer P.R., Hyde J., Identifying efficient dairy producers using data envelopment analysis, *Journal of Dairy Science*, 90: 2555–2562, **2011**.
- [103] Szajner P., Price transmission on milk market in Poland between 2004 and 2017, *Problems of Agricultural Economics*, 4: 3–23, **2017**.
- [104] Szajner, P. Wpływ likwidacji kwot mlecznych i zmian regulacji rynku mleka na perspektywy rozwoju polskiego mleczarstwa, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej*, Warszawa, **2014**.
- [105] Szczygieł A., Piekarska J., Popularne tablice wartości odżywczych żywności, *PZWL*, Warszawa, **1979**.
- [106] Sznajder M., Ekonomia mleczarstwa, *Wydawnictwo Akademii Rolniczej*, **1999**.

- [107] Szymański W., 2002, Globalizacja. Wyzwania i zagrożenia, *Difin*, Warszawa, **2002**.
- [108] Ślusarczyk B.L, Determinanty międzynarodowej pozycji konkurencyjnej gospodarki polskiej w okresie transformacji, *Wydawnictwo Uniwersytetu Rzeszowskiego*, Rzeszów, **2009**.
- [109] Špička J., The competitive environment in the dairy industry and its impact on the food industry, *Agris On-line Papers in Economics and Informatics*, 2: 89–102, **2013**.
- [110] Tadeusiewicz R., Lula P., Wprowadzenie do sieci neuronowych, *StatSoft*, Kraków, **2001**.
- [111] Tadeusiewicz, R. Sieci neuronowe. *Akademicka Oficyna Wydawnicza RM*, Seria: Problemy Współczesnej Nauki i Techniki. Informatyka, Warszawa **1993**.
- [112] Tauer L.W., Mishra A.K., Dairy farm cost efficiency, *Journal of Dairy Science*, 89: 4937–4943, **2006**.
- [113] Teresienė D., Dubauskas G., Modelling stock price of Lithuanian manufacture of milk and dairy products companies' volatility with GARCH models, *Management Theory and Studies for Rural Development*, 13: 154–161, **2008**.
- [114] Urban S. (red.), Agrobiznes i biobiznes. Teoria i praktyka, *Wyd. Uniwersytetu Ekonomicznego we Wrocławiu*, Wrocław, 133, **2011**.
- [115] Wang Q., Liu Ch., Zhao Y., Kitos A., Cannella M., Wand, S., Han L., Impacts of the COVID-19 pandemic on the dairy industry: Lessons from China and the United States and policy implications, *Journal of Integrative Agriculture*, 19: 2903–2915, **2020**.
- [116] Ward R.W., Asymetry in retail, wholesale, and shipping point pricing for fresh vegetables, *Marketing Science*, 23: 28–49, **1982**.
- [117] Wierzejski T., Lizińska W., Jakubowska D., Consumption and internationalization: determinants for the development of the dairy market in Poland, *European Research Studies Journal*, 23: 629–644, **2020**.
- [118] Wilczyński A., Kołoszyc E., Świtłyk M., Technical efficiency of dairy farms: An empirical study of producers in Poland, *European Research Studies Journal*, 23: 117–127, **2020**.
- [119] Wilson R., Can you see using convolutional neural networks?, **2015**.

- [120] Winiarski B. (red.), *Polityka gospodarcza*, PWN, Warszawa, **2002**.
- [121] Witkowska D., *Podstawy ekonometrii i teorii prognozowania*, Oficyna Ekonomiczna, Kraków, **2005**.
- [122] Witkowska, D., *Sztuczne sieci neuronowe i metody statystyczne*, Wybrane zagadnienia finansowe, *Wydawnictwo C.H. Beck*, Warszawa, **2002**.
- [123] Wojewodzic T., *Procesy dywestyacji i dezagrarnizacji w rolnictwie o rozdrobnionej strukturze agrarnej*, *Wydawnictwo Uniwersytetu Rolniczego*, **2017**.
- [124] Wrzosek W., *Funkcjonowanie rynku*, PWE, Warszawa, **1998**.
- [125] Wysokiński M., Baran J., Florkowski W.J., Concentration of milk production in Poland, *In: Proceedings of the 2015 International Conference Economic Science for Rural Development No. 37*, Jelgava, Apr 23–24, 2015: 93–104, **2015**.
- [126] Yao J, Tan CL., A case study on using neural networks to perform technical forecasting of forex, *Neurocomputing*. **2000**;34(1–4):79–98. doi: 10.1016/S0925-2312(00)00300-3.
- [127] Yilmaz Ö.T., A study of milk support policies in the European Union and in Turkey, *European Journal of Interdisciplinary Studies*, 9: 17–33, **2017**.
- [128] Zalewski A., *Gospodarka mleczarska a rynek*, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej - PIB*, **2000**.
- [129] Zieliński T.P., *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*, WKiŁ, Warszawa, **2002**.

Artykuły w czasopismach naukowych:

- [130] Adamowicz M., *Kwestia agrarna w Polsce i na świecie*, *Prace Naukowe*, t. 36, SGGW, Warszawa, **2005**.
- [131] Ahmed, N.; Natarajan, T.; Rao, K.R., Discrete cosine transform. *IEEE Trans. Comput.*, **1974**, 100, 90–93.
- [132] André, G.; Berentsen, P.B.M.; Engel, B.; De Koning, C.J.A.M.; Lansink, A.O., Increasing the revenues from automatic milking by using individual variation in milking characteristics. *J. Dairy Sci.*, 93, 942–953, **2010**.

- [133] Asche, F.; Bremnes, H.; Wessels, C., Product Aggregation, Market Integration, and Relationships between Prices. *Am. J. Agric. Econ.*, 81, 568–581, **1999**. doi: 10.2307/1244016.
- [134] Auffhammer M., Hsiang S.M., Schlenker W., Sobel A., Using weather data and climate model output in economic analyses of climate change, *Review of Environmental Economics and Policy*, 7(2), 181-198, **2013**.
- [135] Baer-Nawrocka A., Zmiany w spożyciu i stopniu samowystarczalności żywnościowej w Unii Europejskiej, *Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu*, 360, 19-27, **2014**.
- [136] Baer-Nawrocka, A., Grochowska, R., Kiryluk-Dryjska, E, Seremak-Bulge, J., Szajner, P., Światowy rynek mleka i jego wpływ na polskie mleczarstwo po zniesieniu kwot mlecznych, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej - Państwowy Instytut Badawczy*, Warszawa, **2012**.
- [137] Baffes J., Dennis A., Long-Term Drivers of Food Prices, *Policy Research Working Paper 6455*, Washington D.C, **2013**.
- [138] Bao Y., Xiong, T., Hu Z., Multi step ahead time series prediction using multiple-output support vector regression. *Neurocomputing*, 129, 482—493, **2014**.
- [139] Bao, Y.; Xiong, T.; Hu, Z. PSO-MISMO modeling strategy for multi step ahead time series prediction. *IEEE Trans. Cybern.*, 44, 655–668, **2013**.
- [140] Baxter, M., Real exchange rates and real interest differentials: Have we missed the business-cycle relationship, *J. Monet. Econ.*, 33, 5–37, **1994**.
- [141] Baxter, M.; King, R.G. *Measuring Business Cycles: Approximate Band-Pass Filters for Economic Time Series; Review of Economics and Statistics*, 1999, v81(4,Nov), 575-593, No. 5022; NBER Working Papers, MIT Press, Cambridge, MA, USA; **1995**.
- [142] Becker G.A., Theory of the Allocation of Time, *Economic Journal*, **1965**.
- [143] Beghin, J., Dairy Markets in Asia. An Overview of Recent Findings and Implications, Briefing Paper 05-BP-47, *Centre of Agricultural and Rural Development*, Iowa State: Ames, IA, USA, **2005**.
- [144] Bhosale, M.D.; Singh, T.P. Development of Lifetime Milk Yield Equation Using Artificial Neural Network in Holstein Friesian Cross Breddairy Cattle and Comparison with Multiple Linear Regression Model. *Curr. Sci.*, 113, 951–955, **2017**.

- [145] Bierlen R., Barry P.J., Dixon B.L., Ahrendsen B.L., Credit constraints, farm characteristics and the arm economy: Differential impacts on feeder cattle and beef cow inventories, *American Journal of Agricultural Economics*, nr 80, s. 708-709, **1998**.
- [146] Bowden G., Dandy G., Maier H. (2005): Forecasting Cyanobacteria (Blue-Green Algae) Using Artificial Neural Network. *Published by the American Society of Civil Engineers, EWRI, Virginia*, **2001**.
- [147] Box, G.E.P.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Holden Day Press: San Francisco, CA, USA, **1970**.
- [148] Brodziak A., Król J., Litwińczuk Z., Białka serwatkowe - właściwości funkcjonalne i zastosowanie, *Przemysł Spożywczy*, 3, 35-37, **2012**.
- [149] Candès Emmanuel J.; Romberg Justin K.; Tao, Terence, Stable signal recovery from incomplete and inaccurate measurements (PDF). *Communications on Pure and Applied Mathematics*. 59 (8): 1207-1223. arXiv:math/0503066, **2006**.
- [150] Candes, E. J.; Wakin, M. B.; Boyd, S. P., Enhancing sparsity by reweighted l1 minimization. *J. Fourier Anal. Applicat.* 14 (5-6): 877-905. arXiv:0711.1612. doi:10.1007/s00041-008-9045-x. S2CID 5879257. **2008**.
- [151] Candès, E.J., and Wakin, M.B., An Introduction To Compressive Sampling, *IEEE Signal Processing Magazine*, V.21, **2008**.
- [152] Candès, Emmanuel J.; Romberg, Justin K.; Tao, Terence, Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Fourier Information (PDF). *IEEE Trans. Inf. Theory*. 52 (8): 489-509. arXiv:math/0409186. citeSeerX 10.1.1.122.4429. doi:10.1109/tit.2005.862083. S2CID 7033413. **2006**.
- [153] Carreira-Perpinan M.A., Hinton G., On contrastive divergence learning., *AISTATS*, tom 10, 33-40, citeseer, **2005**.
- [154] Cole, J.B.; Null, D.J.; VanRaden, P.M. Best prediction of yields for long lactations. *J. Dairy Sci.*, 92, 1796–1810, **2009**.
- [155] Combettes, P.; Wajs, V., Signal recovery by proximal forward-backward splitting, *Multiscale Model Simul.* 4 (4): 1168-200. doi:10.1137/050626090. S2CID 15064954, (2005).
- [156] Czajkowski D., Rynek mleka przed i po akcesji Polski do Unii Europejskiej : kierunki zmian, *Roczniki Naukowe Stowarzyszenia Ekonomistów Rolnictwa i Agrobiznesu*, nr 15, z 40, **2013**.

- [157] Czarny E., Teoria i praktyka handlu wewnątrzgałęziowego, *Monografie i Opracowania SGH*, nr 496, SGH, Warszawa, **2002**.
- [158] Czubak W., Pawłowski K., Wiza P., Arion F., Polish and Romanian dairy farms using EU investment support: a comparative study, *Annals of the Polish Association of Agricultural and Agribusiness Economists*; Vol 21, no 4, pp. 64-75, e-ISSN 2450-7296 (new 2657-7828), **2019**.
- [159] Czubak W., Rozwój rolnictwa w Polsce z wykorzystaniem wybranych mechanizmów wspólnej polityki rolnej Unii Europejskiej, *UNIWERSYTET PRZYRODNICZY W POZNANIU*, **2013**.
- [160] Czyżewski A., Kata R., Matuszczak A., Wpływ krajowych i unijnych wydatków budżetowych na alokację czynników produkcji w polskim rolnictwie, *Ekonomista*, nr 1, 45-72, **2019**.
- [161] Dietl J., Elementy rynku produktów rolnych, *Ruch Prawniczy i Ekonomiczny* 20, z .3, s. 199-224, **1958**.
- [162] Dixon, M.F.; Klabjan, D.; Bang, J.H., Classification-based financial markets prediction using deep neural networks. *Algorithmic Financ.*, 6, 67–77, **2016**.
- [163] Dong, F., The Outlook for Asian Dairy Markets: The Role of Demographics, Income, and Prices. *Food Policy*, .31, 260–271, **2006**. doi:10.1016/j.foodpol.2006.02.007.
- [164] Dongre V.B., Hi G., Singh R.S., Ruhil A.P., Comparative efficiency of artificial neural networks and multiple linear regression analysis for prediction of first lactation 305-day milk yield in Sahiwal cattle. *Livest. Sci.*, 147, 192–197, **2012**.
- [165] Donoho D. L., For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*. 59 (6): 797-829. doi:10.1002/cpa.20132. S2CID 8510060, **2006**.
- [166] Donoho, D.L., Compressed sensing, *IEEE Transactions on Information Theory*. 52 (4): 1289-1306. doi:10.1109/TIT.2006.871582. S2CID 206737254, **2006**.
- [167] Editha, R.P, Motivational Factors on the Adoption of Natural Farming Technology, *Research Journal of Agriculture and Forestry Sciences*, Vol. 4(1), 14-19, **2016**.
- [168] Eldar, Y.C.; Kutyniok, G. *Compressed Sensing: Theory and Applications*; Cambridge University Press: Cambridge, UK, **2012**.

- [169] Faria, E.L.; Albuquerque, M.P.; Gonzalez, J.L.; Cavalcante, J.T.P.; Albuquerque, M.P., Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods, *Expert Syst. Appl.*, *36*, 12506–12509, **2009**.
- [170] Felczak T., Kosztocłonność i rentowność gospodarstw indywidualnych w zależności od typu rolniczego, *Zeszyty Naukowe SGGW*, nr 89, s. 101, **2011**.
- [171] Fontagn L., Freudenberg M., Long – Term Trends in Intra – Industry Trade, *Frontiers of Research in Intra – Industry Trade*, Palgrave Macmillan, New York, **2002**.
- [172] Ghorbanzadeh O., Blaschke, T., Gholamnia, K., Meena, S. R., Tiede, D., and Aryal, J. Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection, *Remote sensing*, *11*(2), 1-21. **2019**.
- [173] Ghorbanzadeh, O.; Tiede, D.; Dabiri, Z.; Sudmanns, M.; Lang, S. Dwelling extraction in refugee camps using CNN—First experiences and lessons learnt. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 11161-166, **2018**.
- [174] Gomez, V. The Use of Butterworth Filters for Trend and Cycle Estimation in Economic Time Series, *J. Bus. Econ. Stat.*, *19*, 365–373, **2001**.
- [175] Guirado, E.; Tabik, S.; Alcaraz-Segura, D.; Cabello, J.; Herrera, F. Deep-learning convolutional neural networks for scattered shrub detection with google earth imagery. arXiv 2017, arXiv:1706.00917, **2017**.
- [176] Hajdukiewicz, A., Przyczyny wzrostu światowych cen produktów rolno-spożywczych w pierwszej dekadzie XXI wieku, *Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu*, *369*, 239-250, **2014**.
- [177] Hamulczuk, M., Prognozowanie cen surowców rolnych z wykorzystaniem modeli szeregów czasowych, Hayes, B., The Best Bits, *American Scientist*. *97* (4): 276. doi:10.1511/2009.79.276. S2CID 349102. **2009**.
- [178] He K., Zhang X., Ren S., Sun J., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *CoRR*, abs/1502.01852, **2015**.
- [179] He, K.; Lai, K.K.; Yen, J., A hybrid slantlet denoising least squares support vector regression model for exchange rate prediction, *Procedia Comput. Sci.*, *1*, 2397–2405, **2010**.
- [180] Hinton G., A fast learning algorithm for deep belief nets, *Neural Computation*. Vol. 18. - P. 1527-1554. **2006**.

- [181] Hinton, G. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786). - 2006. - P. 504-507. **2006**.
- [182] Hinton, G. Training products of experts by minimizing contrastive divergence *Neural Computation*. Vol. 14. - P. 1771-1800. **2002**.
- [183] Hochreiter S., Schmidhuber J., Long short-term memory. *Neural Comput.*,9, 1735–1780, **1997**.
- [184] Hodrick R.J., Prescott, E.C. Postwar U.S., Business Cycles: An Empirical Investigation, *Journal of Money, Credit and Banking* 1997, Vol. 29, No, 1 (Feb. 1997, Published by: Ohio State University Press), s. 1-6. **1997**.
- [185] Hong, Y., Liu, Y., Wang, S., Granger Causality in Risk and Detection of Extreme Risk Spillover between Financial Markets, *Journal of Econometrics*, 150 (2), pp. 271-287, **2009**.
- [186] Hornik K., Stinchcombe M., White H., Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, 2, 359–366.
- [187] Hush D., Progress in supervised neural networks, *IEE Signal Processing Magazine*, **1993**.
- [188] Jain A., Ashish K. V., Umesh C. J. Short-Term Water Demand Forecast Modeling at IIT Kanpur Using Artificial Neural Networks, *Water Resources Management* 15: 299-321, 20002 Kluwer Academic Publishers, the Netherlands, **2001**.
- [189] Jankowski K., Sosnowski J., Wpływ intensywności gospodarowania na efekty produkcyjne gospodarstw mlecznych, *J. Res. Appl. Agric. Eng.*, 56(1): 55-58, **2011**.
- [190] Jaswal, D.; Soman, K.P., Image classification using convolutional neural networks, *Int. J. Adv. Res. Technol.*, 3, doi:10.1007/978-3-319-16841-8-52, **2014**.
- [191] Jin, J.; Yang, B.; Liang, K.; Wang, X. General image denoising framework based on compressive sensing theory. *Comput. Graph.* **2014**, 38, 382—391.
- [192] Joo C. N., Koo J. Y., Yu M. J., Application of short-term demand prediction model to Seoul, *Water Science and Technology*, Vol 46 No 6-7 pp 255-261, IWA, **2002**.
- [193] Kasztelan P., System regulacji produkcji mleka w Polsce, *Roczniki Nauk Rolnych*, seria G, t. 96, z. 1, **2009**.
- [194] Klamra, M. Podstawowe idee próbkowania oszczędnego. *Technical Transactions* **2012**, 18/109.

- [195] Kominakis, A.P.P.; Abas, Z.; Maltaris, I.; Rogdakis, E. A preliminary study of the application of artificial neural networks to prediction of milk yield in dairy sheep. *Comput. Electron. Agric.* **2002**, *35*, 35–48.
- [196] Krizhevsky A., Sutskever I., Hinton G.E., Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 1097-1105, **2012**.
- [197] Krugman P.R., Intraindustry Specialization and the Gains form Trade, *The Journal of Political Economy*, vol. 89, no. 5, s. 959-973, **1981**.
- [198] Långkvist, M.; Kiselev, A.; Alirezaie, M.; Loutfi, A. Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sens.*, 8, 329, **2016**.
- [199] LeCun Y., Bottou L., Bengio Y. and Haffner P., Gradient-based learning applied to document recognition, *Proc. IEEE* *86*, 2278-2324, **1998**.
- [200] LeCun, Y. et al., Handwritten digit recognition with a back-propagation network, *Proc. Advances in Neural Information Processing Systems* 396-404, **1990**.
- [201] LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; Volume 3361; **1995**, MIT Press, Cambridge, MA, USA.
- [202] Licznar, P., Łomotowski, J., (2006): Zastosowanie sztucznych sieci neuronowych Kohonena do prognozowania dobowego poboru wody. *Ochrona środowiska, Rok 28, Nr 1*, **2001**.
- [203] Litwińczuk, Z., Grodzik, H., Stan hodowli i chowu bydła w Polsce oraz czynniki warunkujące rozwój tego sektora, *Przegląd Hodowlany*, 6, 1-5, **2014**.
- [204] Luciani S.F., A Note to the Concept of Two Way Trade, *Review of World Economics*, vol. 117, no. 1, **1981**.
- [205] Lyons W.B., Fitzpatrick C., Flanagan C., Lewis E., A novel multipoint luminescent coated ultra violet fibre sensor utilising artificial neural network pattern recognition techniques. *Sens. Actuators A Phys.* **2004**, *115*, 267–272.
- [206] Ma J., Tian G., Risks, financing constraints, and high savings ratio in the rural economy of China: A model incorporating precautionary savings and liquidity constraints, *Frontiers of Economics in China*, vol. 1, No. 1, s. 114, **2006**.

- [207] Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.*, 55, 645-657, textbf2017.
- [208] Malczewski, J., Czubak, W. Hybrid Convolutional Neural Networks Based Framework for Skimmed Milk Powder Price Forecasting. *Sustainability*, 13(7), <https://doi.org/10.3390/su13073699>, **2021**.
- [209] Mandecki S., Zagadnienie wielkości gospodarstw rolnych w świetle zmian w strukturze agrarnej niektórych krajów zachodnich, *Zagadnienia Ekonomiki Rolnej*, nr 4, s. 3—31, **1957**.
- [210] Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Huang, Z.; Yuille, A., Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv* **2014**, arXiv:1412.6632.
- [211] Medsker L.R, L.R.; Jain L.C., *Recurrent Neural Networks: Design and Applications*; CRC Press, Boca Raton, Florida, USA, 2001.
- [212] Mirzaee, H., Long-term prediction of chaotic time series with multi-step prediction horizons by a neural network with Levenberg-Marquardt learning algorithm, *Chaos Solitons Fractals* **2009**, 41, 1975–1979.
- [213] Murphy M.D., O'Mahony M.J., Shalloo L., French P., Upton J., Comparison of modelling techniques for milk-production forecasting, *J. Dairy Sci.* **2014**, 97, 3352–3363.
- [214] Nasser, M.; Moeini, A.; Tabesh, M., Forecasting monthly urban water demand using Extended Kalman Filter and Genetic Programming, *Expert Syst. Appl.*, **2011**, 38, 7387–7395.
- [215] Nassif, A., Feijo, C., Araujo, E., The BRICS's Economic Growth Performance before and after the International Financial Crisis, *International Journal of Political Economy*, 45 (4), pp. 294-314, **2016**.
- [216] Neil A.J., Frey R., Estimation of Tail-Related Risk for Heteroscedastic Financial Time Series: an Extreme Value Approach, *Journal of Empirical Finance*, 7 (3), pp. 271-300, **2000**.
- [217] Neill, J., Building Better Global Economic BRICs, *Global Economics Paper*, 66, **2001**.
- [218] Okularczyk S., Dylematy ekologicznej produkcji zwierzęcej w polskich uwarunkowaniach ekonomicznych i rynkowych, *Przegląd Hodowlany*, vol. 72 (03), **2004**.

- [219] Olszewska, M., Produkcja mleka w Polsce na tle świata i krajów Unii Europejskiej, *Wiadomości Zootechniczne*, 53(3), 150-157, **2015**.
- [220] Pailwar V.K., Kaur J., Saxena K., Nijhara M., Impact of membership of financial institutions on rural saving: a micro-level study, *International Business and Economics Research Journal*, nr 9(10), s. 612, **2010**.
- [221] Parzonko A., Bórawski P., Competitiveness of Polish dairy farms in the European Union, *Agricultural Economics-Zemledska Ekonomika*, 66(4), 168–174, **2020**.
- [222] Parzonko, A., Globalne i lokalne uwarunkowania rozwoju produkcji mleka, *SGGW*, Warszawa, **2013**.
- [223] Parzonko, A., Regionalne zmiany produkcji mleka w Polsce - stan i przyczyny, *Roczniki Ekonomiczne Kujawsko-Pomorskiej Szkoły Wyższej w Bydgoszczy*, 7, 218-233, **2014**.
- [224] Pawlewicz, A., Szamrowski, P., Funkcjonowanie i rozwój rynku ekologicznych surowców żywnościowych w nowej perspektywie finansowej w latach 2014-2020. *Więś i Rolnictwo*, 3(164), 175-188. PSR - Powszechny Spis Rolny 2010 GUS, **2014**.
- [225] Pedregosa F., Varoquaux G., Gramfort A., Michel A, et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **2011**, 1110, 2825–2830.
- [226] Penson J.B., Gardner B.L., Implications of the Macroeconomic Outlook for Agriculture, *American Journal of Agricultural Economics*, vol. 70(5), s. 1013-1022, **1998**.
- [227] Pietrzak M., Problem geograficznego zakresu rynków/sektorów w dobie globalizacji, *Zagadnienia Ekonomiki Rolnej*, nr 1, s. 3-21, IERiGŻ-PIB, Warszawa, **2014**.
- [228] Pietrzak, M. Wpływ likwidacji kwot mlecznych i zmian regulacji rynku mleka na perspektywy rozwoju polskiego mleczarstwa, *Problem geograficznego zakresu rynków/sektorów w dobie globalizacji, Zagadnienia Ekonomiki Rolnej, nr 1, s. 3-21*, Warszawa, **2014**.
- [229] Poczta W., Czubak W., Kiryluk-Dryjska E., Sadowski A., Siemiński P., Koncepcja ukierunkowania wsparcia gospodarstw rolnych w perspektywie 2014-2020, *Ekspertyza wykonana dla MRiRW w Warszawie*, **2012**.
- [230] Qayyum, A.; Malik, A.S.; Saad, N.M.; Iqbal, M.; Faris Abdullah, M.; Rasheed, W.; Rashid Abdullah, T.A.; Bin Jafaar, M.Y. Scene classification for aerial images based on CNN using sparse coding technique. *Int. J. Remote Sens.*, **2017**, 38, 2662-2685.

- [231] Radovic, M.; Adarkwa, O.; Wang, Q. Object recognition in aerial images using convolutional neural networks. *J. Imaging*, 3, 21 , **2017**.
- [232] Rasekhschaffe, K.; Jones, R. Machine learning for stock selection. *Financ. Anal. J. Forthcom*, **2019**, 6, 67–77.
- [233] Runowski H., Teoretyczne i praktyczne aspekty obrotu kwotami mlecznymi w gospodarstwach rolnych, *Roczniki Nauk Rolniczych*, Seria G: Ekonomika Rolnictwa, tom 93, nr 1, **2006**.
- [234] Rusielik R., Świtłyk M., Efektywność techniczna produkcji mleka w wybranych europejskich gospodarstwach w latach 2008-2010. *Roczniki Nauk Rolniczych*, Seria G, 99(1), 88-99, **2011**.
- [235] Sak H., Senior A.W., Rao K., Beaufays F., Fast and accurate recurrent neural network acoustic models for speech recognition, CoRR, abs/1507.06947, **2015**.
- [236] Sang, Y.F., Improved wavelet modeling framework for hydrologic time series forecasting, *Water Resour. Manag.* **2011**, 27, 2807–2821.
- [237] Sang, Y.F.; Wang, D.; Wu, J.C.; Zhu, Q.P.; Wang, L. Entropy-based wavelet denoising method for time series analysis, *Entropy* **2009**, 11, 1123–1147.
- [238] Sarnecki J., Przetwórstwo i rynek mleka w Polsce w ujęciu przestrzennym, *Studia i Monografie*, nr 119, IERiGŻ, Warszawa, **2004**.
- [239] Schenk-Hoppe, K.R., Economic Growth and Business Cycles: A Critical Comment on Detrending Time Series, *Stud. Nonlinear Dyn. Econom.* **2001**, 5, 75–86.
- [240] Seremak-Bulge J. (red.), Handel zagraniczny produktami rolno-spożywczymi w latach 1995-2009, praca zbiorowa, *Studia i Monografie*, nr 152, IERiGŻ-PIB, Warszawa, **2011**.
- [241] Shambora, W.E.; Rossiter, R., Are there exploitable inefficiencies in the futures market for oil? *Energy Econ.* **2007**, 29, 18–27.
- [242] Siddiqui, K., Will the Growth of the BRICs Cause a Shift in the Global Balance of Economic Power in the 21st Century?, *International Journal of Political Economy*, No, 45(4), pp. 315-338, **2016**.
- [243] Siwoń Z., Stanisławski J. Stochastyczne prognozowanie chwilowych poborów wody, *Ochrona środowiska* Nr 2(35), **1998**.

- [244] Siwoń, Z., Cieżak, W., Cieżak, J., Bieżące prognozowanie godzinowego poboru wody z miejskich sieci wodociągowych, *Monografie Komitetu Inżynierii środowiska PAN*, vol. 30, 15-33, **2005**.
- [245] Smith, L.P., Forecasting annual milk yields. *Agric. Meteorol.* **1968**, 5, 209–214.
- [246] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, 15, 1929-1958, **2014**.
- [247] Stańko S., Mikuła A., Tendencje na rynku mleka na świecie i w Polsce w latach 2000-2016, *Zeszyty Naukowe SGGW w Warszawie. Problemy Rolnictwa światowego*, tom 18(33), nr z. 1, 235-247, **2018**.
- [248] Stańko S., Tendencje w produkcji, konsumpcji i handlu zagranicznym na rynku mleka i jego artykułów w latach 1990-2005, *Problemy Rolnictwa Światowego*, t. 15, 359, **2006**.
- [249] Stańko S., Zewnętrzne uwarunkowania rozwoju rolnictwa, *Roczniki Nauk Rolniczych*, Seria G, 94(2), 65-79, **2008**.
- [250] Stępień S., Związki wahań cyklicznych w rolnictwie z koniunkturą gospodarczą, *Roczniki Nauk Rolniczych*, Seria G, 98(3), 32-41, **2011**.
- [251] Strzelecka A., Struktura finansowania majątku mikroprzedsiębiorstw rolniczych w Polsce, W:]J. Buko (red.), Uwarunkowania rynkowe rozwoju mikro i małych przedsiębiorstw, *Zeszyty Naukowe Uniwersytetu Szczecińskiego nr 695*, Ekonomiczne Problemy Usług, nr 80, s. 234-245, **2012**.
- [252] Sulmicki, J., Chiny umacniają dominującą pozycję w gospodarce światowej, Akademia Finansów i Biznesu Vistula, *Zeszyty Naukowe Uczelni Vistula. Stosunki Międzynarodowe*, 41 (3), pp. 33-44, **2015**.
- [253] Sych-Winiarek, J. Rynek mleka w 2008 roku, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej*, Warszawa, **2010**.
- [254] Sych-Winiarek, J. Rynek mleka w latach 2009-2010, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej*, Warszawa, **2011**.
- [255] Szczepaniak I. (red.), Konkurencyjność polskich producentów żywności i jej determinanty (1), PW 2015-2019, nr 11, *IERiGŻ-PIB*, Warszawa, **2015**.

- [256] Szajner P., Ocena wpływu likwidacji kwot mlecznych na konkurencyjność polskiego mleczarstwa w kontekście teorii ekonomii, *Problemy Rolnictwa Światowego*, t. 12(27), z. 2, 112, **2012**.
- [257] Świetlik, K., Rynek mleka stan i perspektywy, *Zakład Badań Rynkowych IERiGŻ-PIB*, Warszawa, **2020**.
- [258] Takeuchi L., Lee Y., Applying deep learning to enhance momentum trading strategies in stocks. In *Technical Report*; Stanford University, Stanford, CA, USA 2013.
- [259] Tarnowska A., Miejsce Polski na światowym rynku mleka i jego przetworów, *Roczniki Naukowe Stowarzyszenia Ekonomistów Rolnictwa i Agrobiznesu*, tom 17, nr z. 3, 392-397, **2015**.
- [260] Tkachenko R., Izonin I., Kryvinska N., Tkachenko, P. Multiple Linear Regression Based on Coefficients Identification Using Non-iterative SGTM Neural-like Structure. *Adv. Comput. Intell.*, doi:10.1007/978-3-030-20521-8-39, **2019**.
- [261] Tkachenko R., Izonin I., Model and Principles for the Implementation of Neural-Like Structures Based on Geometric Data Transformations. *Adv. Intell. Syst. Comput.*, doi:10.1007/978-3-319-91008-6-58, 2019.
- [262] Tkachenko R., Izonin I., Vitynskyi P., Lotoshynska N., Pavlyuk O., Development of the Non-Iterative Supervised Learning Predictor Based on the Ito Decomposition and SGTM Neural-Like Structure for Managing Medical Insurance Costs. *Data J.* **2018**, 3, 46.
- [263] Urban R. (red.), 2008, Wpływ integracji z UE na polską gospodarkę żywnościową, PW, nr 90, *IERiGŻ-PIB*, Warszawa, **2008**.
- [264] Urban S., Zmiany w produkcji mleczarskiej w Polsce, *Rocz. Nauk. SERiA*, t. XIII, z. 1, 412-416, **2011**.
- [265] Van Bebber, J.; Reinsch, N.; Junge, W.; Kalm, E., Monitoring daily milk yields with a recursive test day repeatability model (Kalman filter). *J. Dairy Sci.* **1999**, 82, 2421–2429.
- [266] Von Braun J., The World Food Situation. New Driving Forces and Required Actions, *International Food Policy Research Institute*, Washington, **2007**.
- [267] Weldesenbet, T., Asymmetric price transmission in the Slovak liquid milk market. *Agric. ECON-Czech*, 59(11), 512-524, **2013**.

- [268] Willer H., Lernoud J., The world of organic agriculture, *Statistics and emerging trends 2016*, (pp. 1-336). Research Institute of Organic Agriculture FiBL and IFO-AM Organics International, **2016**.
- [269] Wojtyna A., Rola państwa we współczesnej ekonomii, *Ekonomista*, nr 3, 90, **1992**.
- [270] Wziątek-Kubiak A., Orientacja proeksportowa a selektywna polityka przemysłowa, *Studia Ekonomiczne*, nr 33, INE, PAN, Warszawa, **1994**.
- [271] Xie, W.; Yu, L.; Xu, S.; Wang, S., A new method for crude oil price forecasting based on support vector machines, *Computational Science—ICCS*; Springer: Berlin/Heidelberg, Germany, **2006**; pp. 444—451.
- [272] Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical evaluation of rectified activations in convolutional network, *arXiv*, **2015**, arXiv:1505.00853.
- [273] Xuan F.; Zhihui W.; Liang X., Iterative Directional Total Variation Refinement for Compressive Sensing Image Reconstruction. *IEEE Signal Processing Letters*. 20 (11): 1070-1073. Bibcode:2013ISPL...20.1070F. doi:10.1109/LSP.2013.2280571. **2013**.
- [274] Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K., Convolutional neural networks: an overview and application in radiology, *Insights Imaging*, **2018**, 9, 611–629, doi:10.1007/s13244-018-0639-9.
- [275] Yu, L.; Lai, K.K.; Wang, S.; He, K., Oil price forecasting with an EMD-based multiscale neural network learning paradigm. *Computational Science—ICCS*, Springer: Berlin/Heidelberg, Germany, **2007**; pp. 925–932.
- [276] Yu, Y.; Zhao, L. Tang, A compressed sensing based AI learning paradigm for crude oil price forecasting. *Energy Econ.* **2014**, 46, 236–245.
- [277] Yuan, C. Forecasting exchange rates: The multi-state Markov-switching model with smoothing. *Int. Rev. Econ. Financ.* **2011**, 20, 342–362.
- [278] Zakrzewski P., O zmianę dotychczasowego systemu obowiązkowych dostaw zbóż, *Myśl Gospodarcza*, nr 6, s. 67—79, **1957**.
- [279] Zawadzka D., Ocena wykorzystania kredytu handlowego przez przedsiębiorstwa rolnicze w Polsce, W:]H. Zadora, G. Łukasik (red.), *Finanse w niestabilnym otoczeniu – dylematy i wyzwania. Finanse przedsiębiorstw*, *Studia Ekonomiczne, Zeszyty Naukowe Wydziałowe Uniwersytetu Ekonomicznego w Katowicach*, nr 107, Katowice, s. 331-339, **2012**.

- [280] Zawojcka A., Czy spekulacje finansowe wpływają na międzynarodowe ceny towarów rolno-żywnościowych?, *Zeszyty Naukowe SGGW w Warszawie, Problemy Rolnictwa światowego*, 11(1), 177-191, **2011**.
- [281] Zhang W., Yu E.A, Rozelle S., Yang J., Msangi S., The impact of biofuel growth on agriculture: Why is the range of estimates so wide?, *Food Policy*, 38, 227-239, **2013**.
- [282] Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.-S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.*, 5, 8-36, **2017**.
- [283] Ziętara W., Stan i kierunki zmian w produkcji mleka w Polsce, *Rocz. Nauk Rol.*, seria G, t. 93, z. 1, 7, **2006**.
- [284] Zuba-Ciszewska M., Structural changes in the milk production sector and food security - the case of Poland, *Annals of the Polish Association of Agricultural and Agribusiness Economists*, 21 (2), 318-327, **2019**.
- [285] Zychowicz A., Wyzwanie naszych czasów, *Życie gospodarcze*, nr 41, **1994**.

Artykuły w materiałach konferencji naukowych:

- [286] Bouamra-Mechemache, Z., Jongeneel, R., Requillart, V. The dairy industry in an expending EU policies and strategies, *EDIM/KOM: Meeting at DG Agri, April 24*, Brussels, **2007**.
- [287] Chen, B.T.; Chen, M.Y.; Fan, M.H.; Chen, C.C. Forecasting stock price based on fuzzy time-series with equal-frequency partitioning and fast Fourier transform algorithm. In Proceedings of the Computing, Communications and Applications Conference (ComComAp), Hong Kong, China, 11–13 January, **2012**; pp. 238–243.
- [288] Dahmane, M.; Foucher, S.; Beaulieu, M.; Riendeau, F.; Bouroubi, Y.; Benoit, M., Object detection in pleiades images using deep features, *Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Beijing, China; pp. 1552-1555, **2016**.
- [289] Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Fast multiclass object detection in optical remote sensing images using region based convolutional neural networks. *Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Fort Worth, TX, USA, pp. 858-861, **2017**.

- [290] Glorot X., Bengio Y., Understanding the difficulty of training deep feedforward neural networks, *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*. Society for Artificial Intelligence and Statistics, **2010**.
- [291] Graves, A.; Mohamed, A.R.; Hinton, G., Speech recognition with deep recurrent neural networks, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, 26–31 May, **2013**; pp. 6645—6649.
- [292] Han, B.; Xiong, J.; Li, L.; Yang, J.; Wang, Z. Research on millimeter-wave image denoising method based on contourlet and compressed sensing. In Proceedings of the 2010 2nd International Conference on Signal Processing Systems (ICSPS), Dalian, China, 5–7 July, **2010**; Volume 2, pp. V2-471–V2-475.
- [293] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June, **2016**; pp. 770–778.
- [294] He, Z.; Zhou, J.; Dai, H.-N.; Wang, H. Gold Price Forecast Based on LSTM-CNN Model. In Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberScitech), Fukuoka, Japan, 5–8 August, **2019**; pp. 1046–105.
- [295] Henry de Frahan B.H., Tharakan J., Horizontal and Vertical Intra-Industry Trade in the Processed Food Sector, Catholic University of Louvain, *American Agricultural Economics Association Annual Meeting*, Salt Lake City, **1998**.
- [296] Korzeniowski L., Firma w warunkach ryzyka gospodarczego, *Kraków: European Association for Security*, **2002**.
- [297] Kowalski A., Wigier M., Wieliczko B., WPR a konkurencyjność polskiego i europejskiego sektora żywnościowego, *Materiały z konf. 26-28 listopada 2014 -Józefów k/Warszawy*, IERiGŻ-PIB, **2014**.
- [298] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 3431-3440, **2015**.

- [299] Longguang Y., Qing W., Predicting the stock price based on BP neural network and big transaction, *International Conference on Fuzzy System and Knowledge Discovery*, **2012**.
- [300] Malak-Rawlikowska A., Ekonomiczne i organizacyjne skutki wprowadzenia systemu regulacji produkcji mleka w wybranych krajach Unii Europejskiej i w Polsce, *Wydawnictwo SGGW*, Warszawa, **2005**.
- [301] Mech M., Polski sektor mleczarski 2013, *Horyzont Mateusz Sznajder*, Przeźmierowo, **2013**.
- [302] Miegel M., Granice wzrostu, *Forum 9*, 21-23, **2008**.
- [303] Muszyński K., Knapik K., Metoda SSN w bieżącym prognozowaniu zapotrzebowania na wodę, *IV Ogólnopolska Konferencja Naukowo-Techniczna „Aktualne zagadnienia w uzdatnianiu i dystrybucji wody”*, Szczyrk 10-12, **2007**.
- [304] Paoli, C.; Voyant, C.; Muselli, M.; Nivet, M.L. Use of exogenous data to improve an Artificial Neural Networks dedicated to daily global radiation forecasting. In *Proceedings of the 2010 9th International Conference on Environment and Electrical Engineering*, Prague, Czech Republic, 16–19 May 2010; pp. 49–52.
- [305] Rockefeller, B. *Technical Analysis for Dummies*, 3rd ed.; Wiley Publishing, Inc., Hoboken, New Jersey, USA, **2014**.
- [306] Siwoń Z. Problemy krótkoterminowego prognozowania poboru wody w miejskich systemach wodociągowych, *II Ogólnopolska Konferencja Naukowo - Techniczna „Aktualne zagadnienia w uzdatnianiu i dystrybucji wody*, Szczyrk 5-6 czerwiec, **2003**.
- [307] Tieleman T., Training restricted boltzmann machines using approximations to the likelihood gradient, *Proceedings of the 25th international conference on Machine learning*, 1064-1071, ACM, **2008**.
- [308] Tsantekidis, A.; Passalis, N.; Tefas, A.; Kannianen, J.; Gabbouj, M.; Iosifidis, A. Forecasting stock prices from the limit order book using convolutional neural networks. In *Proceedings of the 2017 IEEE 19th Conference on Business Informatics (CBI)*, Thessaloniki, Greece, 24–27 July, **2017**; Volume 1, pp. 7–12.
- [309] Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y., Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, Lille, France, 7–9 July, **2015**, Volume 14, pp. 77–81.

- [310] Yang, H.L.; Lunga, D.; Yuan, J., Toward country scale building detection with convolutional neural network using aerial images. *Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Fort Worth, TX, USA, pp. 870-873, **2017**.
- [311] Zhu, L.; Zhu, Y.; Mao, H.; Gu, M. A new method for sparse signal denoising based on compressed sensing. In *Proceedings of the Second International Symposium on Knowledge Acquisition and Modeling*, Wuhan, China, 30 November-1 December, **2009**; pp. 35–38.
- [312] Zhu, Y.; Groth, O.; Bernstein, M.; Fei-Fei, L. Visual7w: Grounded question answering in images, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June, **2016**; pp. 4995–5004.

Raporty rynkowe:

- [313] ARR, Handel zagraniczny towarami rolno-spożywczymi w 2013 r., *Biuro Analiz i Programowania ARR*, Warszawa, **2014**.
- [314] European Commission 2014a, Milk Market Observatory. *Milk Market Observatory*, Brussels, **2014**.
- [315] Handel zagraniczny na rynku mleka, *Agencja Rynku Rolnego*, s. 1-15, Warszawa, **2005**.
- [316] International Dairy Federation. The World Dairy Situation 2014, *Bulletin of the International Dairy Federation*, 476/2014, London, **2014**.
- [317] International Dairy Federation. The World Dairy Situation 2014, *Bulletin of the International Dairy Federation*, 476/2014, London, **2014** .
- [318] Polski handel zagraniczny podstawowymi produktami rolno-spożywczymi w 2019 r., *Krajowy Ośrodek Wsparcia Rolnictwa*, Biuro Analiz i Strategii.
- [319] Program Wieloletni 2011-2014, *IERiGŻ-PIB*, Warszawa, **2011**.
- [320] Rocznik statystyczny rolnictwa. *GUS*, Warszawa, **2005-2020**.
- [321] Rynek mleka. Stan i perspektywy. Analizy rynkowe, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej*, Warszawa, **2004-2013**.
- [322] Rynek mleka. Stan i perspektywy, *Analizy Rynkowe*, nr 46, 25-29, **2014**.
- [323] Rynek mleka. Stan i perspektywy, nr 53, *IERiGŻ-PIB, KOWR, MRiRW*, Warszawa, **2017**.
- [324] Rynek ziemi rolniczej stan i perspektywy. Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej - PIB, ARR, *Ministerstwo Rolnictwa i Rozwoju Wsi 2016*, **2016**.
- [325] Seremak-Bulge, J. Koncentracja produkcji mleka, Rynek mleka. Stan i perspektywy, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej*, Warszawa, **2003**.
- [326] Seremak-Bulge, J. Przetwórstwo mleka (w:) Rynek mleka stan i perspektywy, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej*, Warszawa, **2015**.
- [327] Seremak-Bulge, J. Restrukturyzacja produkcji mleka, Rynek mleka. Stan i perspektywy, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej*, Warszawa, **2013**.

- [328] Seremak-Bulge, J., Relacje cenowe na polskim rynku mleka po akcesji do Unii Europejskiej, *Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu*, Polityka Ekonomiczna 402, 359-367, **2015**.
- [329] Seremak-Bulge J. (red.), Rozwój rynku mleczarskiego i zmiany jego funkcjonowania w latach 1990–2005, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej - PIB*, **2005**.
- [330] Seremak-Bulge, J., Rynek mleka, wrzesień 2011, s.31, *Instytut Ekonomiki Rolnictwa i Gospodarki Żywnościowej*, Warszawa, **2012**.

Pozycje internetowe (netografie):

- [331] FAOSTAT, [online], <http://faostat.fao.org>, dostęp 8-12.02.2015.
- [332] Mleko naszym hitem eksportowym, <http://biznes.newsweek.pl/mleko-naszym-hitem-eksportowym,90249,1,1.html>/15.05.2014.
- [333] Murray, Ch.J., Cyclical Properties of Baxter-King Filtered Series, s.1, user www.servic e.emory.edu/~zliu5/seminars/murray.pdf, **2012**.
- [334] Weng, Z. An R Package for Continuous Time Autoregressive Models via Kaman Filter, Cran. Available online: r-project.org/web/packages/cts/vignettes/kf.pdf (accessed on 25 May 2012).
- [335] Woitek U., A Note on the Baxter-King Filter, s. 3, dspace.gla.ac.uk/bitstream/1905/593/1/9813.pdf, **2012**.

Akty prawne:

- [336] Decyzja Komisji z dn. 7 czerwca 1985 r. ustanawiająca wspólnotową typologię gospodarstw rolnych (85/377/ EWG), *Dz.U. L 220 z 178*, **1985**.
- [337] Dyrektywa 92/46/EWG z dnia 16 czerwca 1992 r. ustanawiająca przepisy zdrowotne dla produkcji i wprowadzania do obrotu surowego mleka, mleka poddanego obróbce termicznej i produktów na bazie mleka, *Dz.U. L 268*, z 14.9.1992, 1.

Spis ilustracji

1.1	Przykład modelu TAR na tle zbioru obserwacji w układzie współrzędnych (x_n, x_{n-1})	28
1.2	Przykład modelu STAR na tle zbioru obserwacji w układzie współrzędnych (x_n, x_{n-1})	29
1.3	Schemat Boxa-Jenkinsa do identyfikacji modeli	44
1.4	Schemat doboru modeli TAR	45
1.5	Przebieg notowań kursu dolara w 2015 roku	49
1.6	Zmiana procentowa notowań kursu USD w 2015 roku.	50
1.7	Wykres funkcji autokorelacji dla danych % zmiany kursu USD	51
1.8	Wykres częściowej funkcji autokorelacji dla danych % zmiany kursu USD	51
1.9	Wykres % zmian kursu USD i prognozy wyznaczonej za pomocą modeli AR(1)	53

1.10 Wykres % zmian kursu USD i prognozy wyznaczonej za pomocą modeli AR(20)	54
1.11 Wykres % zmian kursu USD i prognozy dla modeli ARMA(3,1)	54
1.12 Wykres % zmian kursu USD i prognozy wyznaczonej za pomocą modeli ARMA(3,2)	56
1.13 Prognoza X4 dla modelu ARIMAX (1,0,1).	59
1.14 Wykresy funkcji autokorelacji oraz autokorelacji częściowej	60
1.15 Wykres podziału danych wejściowych na klastry	60
1.16 Drzewo decyzyjne	61
1.17 Podział przestrzeni na zbiory rozmyte o funkcjach przynależności typu trójkątnego	62
1.18 Wartości prognozy (ANFIS) ciągu danych testowych na tle danych wejściowych przy funkcjach przynależności typu trójkątnego	62
1.19 Zmienność błędu w procesie uczenia prognozy dla danych testowych w systemie ANFIS (3 zbiory rozmyte).	63
1.20 Zmienność błędu w procesie uczenia prognozy dla danych testowych w systemie ANFIS (7 zbiorów rozmytych).	64
1.21 Przebieg notowań USD/PLN prognozowanych według modelu ARX w zależności od notowań euro.	66
1.22 Przebieg notowań EUR/PLN prognozowanych według modelu ARX w zależności od notowań dolara.	66
1.23 Przebieg notowań USD/PLN prognozowanych według modelu ARX w zależności od notowań PMP.	67
1.24 Przebieg notowań PMP prognozowanych według modelu ARX w zależności od notowań dolara.	67
1.25 Przebieg prognoz zmiennej OMP według modelu ARX w zależności od zmiennej X4.	68
1.26 Schematyczny rysunek neuronu biologicznego [48]	73
1.27 Matematyczny model pojedynczego neuronu [15]	74
1.28 Schemat sztucznej sieci neuronowej [16]	75
1.29 Przebieg powszechnie stosowanych funkcji aktywacji wraz z ich gradientem	76
1.30 Przebieg powszechnie stosowanych funkcji aktywacji wraz z ich gradientem	77
1.31 (a) Typowa topologia połączeń sieci wielowarstwowej, (b) topologia zoptymalizowana dzięki technice dropout	86
1.32 (1-3) Wstępne szkolenie kolejnych warstw, (4) douczenie całości modelu . .	87
1.33 Prognozowanie z użyciem sztucznych sieci neuronowych	88
1.34 Przykładowa architektura sieci	91

1.35	Architektura autokodera	95
1.36	Struktura modelu ograniczonej maszyny Boltzmannna	96
1.37	Zastosowanie łańcucha Markowa w algorytmie <i>Contrastive Divergence</i>	99
1.38	Architektura spłotowej sieci neuronowej rozpoznającej litery [119]	100
1.39	Warstwy łączące redukują wymiar danych poprzez ekstrakcję prostych statystyk z map aktywacji	101
1.40	Sieci rekurencyjne różnią się pętlami łączącymi wyjście danego neuronu z jego wejściem	102
1.41	Głębokie sieci neuronowe są zbudowane z większej liczby warstw niż sieci klasyczne	102
1.42	Ekstrakcja i przetwarzanie cech sygnału dwuwymiarowego przez głęboką sieć neuronową	104
1.43	Schemat blokowy filtru cyfrowego („tylko” NOI/IIR, AR - <i>Autoregressive</i> - ang.) dla $N = 3$ z linią opóźniającą tylko na wyjściu	110
1.44	Schemat blokowy filtru Kalmana	111
1.45	Porównanie algorytmów próbkowania oszczędnego (b) do podejścia klasycznego (a)	115
1.46	a) sygnał o wartościach niezerowych w przedziale $[-3; 3]$, b) sygnał rzadki, c) sygnał prawie rzadki	118
2.1	Odszumianie danych pomiarowych	125
2.2	Schemat ideowy zastosowanego algorytmu predykcyjnego	127
2.3	Wartość średnia mocy widma procesu jednowymiarowego dysku Poissona. Proces ma związek z widmem szumu niebieskiego	130
2.4	Architektura spłotowej sieci neuronowej w proponowanym rozwiązaniu	134
2.5	Kształtowanie się średniego absolutnego błędu procentowego (MAPE) oraz procentowego błędu średniokwadratowego (MSPE) dla analizowanych metod predykcji	143
2.6	Odszumione dane przebiegu czasowego cen OMP (linia przerywana) oraz dane obciążone (linia ciągła)	144
2.7	Wartość D_{stat} dla poddanych testom algorytmów predykcji od 1 do 5 kroków naprzód	154
2.8	Wartość RMSE dla poddanych testom algorytmów predykcji od 1 do 5 kroków naprzód	160

Spis tabel

1.1	Tabela porównawcza modeli parametrycznych	30
-----	---	----

1.2	Wyniki testu ADF dla ciągu danych notowań zmiany kursu USD	50
1.3	Wyniki testu ADF dla ciągu % zmiany kursu USD	50
1.4	Parametry modelu ARIMA(3,1,1)	52
1.5	Porównanie parametry modeli ARMA(3,1) i ARMA(3,2)	55
1.6	Kryteria informacyjne dla modelu ARIMAX (1,0,1)	58
1.7	Zależność błędów od ilości zbiorów rozmytych	65
2.1	Odszumianie szeregu czasowego oparte na metodologii próbkowania oszczęd- nego.	129
2.2	Wartości błędów predykcji dla wybranych metod prognozowania wartości cen OMP	145
2.3	Punktowa ocena wybranych metod predykcji	146
2.4	Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 1 krok naprzód. *Algorytm proponowany przez autora niniejszej pracy	149
2.5	Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 2 kroki naprzód. *Algorytm proponowany przez autora niniejszej pracy	150
2.6	Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 3 kroki naprzód. *Algorytm proponowany przez autora niniejszej pracy	151
2.7	Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 4 kroki naprzód. *Algorytm proponowany przez autora niniejszej pracy	152
2.8	Porównania D_{stat} dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 5 kroków naprzód. *Algorytm propono- wany przez autora niniejszej pracy	153
2.9	Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 1 krok naprzód. *Algorytm proponowany przez autora niniejszej pracy	155
2.10	Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 2 kroki naprzód. *Algorytm proponowany przez autora niniejszej pracy	156
2.11	Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 3 kroki naprzód. *Algorytm proponowany przez autora niniejszej pracy	157

2.12	Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 4 kroki naprzód. *Algorytm proponowany przez autora niniejszej pracy	158
2.13	Porównania RMSE dla autorskiego algorytmu i rozwiązań konkurencyjnych dla przypadków prognozowania o 5 kroków do przodu. *Algorytm proponowany przez autora niniejszej pracy	159