ENHANCING FINANCIAL MARKET FORECASTING USING DEEP LEARNING AND COMPUTER VISION-BASED TECHNICAL ANALYSIS

2025
PhD. THESIS
COMPUTER ENGINEERING

Edrees Ramadan MERSAL MORCELI

Thesis Advisor
Assoc. Prof. Dr. Hakan KUTUCU

ENHANCING FINANCIAL MARKET FORECASTING USING DEEP LEARNING AND COMPUTER VISION-BASED TECHNICAL ANALYSIS

Edrees Ramadan MERSAL MORCELI

Thesis Advisor
Assoc. Prof. Dr. Hakan KUTUCU

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering
Prepared as
PhD. Thesis

KARABUK
April 2025

I certify that in my opinion, the thesis submitted by Edrees Ramadan MERSAL MORCELI titled “ENHANCING FINANCIAL MARKET FORECASTING USING DEEP LEARNING AND COMPUTER VISION-BASED TECHNICAL ANALYSIS” is fully adequate in scope and quality as a thesis for the degree of PhD.

Assoc. Prof. Dr. Hakan KUTUCU ....
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a PhD thesis. 25-04-2025

Examining Committee Members (Institutions) Signature

Chairman : Prof. Dr. İlker TÜRKER (KBU) ....

Member : Assoc. Prof. Dr. Hakan KUTUCU (KBU) ....

Member : Assoc. Prof. Dr.Ümit ATİLA (GU) ....

Member : Assist. Prof. Dr. Kürşat Mustafa KARAOĞLAN (KBU) ....

Member : Assist. Prof. Dr. Bayram AKGÜL (BARU) ....

The degree of PhD by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Assoc. Prof. Dr. Zeynep ÖZCAN ....
Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Edrees Ramadan MERSAL MORCELI

ABSTRACT

PhD. Thesis

ENHANCING FINANCIAL MARKET FORECASTING USING DEEP LEARNING AND COMPUTER VISION-BASED TECHNICAL ANALYSIS

Edrees Ramadan MERSAL MORCELI

Karabük University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assoc. Prof. Dr. Hakan KUTUCU
April 2025, 102 pages

Over the centuries, Japanese candlestick (JC) patterns have garnered considerable interest from market participants because of their capacity to illuminate the underlying psychology of financial markets. Originating in 17th-century Japan, where rice traders first developed this illustrative method of charting—candlestick patterns, it evolved into a cornerstone of contemporary technical analysis, shaping countless trading strategies across global markets. The present investigation advances this venerable tradition by proposing an innovative framework that leverages convolutional neural networks (CNNs) to forecast the directional movements of subsequent candlesticks. This comprehensive study underscores the critical role of historical pattern recognition in anticipating price fluctuations, substantively contributing to the literature on algorithmic trading and intelligent financial forecasting.

Methodologically, this research employed a meticulous three-tiered approach to construct a high-fidelity dataset. First, the raw price data were parsed into subcharts using a sliding window technique, which enabled the capture of temporal dynamics within bounded intervals. Next, the Ta-lib library was harnessed to validate the presence of predefined candlestick patterns within each subchart, thereby infusing the dataset with qualitative indicators of market sentiment and potential trend reversals. Finally, each window's directional inclination was determined through an integrative process involving technical indicators such as moving averages that collectively furnished robust evidence regarding prospective price trajectories. These carefully curated data elements form a solid foundation for the subsequent model training phase, wherein a CNN is developed to automatically extract salient visual and temporal features, ultimately achieving an impressive predictive accuracy of up to 99.3 percent.

Crucially, the reliability and generalizability of these findings were verified using a rigorous cross-validation scheme. In this validation protocol, the dataset is split into multiple distinct subsets, and iterative rounds of model training and testing are conducted on various combinations of these partitions. Through this systematic procedure, the capacity of the model to generalize was ascertained, thereby diminishing the risk of overfitting and solidifying the robustness of the observed predictive performance. The consistently high accuracy rates recorded across different training–testing folds testify to the adaptability of this approach in real-world trading environments.

From a broader perspective, this study demonstrates how modern artificial intelligence techniques can augment, refine, and potentially supersede traditional technical analysis methods. By blending the timeless insights of candlestick charting with the computational power of CNNs, traders and analysts can derive more nuanced data-driven strategies designed to anticipate inflection points and capitalize on emerging trends.

Keywords **:** Stock market, Buy-sell strategy, CNN, Cross-validation, Japanese candlestick**.**

Science Code : 915.1.092

ÖZET

Doktora Tezi

DERİN ÖĞRENME VE BİLGİSAYARLİ GÖRÜ TABANLI TEKNİK ANALİZ KULLANARAK FİNANSAL PİYASA TAHMİNLERİNİN GELİŞTİRİLMESİ

Edrees Ramadan MERSAL MORCELI

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Doç. Dr. Hakan KUTUCU

Nisan 2025, 102 sayfa

Yüzyıllar boyunca Japon mum (JC) formasyonları, finansal piyasaların temel psikolojisini aydınlattıkları düşüncesiyle yatırımcıların yoğun ilgisini çekmiştir. İlk kez 17. yüzyılda Japonya'daki pirinç tüccarları tarafından geliştirilen bu gösterim yöntemi, günümüz teknik analiz yaklaşımlarının temel direklerinden biri haline gelerek küresel piyasalarda sayısız yatırım stratejisine esin kaynağı olmuştur. Bu çalışma, söz konusu köklü geleneği ileriye taşıyarak, gelecek mum çubuklarının yönelimlerini tahmin edebilmek amacıyla evrişimli sinir ağlarını (CNN) kullanan yenilikçi bir çerçeve önermektedir. Araştırma kapsamında, tarihi formasyon tespitinin fiyat oynaklığını öngörmedeki önemini vurgulayarak, algoritmik alım-satım ve yapay zekâ odaklı finansal tahmin literatürüne kayda değer bir katkı sunulmaktadır.

Yöntemsel olarak, yüksek doğruluklu bir veri kümesi oluşturmak amacıyla üç aşamalı titiz bir yaklaşım benimsenmiştir. İlk aşamada, kayan pencere yöntemi aracılığıyla ham fiyat verileri alt grafiklere ayrılmış ve böylece zamana yayılan dinamikler belirli aralıklar içinde yakalanmıştır. İkinci aşamada ise Ta-lib kütüphanesi kullanılarak, önceden tanımlanmış mum formasyonlarının her alt grafikte mevcut olup olmadığı doğrulanmış ve veri kümesine piyasa duyarlılığı ile muhtemel trend dönüşlerini yansıtan nitel göstergeler eklenmiştir. Son aşamada, her pencerenin yönsel eğilimi, hareketli ortalamalar ve momentum osilatörleri gibi teknik göstergelerin bütünsel bir şekilde değerlendirilmesiyle belirlenmiştir. Bu özenli veri işleme süreci, önemli görsel ve zamansal özellikleri otomatik olarak çıkaran bir CNN modelinin eğitimi için sağlam bir temel oluşturmuş ve neticede %99,3'e varan etkileyici bir tahmin doğruluğu elde edilmiştir.

Modelin güvenilirliği ve genellenebilirliği, çoklu çapraz doğrulama yöntemiyle pekiştirilmiştir. Bu doğrulama sürecinde, veri kümesi birden fazla farklı alt kümeye ayrılmış ve eğitme ile test aşamaları farklı bölümler üzerinde tekrarlı olarak gerçekleştirilmiştir. Böylece, modelin aşırı uyum (overfitting) riskini en aza indirgeme ve öngörü performansını güçlendirme amacına hizmet eden sistematik bir yöntem devreye sokulmuştur. Farklı eğitim–test katmanlarında elde edilen sürekli yüksek doğruluk oranları, yaklaşımın gerçek piyasa koşullarına uyarlanabilirliğine dair güçlü kanıtlar sunmaktadır.

Daha geniş bir açıdan bakıldığında, bu araştırma, yapay zekâ temelli tekniklerin geleneksel teknik analiz yöntemlerini nasıl güçlendirebileceğini ve hatta ileride yer yer geçersiz kılabileceğini göstermektedir. Mum çubuğu analizinin zamansız içgörüleri, CNN'nin hesaplama gücüyle harmanlanarak, yatırımcıların kritik dönüş noktalarını öngörebileceği ve gelişmekte olan trendlerden yararlanabileceği daha incelikli, veri odaklı stratejiler geliştirmelerine olanak tanımaktadır.

Anahtar Sözcükler **:** Borsa, Al-sat stratejisi, CNN, Çapraz doğrulama, Japon mum çubuğu.

Science Code **: 915.1.092**

ACKNOWLEDGMENT

In the name of Allah, the Most Gracious, the Most Merciful.

First and foremost, I am profoundly grateful to Allah Almighty for granting me the strength, perseverance, and wisdom to pursue and successfully complete my doctoral studies.

I extend my heartfelt appreciation to my beloved family for their unwavering patience, sacrifices, and unconditional support throughout this journey. Their encouragement and spiritual guidance have been invaluable, and I am deeply indebted to my mother and all my family members for their enduring belief in me.

I wish to express my sincere gratitude to my supervisor, Assoc. Prof. Dr. Hakan KUTUCU, for his continuous support, insightful guidance, and steadfast motivation. His expertise and constructive feedback have been instrumental in shaping my research and enhancing my academic growth.

I would also like to express my special thanks to Assist. Prof. Dr. Kürşat Mustafa KARAOĞLAN for his valuable support and contributions. His guidance and encouragement have significantly enriched my work.

My appreciation also extends to the esteemed faculty and administrative staff of the Computer Engineering Department at Karabuk University. I am particularly thankful to the professors who contributed to my academic development through their teaching, mentorship, and collaborative spirit.

Finally, I am grateful to my friends and colleagues for their encouragement, thoughtful discussions, and companionship throughout this endeavor. Their support has made this journey more enriched and memorable. discussions, and companionship throughout this endeavor. Their support has made this journey more enriched and memorable.

CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

LIST OF TABELS

Page

SYMBOLS AND ABBREVITIONS INDEX

ABBREVITIONS

ACF	: Autocorrelation Function
FA	: Fundamental Analysis
TA	: Technical Analysis
JC	: Japanese Candlestick
CNN	: Convolutional Neural Network
CNNs	: Convolutional Neural Networks
RNN	: Recurrent Neural Network
RNNs	: Recurrent Neural Networks
LSTM	: Long Short-Term Memory
ML	: Machine Learning
OHLC	: Open, High, Low, Close (price data)
SMA	: Simple Moving Average
MACD	: Moving Average Convergence Divergence
ARIMA	: AutoRegressive Integrated Moving Average
ARMA	: AutoRegressive Moving Average
PACF	: Partial Autocorrelation Function
DDQN	: Double Deep Q-Network
YOLO	: “You Only Look Once” (an object-detection algorithm)
GAF	: Gramian Angular Field
SSA	: Sparrow Search Algorithm (used in one cited study)
BiGRU	: Bidirectional Gated Recurrent Unit
RMSE	: Root Mean Squared Error
MAPE	: Mean Absolute Percentage Error
SMOTE	: Synthetic Minority Over-sampling Technique
XGBoost	: eXtreme Gradient Boosting
K-Fold	: K-Fold Cross Validation (used for training/validation split

PART 1

INTRODUCTION

1.1. BACKGROUND

Forecasting underpins strategic decision-making in financial markets. Through advanced predictive models, traders and institutions endeavor to anticipate future scenarios, mitigate risks, and optimize investment outcomes. Thus, effective market analysis relies on a solid grasp of historical price behavior, mastery of analytical frameworks, and proficiency with technical tools.

Broadly speaking, market analysis can be categorized into fundamental analysis (FA) and technical analysis (TA). An FA entails an in-depth review of an entity's financial statements, assessing revenue streams, cost structures, and overall financial performance to derive its intrinsic value. By contrast, TA employs historical market data and statistical patterns to forecast price movements [1]. Specifically, TA leverages insights into price actions, trading volumes, and technical indicators to detect signals of pending shifts in market direction.

Financial markets have evolved over centuries, shaping global economic development and enabling capital formation. Although rudimentary trade mechanisms existed in ancient Mesopotamia, more formalized stock exchanges emerged in the 17th century. The Amsterdam Stock Exchange (1602) was among the first to trade equity securities, most notably for the Dutch East India Company, and it marked a seminal milestone in modern finance.

Within the TA, candlestick charting is particularly important because of its visual depth and historical significance. Originating in 17th-century Japan, candlestick techniques gained prominence when rice traders observed recurring formations that signaled probable price reversals [2] . Introduced to Western markets in the 1990s, candlestick patterns such as hammer (indicating potential bullish reversals) and doji (suggesting indecision) have become indispensable across trading platforms [3] & [4] & [5]. Each candlestick represents four primary pieces of information: opening, closing, and the highest and lowest prices over a specified interval. The body denotes the range between the opening and closing prices, whereas the upper and lower shadows indicate high and low values, respectively, for that period [6], as shown in Figure 1.1.

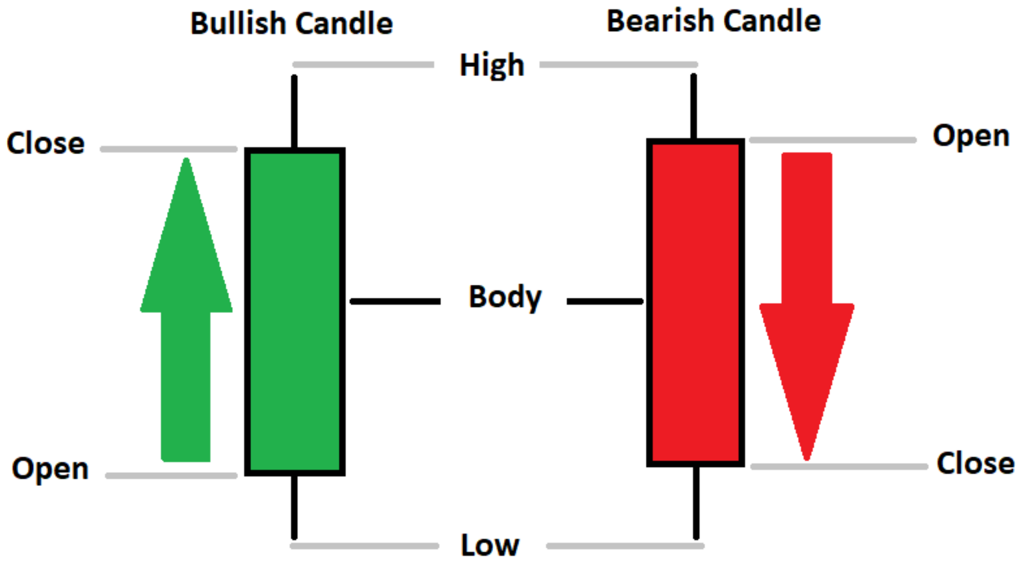


Figure 1.1. The components for bullish and bearish candles

Despite its utility, the identification of candlestick formation poses statistical and operational complexities [7]. Short-term fluctuations and noise can distort pattern recognition, whereas overlapping formations complicate systematic analysis. The exponential growth of the possible geometric variations adds to this difficulty. Furthermore, aligning historical candlestick patterns with current market data remains challenging, given the emergence of new serial formations and the constant evolution of price behaviors [8] .

Crucially, candlestick patterns derive greater significance when analyzed as part of a sequence rather than in isolation. Grouped candlesticks may form shapes resembling everyday objects such as a hammer or doji that signal potential reversals or continuations. These visual cues help traders make informed decisions about buying and sell orders [9] . For instance, doji highlights the balance of power between buyers and sellers, frequently presaging a trend reversal.

Recent advances in machine learning (ML) have further enhanced technical analysis. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and other modified architectures show promise in detecting and categorizing price variations [10]. Most current research involving candlestick charts focuses on technical indicators, such as candlestick color, shadow length, and the collective shape of candlesticks, to build predictive models. Typically, these data are converted into tensors or vectors, allowing numerical representation of the chart patterns. However, such vectorization may underutilize the visual context embedded in candlestick charts, underscoring the potential of image-based ML methodologies to capture both numeric and graphical dimensions of market data.

In summary, candlestick charting provides a rich tapestry of historical insights and future forecasting potential, particularly when applied alongside robust ML algorithms. Analysts can derive more comprehensive and accurate assessments of market dynamics by integrating domain knowledge from TAs with advanced neural network architectures, thereby facilitating better-informed trading strategies and investment decisions.

1.2. CONTEXT AND MOTIVATION

1.2.1. Context and Motivation: Leveraging Candlestick Patterns and Computer Vision in Stock Market Analysis

Candlestick charts and their underlying patterns have long served as a cornerstone of technical analysis, providing traders with a visually intuitive means of interpreting market sentiments. By capturing four key price points, opening, closing, high, and low, each candlestick conveys both intra-period market dynamics and broader trend information. When multiple candlesticks form recognizable patterns (Doji, Hammer, Engulfing), they often signal pivotal market shifts, offering traders actionable insights into potential reversals or continuations.

1.2.2. Ta-lib Integration for Pattern Detection

The Ta-lib library further enhances the utility of candlestick charts by automating the pattern identification. Instead of manually scanning hundreds or thousands of candles, researchers and traders can employ the built-in functions of Ta-lib to systematically detect established patterns (Morning Star and Three White Soldiers). Automation significantly reduces human error and subjectivity, enabling more robust and repeatable strategies. Moreover, Ta-lib’s straightforward integration with popular programming languages (such as Python) streamlines data processing, making it simpler to incorporate candlestick pattern recognition into the existing trading algorithms and research pipelines.

1.2.3. Computer Vision for Next-Candle Prediction

While Ta-lib assists in identifying patterns, computer vision techniques offer an additional layer of insight by analyzing the visual features of candlestick charts. For instance, Convolutional Neural Networks (CNNs) can learn high-level representations of chart movements that extend beyond the preset rules or signal thresholds. By processing candlestick formations as images rather than numeric series, computer vision models can capture nuances such as slight variations in candle bodies or shadows, which might go unnoticed in purely numerical approaches.

However, computer vision applications in financial markets remain relatively underexplored compared with other fields, such as healthcare or autonomous driving, where image-based models have already achieved remarkable results. The high volatility and noise inherent in stock market data present unique challenges, including non-stationarity and regime shifts. These factors underscore the need for further research and specialized model adaptations that account for financial-specific complexities such as sudden macroeconomic events or institutional trading behaviors.

1.3. PROBLEM STATEMENT

Despite the long-recognized usefulness of Japanese Candlestick (JC) patterns in signaling potential market shifts, accurately predicting the direction of the next candlestick whether bullish (up) or bearish (down) remains a significant challenge. Existing technical analysis methods and earlier-generation machine learning models often struggle to accommodate the high volatility and diverse external forces that shape financial markets, leading to inconsistent or incomplete forecasts. Moreover, many prior studies have focused only on a handful of these patterns, overlooking the comprehensive set of 61 JC patterns available through the Ta-lib library. This narrow scope limits predictive power and fails to leverage modern data-driven approaches.

To address these gaps, this research proposes employing computer vision techniques to interpret Japanese Candlestick charts as full-fledged images rather than simple numeric series. By capturing visual nuances and leveraging the entire range of recognized patterns, we aimed to construct an automated system that classifies the next candle as upward or downwards with greater accuracy. This approach requires seamless integration of real market data, multiple timeframes, and advanced technical indicators such as moving averages to ensure a robust trend correlation. Through this holistic methodology, we seek to overcome current limitations and offer a more reliable and comprehensive framework for next-candle-direction forecasting.

1.4. AIM OF THE STUD

This study aims to design and implement a robust candlestick pattern recognition framework that leverages Convolutional Neural Networks (CNNs) to enhance market trend forecasting. By integrating deep learning techniques with the established practice of candlestick analysis, this study seeks to

A comprehensive dataset, featuring 61 distinct candlestick patterns, was constructed. A sliding window technique was employed to segment the data into subcharts, and a technical analysis library was used to validate the identified formations.

Develop and train a CNN-based predictive model to classify these patterns as bullish or bearish, incorporating additional technical indicators such as Simple Moving Averages (SMAs) to reinforce trend validation.

Evaluation of model performance across various time windows, shift sizes, and validation methods. The goal is to surpass conventional forecasting techniques, achieve high predictive accuracy, and present a dependable framework for market prediction.

1.5. CONTRIBUTIONS

- Extensive Candlestick Coverage

This study encompasses 61 distinct candlestick patterns, significantly more than many earlier studies, thus offering a more granular view of potential signals in the market.

- Structured Data Preparation

By combining a sliding window approach with technical indicator validation, this method ensures that each subchart is accurately labeled, aligning chart appearance with actual price dynamics.

- High-Precision Forecasting

The proposed CNN model consistently demonstrates high accuracy (up to 99.3% in certain configurations), outperforming the existing methods reported in the literature, which often exceed 92%.

- Robust Validation

Cross-validation strategies were used to verify the consistency and reliability of the model. Multiple temporal configurations with varying window sizes and shifts were evaluated systematically.

- Practical Applicability

Although designed and tested on forex data, the framework can be adapted to other markets and time frames. It offers a scalable pathway toward real-time automated candlestick analysis, equipping traders and analysts with a powerful tool for informed decision-making.

1.6. THESIS OUTLINE

The present study was conducted in this context. The remainder of this dissertation is structured as follows:

Chapter 1

This chapter presents an overview of market forecasting, particularly focusing on Japanese Candlestick (JC) patterns and deep learning techniques. This highlights the research problem, principal objectives, and specific contributions of the study.

Chapter 2

We reviewed the literature on candlestick pattern analysis, technical indicators, and deep-learning-based predictive models in financial markets. This chapter discusses relevant studies, identifies gaps, and situates our research within the existing body of knowledge.

Chapter 3

The concepts, methods, and algorithms employed for candlestick chart detection and next-candle-direction prediction are explained in detail. This includes an in-depth discussion of Convolutional Neural Networks (CNNs), sliding window techniques, and the role of Ta-lib in automated pattern identification.

Chapter 4

The implementation details of the proposed methodology are proposed, outlining the data preprocessing steps, model architecture, and training process. Technical specifics, such as hyperparameter tuning, validation strategies, and the experimental setup, are also discussed.

Chapter 5

Analysis of the experimental results Metrics such as accuracy, precision, and recall are presented alongside visual aids (confusion matrices and ROC curves) to evaluate model performance and justify its effectiveness.

Chapter 6

This dissertation is excluded by summarizing the key findings and contributions. This chapter also introduces future research perspectives, highlighting potential extensions and improvements to the current work.

PART 2

LITRATURE SURVEY

2.1. HISTORICAL REVIEW

Financial market forecasting and candlestick analysis are inherently dynamic and are influenced by economic, political, and behavioral factors that can trigger rapid price fluctuations [11]. Traditional analytical techniques, such as fundamental and statistical modeling, often struggle to capture the complexity of these fluctuations. Consequently, researchers have increasingly turned to Japanese Candlestick (JC) patterns, a charting method with roots in 17th-century rice trading, to gain insights into potential price reversals, continuations, or sentiment shifts [12], [13]. These patterns, ranging from Doji and Hammer to more complex formations, such as Kicking and Mat Hold, have been extensively used in technical analyses to anticipate future trend movements [9].

2.1.1. Prior Studies on Candlestick Pattern Recognition

Numerous studies have highlighted candlestick-based approaches, each focusing on different aspects of pattern identification, predictive modeling, and trade decision-making. [14] integrated object detection methods and Gramian Angular Field (GAF) encoding to detect chart patterns and examine their correlation with price movements. [15] adapted a YOLO-based architecture for candlestick detection, illustrating how time-series encoding and deep neural networks can facilitate pattern categorization. Similarly, [16] built a candle-pattern recognition system to classify daily candlestick types (24 distinct patterns) and then used XGBoost to predict directional trends. Many of these studies underscore the importance of analyzing multiple patterns, often beyond a handful, to capture a richer set of market signals [13].

2.1.2. Deep Learning Approaches for Market Trend Prediction

The rapid proliferation of Artificial Intelligence (AI) and data availability has prompted researchers to explore deep learning models, notably Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and LSTMs, for market prediction.

2.1.2.1. CNN-Based Methods:

A growing body of research uses CNNs on candlestick chart images to detect spatial features. [17] utilized CNNs within a Double Deep Q-Network framework to interpret candlestick patterns, achieving better returns than a benchmark S&P 500 strategy. [18] divided candlestick charts into subparts and employed CNN autoencoders to extract features before using RNNs for final predictions.

2.1.2.2. RNN and LSTM Approaches

Various researchers [19] have integrated LSTMs or RNNs to handle sequential dependencies in candlestick data. Although these architectures effectively model temporal patterns, they sometimes require larger datasets and meticulous hyperparameter tuning to avoid vanishing or exploding the gradients.

2.1.2.3. Hybrid and Ensemble Models

Several studies merged CNNs with LSTMs or ensemble algorithms (e.g., XGBoost, Random Forest) to capture both the spatial patterns in candlestick images and the temporal nuances of time-series data [20], [21].

2.1.3. Integrating Technical Indicators and Feature Engineering

For instance, [22] combined daily candlestick patterns with multiple indicators in a deep neural network to enhance intraday price forecasts. Similarly, [21] revealed that certain candlestick formations used in tandem with indicators or substitutes improved the predictive performance of crypto-asset forecasts.

Feature Extraction Techniques

Image-based: Transforming candlestick charts into image data (e.g., GAF or wavelet-transformed charts) enables CNNs to detect geometric and textural features that numeric-only approaches can overlook [7], [23]

Vector-based: Some models rely on numeric encoding of candlestick attributes (body length and shadow ratios) and technical indicators to feed into conventional ML or hybrid neural networks ([6].

2.1.4. Comparative Insights on Pattern Recognition Techniques

Research comparing classical approaches (e.g., manual chart scanning and rule-based detection) with AI-driven methods has concluded that deep learning models generally outperform simpler systems, albeit at a higher computational cost [9]. [24] noted that MLP and CNN performed better than random forests and AdaBoost in certain candlestick tasks but cautioned that ensemble methods such as XGBoost can surpass deep networks if the data are imbalanced, or patterns are rare.

The key limitations identified in prior studies include the following.

Data Imbalance and Noise: Less frequent patterns, such as the Kicker or Concealing Baby Swallow, can be overlooked in training and skewing predictions [25].

High-Frequency Volatility: Standard models may struggle in real-time contexts with abrupt price swings unless a specialized architecture is used.

Overfitting Risks: Deep models trained on limited datasets can latch onto noise unless robust validation (e.g., stratified k-fold) or regularization is enforced.

2.2. RELATED WORKS

In recent years, the integration of artificial intelligence with traditional technical analysis has transformed financial forecasting. A significant body of research has focused on leveraging Japanese Candlestick (JC) patterns, which encapsulate key price points, such as opening, closing, high, and low values, to extract actionable market signals. Early studies by Andriyanto (2020) [26] demonstrated that image-based approaches using Convolutional Neural Networks (CNNs) can effectively predict short-term directional movements by analyzing three-day candlestick sequences, highlighting the potential of deep learning for automating what was once a manual process.

Barra et al. (2020) [23] extended this idea by employing an ensemble of CNNs on images generated via Gramian Angular Field (GAF) transformations. Their multiresolution approach to Standard & Poor’s 500 index data revealed that such visual encoding techniques can capture subtle geometrical features in candlestick charts, resulting in a robust predictive performance that outperforms traditional buy-and-hold strategies.

In the context of real-time object detection, Birogul et al. (2020) [27] applied the YOLO algorithm to 2D candlestick charts to extract buy–sell signals. Their work demonstrated how deep learning can shift traditional technical analysis toward more automated, visually driven systems, offering traders a dynamic decision-making tool.

Efforts to integrate reinforcement learning with candlestick analysis are exemplified by Brim and Flann (2022), [17] who combined a CNN with a Double Deep Q-Network (DDQN) to capture and act upon features extracted from recent candles. Their model, trained on S&P 500 data, consistently produced superior returns, thereby illustrating the effectiveness of hybrid approaches in volatile markets.

To address the challenges of noise and false signals, Cagliero et al. (2023) [28] proposed a framework that separates machine learning from the pattern recognition process. They developed a more reliable system for market forecasting under volatile conditions by fine-tuning the classification pipeline to reduce redundant and conflicting signals.

Two influential studies by Chen and Tsai (2020, 2022) [25], [29] paved the way for modern candlestick analysis using image-based deep learning. In their 2020 study, they converted time-series candlestick data into GAF images for classification using CNNs, demonstrating improved accuracy over traditional LSTM models. Their 2022 study builds on this foundation by adapting a YOLO-based architecture to dynamically detect multiple candlestick patterns, underscoring the evolving nature of visual pattern recognition in financial contexts.

Chen et al. (2024) [30] integrated candlestick pattern inputs with a Sparrow Search Algorithm (SSA) and a BiGRU network, resulting in a system that significantly reduced forecasting errors (e.g., lower MAPE and RMSE) in Chinese stock markets. This study exemplifies how evolutionary optimization can enhance deep learning models for financial predictions.

Exploring the profitability of classical patterns, Cohen (2020) [11] investigated formations such as “stairs” and Harami. His findings reveal that, while some patterns consistently yield profits, others do not, emphasizing the need for models that can differentiate between robust and less reliable signals.

Dakalbab et al. (2024) [11] provide context by linking macroeconomic and political factors to trader sentiment, suggesting that external events can significantly influence candlestick patterns. This underscores the challenge of isolating purely technical signals in an environment characterized by frequent disruptions.

To mitigate noise in high-frequency data, Du et al. (2020) [7] applied wavelet transforms to denoise log returns before feeding the cleaned data into CNNs for classification. Their method effectively enhanced the signal-to-noise ratio, thereby improving the reliability of intraday trading of candlestick patterns.

Focusing on short-term predictive power, Heinz et al. (2021) [31] evaluated the conditions under which bullish and bearish engulfing patterns performed reliably. They noted that specific price criteria are critical for maximizing accuracy, although some formations, such as Harami, remain inconsistent.

Hung and Chen (2021) [18] introduced a multistage approach in which candlestick charts were first segmented into subcharts, processed via a CNN autoencoder, and then analyzed using a recurrent neural network (RNN) for sequential prediction. Their method achieved higher accuracy on the Taiwan Exchange data by effectively capturing both the spatial and temporal features.

The “Deep Candlestick Predictor” developed by Hung et al. (2020) [9] further demonstrates that CNN autoencoders can extract vital local features from candlestick subcharts, which, when combined with 1D-CNN layers, lead to more robust trend forecasts compared to traditional methods.

In a comparative analysis, Jearanaitanakij and Passaya (2019) [4] collected over 1,800 labeled candlestick images to train a dedicated CNN model. Their findings suggest that custom-designed CNNs can offer faster training and higher accuracy for classifying candlestick charts into bullish, bearish, or sideways categories.

To address the issue of imbalanced datasets, Karmelia et al. (2022) [32] employed techniques, such as undersampling and SMOTE, to better represent rare candlestick patterns within their Feed-forward Neural Network framework. Although the overall classification accuracy is high, the study reveals challenges in achieving a balanced F1 score, indicating that further refinement is required to capture less frequent but critical patterns.

A comparative study by Kusuma et al. (2019) [33] investigated the influence of chart image dimensions and feature sets on prediction accuracy. Their evaluation across Taiwanese and Indonesian stock markets demonstrated that choices regarding image resolution and the inclusion of volume data significantly affected model performance, highlighting the sensitivity of deep learning models to input parameter configurations.

Lee et al. (2019) [34] merge CNNs with deep Q-networks to approximate stock chart images and generate trading signals across multiple international markets. Their cross-market validation showed that image-based reinforcement learning strategies can effectively adapt to diverse global market conditions.

Liang et al. (2022) [20] propose a method that mines sequential patterns from multidimensional candlestick data, introducing a novel similarity measure to align new sequences with historical patterns. This method shows that combining geometric pattern recognition with temporal sequence alignment can enhance the adaptability of forecasting models under dynamic market conditions.

An ensemble approach by Lin et al. (2021) [6] automatically selects optimal prediction methods, such as random forest, gradient boosting, and LSTMto, to forecast daily candlestick patterns. Their findings suggest that integrating various models guided by both technical indicators and candlestick features leads to improved risk-adjusted returns compared to traditional strategies.

Madbouly et al. (2020) [35] explore an alternative representation by integrating Heikin-Ashi candlesticks with fuzzy time series and a cloud model. Their method smooths volatile data and manages uncertainty through fuzzy logic, yielding higher forecast stability and accuracy in markets with significant noise.

For intraday forecasting, Naik and Mohan (2020) [22] integrated multiple technical indicators with candlestick chart data within a deep neural network framework. Their method improves upon standard numeric-only models by providing a richer feature set that captures both the directional movement and underlying momentum of the market.

Nakayama et al. (2019) [36] address the incorporation of order-flow information into candlestick-like images. By encoding market microstructure events and using logistic regression combined with CNNs, they showed that granular trading details can be effectively visualized and used to predict short-term price trends.

Nguyen et al. (2023) [14] merge object detection with GAF time-series encoding to identify complex candlestick formations, including overlapping patterns and multiple chart events. Their advanced detection algorithm increases the detection rate of nuanced candlestick features, underscoring the capability of modern computer vision techniques to manage complex financial data.

Orquín-Serrano (2020) [37] targeted the adaptive classification of candlestick patterns for the EUR–USD pair using statistical inference methods. Despite sophisticated labeling, this study reveals that net returns may remain suboptimal when transaction costs are factored in, highlighting the inherent challenges of relying solely on candlestick data under certain market conditions.

Orte et al. (2023) [21] compare input strategies candlestick-only, technical indicators-only, and a combination using a random forest model for crypto futures markets. Their iterative retraining approach demonstrated that candlestick patterns can offer critical insights and often outperform traditional indicators under volatile conditions.

Pan et al. (2020) [38] develop a deep learning portfolio model by transforming candlestick data into images and using convolutional autoencoders to extract latent features. By clustering these features and selecting stocks based on Sharpe ratios, this study demonstrates that sophisticated image-based feature engineering can support effective portfolio management.

Ramadhan et al. (2022) [39] proposed a hybrid CNN-LSTM model in which CNN layers capture spatial candlestick features and LSTM layers process temporal dependencies. This combined approach yields robust accuracy in short-term predictions, even amidst high market volatility, thereby demonstrating the benefits of merging spatial and sequential deep learning techniques.

Santur (2022) [16] employed a multiphase process that classifies 24 candlestick patterns using one-hot encoding and then applies XGBoost for trend prediction. Testing various global indices, this method consistently outperformed traditional buy-

and-hold strategies, illustrating the efficacy of ensemble learning in capturing dynamic market signals.

Thammakesorn and Sornil (2019) [2] proposed a feature-based extraction method that integrates candlestick pattern signals with a Chi-square Automatic Interaction Detector to construct profitable trading strategies. Their work suggests that rule-based systems grounded in candlestick features can achieve competitive returns compared to conventional technical indicators.

Table 2. 1. Overview of recent research on candlestick pattern recognition and trading models.

Ref	Market	Model	Performance Metric
[26]	Indonesian Mining Index (JKMING)	CNN	Directional accuracy (qualitative)
[23]	S&P 500 Index Future	Ensemble CNN with GAF encoding	Outperforms buy-and-hold; high classification accuracy
[27]	General stock market/investment tools	YOLO algorithm	Detection of “buy-sell” signals (qualitative)
[17]	S&P 500	CNN + Double Deep Q-Network (DDQN)	Returns above S&P 500 benchmark
[28]	General financial markets	Machine learning with separated pattern recognition	Reduction in false signals (qualitative)
[40]	Chinese stock market	SSA-CPBiGRU	Reduction in MAPE, RMSE; +2.05% improvement in R^2
[29]	General financial markets	CNN with Gramian Angular Field (GAF) encoding	Improved accuracy over LSTM
[25]	General financial markets	YOLO-based dynamic detection	Improved detection speed and accuracy
[41]	Stocks (20 tested)	Rule-based candlestick analysis	Profitability varies by pattern
[11]	General markets	N/A (contextual analysis)	Qualitative impact on trader sentiment
[7]	Intraday stock data	Wavelet transform + CNN	Binary classification accuracy

[42]	Stock market (K-line)	Fuzzy algorithms and clustering	Improved robustness (qualitative)
[31]	Stocks	Analysis of Engulfing/Harami patterns	Improved prediction under specific criteria
[18]	Taiwan Exchange	CNN autoencoder + RNN	Higher trend forecasting accuracy
[9]	Taiwan Exchange Weighted Stock Index	Deep Candlestick Predictor (CNN autoencoder + 1D CNN)	Outperforms traditional methods
[4]	General stock market	Custom CNN	High classification accuracy; faster training
[32]	Historical stock data	Feed-forward Neural Network with SMOTE	High overall accuracy; lower F1 score
[33]	Taiwanese & Indonesian stock markets	CNN, Random Forest, VGG networks	Sensitivity to image size and volume inclusion
[34]	U.S. stocks (global testing in 31 countries)	CNN + Deep Q-Network	0.1% to 1.0% return per transaction
[8]	General stocks	Sequential pattern mining + similarity measure	Correlation with future trends (qualitative)
[6]	Various indices/stock markets	Ensemble ML (RF, GBDT, LSTM, etc.)	Improved Sharpe ratio; reduced drawdown
[35]	General stock markets	Heikin-Ashi + Fuzzy time series + Cloud model	High forecasting accuracy
[22]	Intraday stock data	Deep Neural Network (with 10 technical indicators)	Improved prediction performance
[36]	High-frequency trading data	Logistic Regression + CNN on order-flow images	Short-term prediction accuracy
[14]	General stock market	Object detection + GAF encoding	Improved detection rate of overlapping patterns
[37]	Forex (EUR–USD)	Adaptive candlestick classification (statistical inference)	Net returns not positive after transaction costs
[21]	Crypto futures	Random Forest (iteratively retrained)	Candlestick patterns often outperform technical indicators
[38]	Chinese stock market	Convolutional autoencoder + K-means clustering	High Sharpe ratio; improved risk-adjusted returns

[39]	Short-term trading positions	Hybrid CNN-LSTM	Robust accuracy in short-term predictions
[16]	Global indices	Candlestick pattern recognition + One Hot Encoding + XGBoost	Outperforms buy-and-hold strategy
[2]	Stock trading strategies	Feature extraction + Chi-square Automatic Interaction Detector	Superior profitability relative to common technical indicators

PART 3

MATERIAL AND METHODS

3.1. TIME SERIES

A time series is an ordered sequence of observations recorded over time, often viewed as the realization of a stochastic process (a sequence of random variables indexed by time). Unlike independent observations, time-series data points are temporally correlated; what happens at time t depends on previous times.

In analyzing time series, statisticians examine properties such as stationarity and autocorrelation to characterize the data-generating process and inform suitable modeling and forecasting techniques.

3.1.1. Forcasting Methods

Time series forecasting aims to predict future values $X_{n+1}, X_{n+2}, \ldots$, given observed data $X_1, X_2, \ldots, X_n$. Forecasting methods can range from simple empirical techniques to complex stochastic models.

- **Autoregressive Integrated Moving Average (ARIMA) Models:** These are classic statistical models that combine autoregression (AR) and moving average (MA) components to capture different aspects of autocorrelation . The ARMA (p, q) model assumes that the time series can be modeled as a linear function of its previous p values and previous q random shocks (errors). In equation form, an ARMA model appears as follows:

$$X_{\mathrm{t}} = \delta + \phi_1 X_{\mathrm{t-1}} + \phi_2 X_{\mathrm{t-2}} + \ldots + \phi_{\mathrm{p}} X_{\mathrm{t-p}} + A_{\mathrm{t}} - \theta_1 A_{\mathrm{t-1}} - \ldots - \theta q A_{\mathrm{t-}} q \quad (3.1)$$

Where $\varphi_{(\mathrm{i})}$ are the autoregressive coefficients, $\theta_{(\mathrm{j})}$ are the moving average coefficients, δ is a constant term (drift), and A_{t} represents the white noise error term at time t. The AR part (the φ terms) captures persistence illustrating how past values influence the current observation while the MA part (the θ terms) reflects the effect of previous shock residuals on the current value. If the series is nonstationary, differencing can be applied d times to achieve stationarity, resulting in an ARIMA (p, d, q) model. ARIMA models (and their seasonal variants, SARIMA) are based on the Box-Jenkins methodology for model identification, estimation, and diagnostic checking. This approach leverages the ACF and PACF patterns to select suitable orders for p and q. stochastic foundation, ARIMA models provide not only point forecasts but also confidence intervals based on the underlying probability model [43].

- **Exponential Smoothing Methods:** In exponential smoothing, forecasts are constructed using exponentially weighted averages of past observations, giving more weight to recent data. Methods such as Simple Exponential Smoothing (for level forecasting), Holt’s linear method (which adds a trend component), and Holt-Winters (which add seasonal components) are effective for patterns with trends and seasonality. These methods are not explicitly stochastic models but can be interpreted in state-space form with certain error assumptions. They are valued for their simplicity and strong empirical performance, especially for short-term forecasts in business contexts [44].
- **Machine Learning and Neural Networks:** Modern approaches treat forecasting as a general prediction problem and use machine-learning algorithms to capture possible nonlinear relationships. Feedforward neural networks or recurrent neural networks (RNNs) can be trained on the sliding window inputs of past values to predict future values. More recent techniques include Long Short-Term Memory (LSTM) networks and transformers that are adept at learning long-range dependencies in sequential data. These models do not assume a specific stochastic structure a priori; instead, they learn patterns directly from data, which can be advantageous if the true data-generating process is highly complex or nonlinear. However, they typically require large amounts of data and careful regularization to avoid overfitting.

In practice, the model choice depends on the characteristics of the time series and the forecasting horizon. For instance, ARIMA models are often effective for the short-term forecasting of stationary or different series, leveraging well-understood statistical properties and diagnostic checks. On the other hand, when data show regime changes or nonlinear patterns that are difficult to capture with linear models, machine learning methods might offer improvements, albeit at the cost of interpretability. Ultimately, evaluating the forecast accuracy on held-out time periods (e.g., via rolling forecast origin evaluation) and checking residuals are crucial steps, regardless of the method used, to ensure that the model has captured the signal and not just overfit noise.

3.1.1.1. Time Series Forecasting vs. Regression Analysis

Although both time-series forecasting and regression analysis are used to predict future values, they fundamentally differ in their methodologies and objectives.

3.1.1.2. Time-Series Forecasting

Time series forecasting is an extrapolation technique. It relies on inherent temporal ordering and autocorrelation of past observations to predict future outcomes. In this approach, models such as ARIMA, exponential smoothing, or even advanced neural network architectures explicitly incorporate the time-dependent structure of the data. These models were designed to extend the observed sequence beyond its current range by capturing the trends, seasonality, and cyclical patterns inherent in the data [43].

3.1.1.3. Regression Analysis

In contrast, regression analysis is typically viewed as an interpolation method that aims to explain the relationships between two or more variables based on the observed data. A regression model quantifies how changes in one or more independent variables are associated with changes in dependent variables. While regression models can be used to predict future values, their primary purpose is to uncover and mathematically describe the historical relationships among variables, rather than relying on the sequential dependency found in time-series data [45].

3.2. THE SLIDING WINDOW MECHANISM: CONCEPT AND APPLICATION

The sliding-window mechanism is a fundamental technique for transforming a time series into a collection of fixed-length segments. This approach not only facilitates the use of supervised learning algorithms by converting sequential data into independent samples but also enables detailed analysis of localized patterns within the data [46].

3.2.1. Defining the Sliding Window

At its core, this method involves defining a “window” as a consecutive block of observations of fixed length, denoted by L. For a given timeseries X_t, each window captures a contiguous segment starting at time t and ending at time $t + L - 1$. Mathematically, this can be expressed as:

$$X(t) = [\, x_t, x_{t+1}, \dots, x_{t+L-1}], \tag{3.2}$$

where $t = 1,2, \dots, N - L + 1$ in a time series of length N.

3.2.2. Creating the Windows

The process was initiated at the beginning of the series. The first window is simply the sequence $X(1) = [x_1, x_2, \dots, x_L]$. Once the initial segment is defined, the window is systematically moved forward by a fixed step size—commonly referred to as the “stride.” When the stride is set to 1, the subsequent window is

$$X(2) = [x_2, x_3, \dots, x_{L+1}] \tag{3.3}$$

This sliding operation is repeated until the entire time series is segmented into overlapping or, if desired, into disjoint windows.

3.2.3. Overlap and Its Significance

A key characteristic of the sliding window method is its potential for overlapping consecutive windows. Adjacent windows share a substantial portion of their data, with a stride smaller than the window length. This overlapping is not a drawback; rather, it ensures that subtle transitions or transient patterns in the time series are comprehensively captured. The redundancy inherent in overlapping windows increases the robustness of pattern recognition and forecasting tasks, albeit at the cost of introducing dependence among samples, which must be carefully considered in subsequent analyses [47].

3.2.4. Application in Forecasting

One of the most common applications of the sliding window technique is time series forecasting. In this context, each window served as an input sample for the predictive model. The immediate next value in the series, denoted by x_{t+L}, is used as the target output. This strategy converts the problem into a supervised learning task where input-output pairs $(X(t), y(t))$ are formed, with:

$$y(t) = x_{t+L} \tag{3.4}$$

Thus, the model was trained to predict the next time point based on the historical data encapsulated within each window. For example, with daily observations and $L = 3$, a sample could be constructed as

$$[x_t \, , x_{t+1} \, , x_{t+2}] \rightarrow x_{t+3}. \tag{3.5}$$

3.2.5. Design Configurations: Stride, Window Size, and Beyond

When employing sliding windows, several configuration choices significantly impact the performance and efficiency:

- **Window Size (L)**: The choice of L determines the amount of historical data used for forecasting or pattern analysis. A small window may capture only short-term dynamics, potentially missing longer-term trends, whereas a large window may incorporate extraneous or non-stationary information. Often, the optimal window size is determined empirically or through domain expertise [48].
- **Stride (Step Size)**: Stride defines the extent to which the window moves at each step. A stride of 1 maximizes the number of training samples but also increases the overlap between windows. Larger strides can reduce redundancy and computational load but may sacrifice important details.
- **Overlapping versus Disjoint Windows**: While overlapping windows (stride <L) are commonly preferred for their comprehensive data representation, disjoint windows (stride =L) might be used in certain signal processing applications where independence between samples is crucial [46].
- **Expanding vs. Fixed Windows**: In forecasting, one might choose between a fixed (rolling) window that maintains a constant length L, or an expanding window that gradually incorporates all past observations. The former can adapt to local changes, whereas the latter benefits from a larger historical context, if the process is relatively stable.
- **Multivariate and multistep forecasting**: For multivariate time series, the window can include lagged values for several variables, forming a lagged feature matrix. In multistep forecasting, the window might be used to predict a sequence of future values, which requires either separate models for each forecast horizon or an integrated sequence-to-sequence approach [49].

3.2.6. Computational Considerations

In practical implementations, efficiency is paramount, particularly for large datasets. Instead of physically copying overlapping data for each window, algorithms often employ rolling computations that incrementally update statistics (such as moving averages or variances) incrementally. This approach minimizes computational overhead and memory usage, enabling real-time or large-scale applications.

3.3. PATTERN IDENTIFICATION USING TA-LIB

A comprehensive summary of the candlestick pattern functions provided by TA-Lib is provided below. While TA-Lib defines 55 functions, many of these functions return distinct signals depending on whether a bullish or bearish variant is detected, yielding 61 distinct pattern signals. Each pattern is based on specific mathematical relationships between the Open, High, Low, and Close (OHLC) values for one or more periods. The following list summarizes each pattern by its TA-Lib function name and provides a brief description of its conceptual criteria [50].

For any candle with index i (and previous candles as $i-1, i-2, etc.$), define:

- O_i = Open price
- H_i = High price
- L_i = Low price
- C_i = Close price
- Real Body

$$(RB_i) = |C_i - O_i| \tag{3.6}$$

- Upper Shadow

$$(US_i) = H_i - max(O_i, C_i) \tag{3.7}$$

- Lower Shadow

$$(LS_i) = min(O_i, C_i) - L_i \tag{3.8}$$

Tolerances:

- δ (delta) is a small tolerance for Doji conditions (i.e., $|C_i - O_i| \leq \delta$).
- ε (epsilon) is a small value used for “near” comparisons.

- Terms such as Avg Short Body and Avg Long Shadow represent adaptive averages computed over recent candles.
- TA-Lib functions typically return +100 for bullish signals and –100 for bearish signals.

CDL2CROWS

Concept: Two consecutive bearish candles signal a potential reversal in an uptrend.

Equations:

$$C_{(\mathrm{i}-1)} < O_{(\mathrm{i}-1)}\; and\; C_{\mathrm{i}} < O_{\mathrm{i}} \tag{3.9}$$

$$O_{\mathrm{i}} < C_{(\mathrm{i}-1)}\; and\; RB_{\mathrm{i}} > RB_{(\mathrm{i}-1)} \tag{3.10}$$

CDL3BLACKCROWS

Concept: Three consecutive long-bearish candles with progressively lower closes, indicating strong seller control.

Equations:

for $k = i - 2, i - 1, i$

For each candle:

$$C_{\mathrm{k}} < O_{\mathrm{k}}, O_{(\mathrm{i}-1)} \geq C_{(\mathrm{i}-2)}\; and\; O_{\mathrm{i}} \geq C_{(\mathrm{i}-1)}, with\; C_{(\mathrm{i}-2)} > C_{(\mathrm{i}-1)} > C_{\mathrm{i}} \tag{3.11}$$

CDL3INSIDE

Concept: A small candle whose body is entirely contained within the body of the previous candle; the bearish variant indicates a pause in an uptrend, whereas the bullish version (colors reversed) indicates a reversal from a downtrend.

Equations:

$$min(O_{\mathrm{i}}, C_{\mathrm{i}}) \; > \; min(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)}) \; and \; \; max(O_{\mathrm{i}}, C_{\mathrm{i}}) \; < \; max(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)}) \quad (3.12)$$

CDL3LINESTRIKE

Concept: Three-candle formation, where the final candle (with reversal) closes near or beyond the opening of the first candle.

Equations:

$$C_{\mathrm{i}} \; \leq \; O_{(\mathrm{i}-3)} \; and \; \; RB_{\mathrm{i}} \; is \; long \; relative \; to \; the \; intermediate \; candles. \qquad (3.13)$$

CDL3OUTSIDE

Concept: A pattern in which a candle’s body is wider than that of the previous candle, with the final candle reversing the preceding trend.

Equations:

$$O_{\mathrm{i}} \; \leq \; min(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)}) \; and \; C_{\mathrm{i}} \; \geq \; max(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)}) \qquad (3.14)$$

CDL3STARSINSOUTH

Concept: A three-candle pattern at the bottom of a downwards trend, where the final candle is bullish.

Equations:

let Candle i– 2 $and \; i \; -$ 1 be small indecisive) and Candle i is bullish

$$C_{(\mathrm{i}-2)} \; < \; O_{(\mathrm{i}-2)} \;\; ,|\, RB_{(\mathrm{i}-1)} \,| \leq \; \delta, \; C_{\mathrm{i}} \; > \; O_{\mathrm{i}} \qquad (3.15)$$

CDL3WHITESOLDIERS

Concept: Three consecutive long bullish candles with progressively higher closes indicating strong upward momentum.

Equations:

for $k\ =\ i - 2, i - 1, i$:

$C_{\mathrm{k}}\ >\ C_{k+1}\ , C_{\mathrm{k}} - \varepsilon\ , and\ C_{i-2} < C_{i-1} < C_i$ (3.16)

CDLABANDONEDBABY

Concept: A three-candle pattern with a doji isolated by gaps from its neighbors signaling a sharp reversal.

Equations:

$$\left|C^{(}{}_{\mathrm{i}}{}^{-1)} - O^{(}{}_{\mathrm{i}}{}^{-1)}\right| \leq\ \delta, and\ \ gaps\ exists\ that\ :$$

$O_{(\mathrm{i-1})}\ >\ H_{(\mathrm{i-2})}\ +\ \varepsilon\ , O_{(\mathrm{i-1})}\ <\ L_{\mathrm{i}} - \varepsilon$ (3.17)

CDLADVANCEBLOCK

Concept: In an uptrend, a series of bullish candles with diminishing strength, suggesting that the uptrend may soon reverse.

Equations:

$RB_{(\mathrm{i-2})}\ >\ RB_{(\mathrm{i-1})}\ >\ RB_{\mathrm{i}}$ (bodies become progressively smaller) (3.18)

CDLBELTHOLD

Concept: A long bullish candle with little or no shadow, indicating strong buyer control from open to close.

Equation (Bullish):

$$O_{\mathrm{i}} \approx L_{\mathrm{i}}, C_{\mathrm{i}} \approx H_{\mathrm{i}}, with\ US_{\mathrm{i}} \approx 0\ and\ LS_{\mathrm{i}} \approx 0 \tag{3.19}$$

CDLBREAKAWAY

Concept: A pattern in which the current candle “breaks away” from the previous trend’s range, signaling a potential reversal.

quations:

$$Candle\ i\ satisfies\ \ C_{\mathrm{i}} > H_{(\mathrm{i}-1)}\ (bullish)\ or\ C_{\mathrm{i}} < L_{(\mathrm{i}-1)}\ (bearish) \tag{3.20}$$

CDLCLOSINGMARUBOZU

Concept: A candle with virtually no shadows closing at the extreme of its range reinforces the current trend.

Equation (Bullish):

$$O_{\mathrm{i}} \approx L_{\mathrm{i}}\ and\ C_{\mathrm{i}} \approx H_{\mathrm{i}} \tag{3.21}$$

CDLCONCEALBABYSWALL

Concept: A two-candle pattern in which a small candle is completely contained within the body of a preceding large candle, indicating potential reversal. Equations:

$$max(O_{\mathrm{i}}, C_{\mathrm{i}}) < max(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)})\ and\ \ min(O_{\mathrm{i}}, C_{\mathrm{i}}) > min(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)}) \tag{3.22}$$

CDLCOUNTERATTACK

Concept: Two consecutive candles with nearly identical closes follow a trend, suggesting a potential reversal.

Equations:

$$|C_{\mathrm{i}} - C_{(\mathrm{i}-1)}| \leq \varepsilon \; and \; after \; a \; sustained \; trend \tag{3.23}$$

CDLDARKCLOUDCOVER

Concept: A bearish reversal pattern in which a bullish candle is followed by a bearish candle that opens above the previous height but closes below the midpoint of the candle's body.

Equations:

$$i\text{–}1\text{:}\; C_{(\mathrm{i}-1)} > O_{(\mathrm{i}-1)}\,, O_{\mathrm{i}} > H_{(\mathrm{i}-1)}\,, C_{\mathrm{i}} < \frac{(O_{(\mathrm{i}-1)} + C_{(\mathrm{i}-1)})}{2} \tag{3.24}$$

CDLDOJI

Concept: A candle where the open and close are nearly identical, reflecting market indecision.

Equations:

$$|C_{\mathrm{i}} - O_{\mathrm{i}}| \leq \delta \tag{3.25}$$

CDLDOJISTAR

Concept: A doji appears with a gap relative to the previous candle, suggesting a potential reversal when found after a trend.

Equations:

$$|C_{\mathrm{i}} - O_{\mathrm{i}}| \leq \delta \; and \; a \; gap \; exists \; (O_{\mathrm{i}} \; is \; not \; approximately \; equal \; to \; O_{(\mathrm{i}-1)}) \quad (3.26)$$

CDLDRAGONFLYDOJI

Concept: A doji with a long lower shadow and no upper shadow indicates that sellers drove prices down, but buyers typically regained control and bullish in a downtrend.

Equations:

$$|C_{\mathrm{i}} - O_{\mathrm{i}}| \leq \delta, US_{\mathrm{i}} \approx 0, \quad LS_{\mathrm{i}} \gg RB_{\mathrm{i}} \quad (3.27)$$

CDLENGULFING

Concept: A two-candle pattern where the second candle's body completely engulfs the previous candle's body; bullish signals reverse upward, and bearish signals reverse downwards.

Equation (Bullish):

$$C_{(\mathrm{i}-1)} < O_{(\mathrm{i}-1)}, C_{\mathrm{i}} > O_{\mathrm{i}}, O_{\mathrm{i}} \leq C_{(\mathrm{i}-1)} \; and \; C_{\mathrm{i}} \geq O_{(\mathrm{i}-1)} \quad (3.28)$$

CDLEVENINGDOJISTAR

Concept: A variant of Evening Star, where the middle candle is a doji, signals a bearish reversal in an uptrend.

Equations:

$$Candle \; i-2: C_{(\mathrm{i}-2)} > O_{(\mathrm{i}-2)} Candle \; i-1: |C_{(\mathrm{i}-1)} - O_{(\mathrm{i}-1)}| \leq \delta$$
$$Candle \; i: C_{\mathrm{i}} < O_{\mathrm{i}} \; and \; C_{\mathrm{i}} \leq \frac{(O_{(\mathrm{i}-2)} + C_{(\mathrm{i}-2)})}{2} \quad (3.29)$$

CDLEVENINGSTAR

Concept: A classic three-candle-bearish reversal pattern: a long bullish candle is followed by a small candle (star), followed by a long bearish candle that closes well into the first candle's body.

Equations:

$$Candle\ i\text{–}\ 2\text{: } C_{(\mathrm{i}-2)}\ >\ O_{(\mathrm{i}-2)}Candle\ i\text{–}\ 1\text{: } RB_{(\mathrm{i}-1)}\ is\ small$$

$$Candle\ i\text{: } C_{\mathrm{i}}\ <\ O_{\mathrm{i}}\ and\ C_{\mathrm{i}}\ \leq\ \frac{(O_{(\mathrm{i}-2)}+C_{(\mathrm{i}-2)})}{2} \qquad (3.30)$$

CDLGAPSIDESIDEWHITE

Concept: A bullish gap pattern in which the second candle opens above the first candle, and both have similar closing levels. The bearish variant is the inverse.

Equation (Bullish):

$$O_{\mathrm{i}}\ >\ C_{(\mathrm{i}-1)}\ +\ \varepsilon\ and\ |C_{\mathrm{i}} - C_{(\mathrm{i}-1)}|\ \leq\ \varepsilon \qquad (3.31)$$

CDLGRAVESTONEDOJI

Concept: A doji with a long upper shadow and no lower shadow, indicating the rejection of higher prices and potential bearish reversal.

Equations:

$$|C_{\mathrm{i}} - O_{\mathrm{i}}|\ \leq\ \delta, US_{\mathrm{i}}\ \gg RB_{\mathrm{i}}, and\ LS_{\mathrm{i}}\ \approx 0 \qquad (3.32)$$

CDLHAMMER

Concept: A single bullish reversal candle with a small body near the top, long lower shadow, and little or no upper shadow.

Equations:

$$RB_{\mathrm{i}} < AvgShortBody, US_{\mathrm{i}} < AvgVeryShortShadow$$
$$LS_{\mathrm{i}} > AvgLongShadow, and\ min(O_{\mathrm{i}}, C_{\mathrm{i}}) \leq L_{(\mathrm{i}-1)} + \varepsilon \qquad (3.33)$$

CDLHANGINGMAN

Concept: Similar in shape to the Hammer but appearing in an uptrend, suggesting a bearish reversal.

Equations:

$$RB_{\mathrm{i}} < AvgShortBody, US_{\mathrm{i}} < AvgVeryShortShadow$$
$$LS_{\mathrm{i}} > AvgLongShadow, and\ min(O_{\mathrm{i}}, C_{\mathrm{i}}) \geq H_{(\mathrm{i}-1)} - \varepsilon \qquad (3.34)$$

CDLHARAMI

Concept: A two-candle pattern in which a large candle is followed by a small candle entirely contained within its body.

Equation (Bullish):

$$C_{(\mathrm{i}-1)} < O_{(\mathrm{i}-1)}, \qquad C_{\mathrm{i}} > O_{\mathrm{i}}$$
$$And: max(O_{\mathrm{i}}, C_{\mathrm{i}}) < max(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)})\ and\ min(O_{\mathrm{i}}, C_{\mathrm{i}}) > min(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)}) \qquad (3.35)$$

CDLHARAMICROSS

Concept: A variation of the Harami pattern, in which the second candle is a doji, emphasizing indecision.

Equations:

$$|C_{\mathrm{i}} - O_{\mathrm{i}}| \leq \delta, with\ Harami\ containment\ as\ CDLHARAMI \qquad (3.36)$$

CDLHIGHWAVE

Concept: A candle with extremely long shadows on both sides and a very small body suggests market indecision that may precede a reversal.

Equations:

$$RB_{\mathrm{i}} \ll US_{\mathrm{i}}\ and\ LS_{\mathrm{i}} \tag{3.37}$$

CDLHIKKAKE

Concept: A pattern that identifies false breakouts. If the candle movement contradicts the expected breakout, it signals a reversal.

Equations:

$$If\ |RB_{\mathrm{i}}| < AvgShortBody\ and\ a\ gap\ occurs\ that\ is\ later\ invalidated, the\ Hikkake\ condition\ is\ met. \tag{3.38}$$

CDLHIKKAKE2

Concept: A refined version of the Hikkake pattern with adjusted thresholds for breakout and reversal.

Equations:

$$Similar\ to\ CDLHIKKAKE\ Pattern, but\ with\ |RB_{\mathrm{i}}| < k \times AvgShortBody\ (k < 1)\ and\ adjusted\ gap\ conditions. \tag{3.39}$$

CDLHOMINGPIGEON

Concept: A two-candle pattern in which the second candle's body is completely contained within the first candle's body suggests potential bullish reversal.

Equations:

$$min(O_i, C_i) > min(O_{(i-1)}, C_{(i-1)})\ and\ max(O_i, C_i) < max(O_{(i-1)}, C_{(i-1)}) \tag{3.40}$$

CDLIDENTICAL3CROWS

Concept: A variant of three black crows where three bearish candles have nearly identical shapes and sizes, reinforcing the bearish signal.

Equations:

$$RB_{(i-2)} \approx RB_{(i-1)} \approx RB_i, with\ C_{(i-2)} > C_{(i-1)} > C_i \tag{3.41}$$

CDLINNECK

Concept: A two-candle pattern in which the second candle opens near the previous candle's close and closes near its open, suggesting a reversal.

Equations:

$$|O_i - C_{(i-1)}| \leq \varepsilon\ and\ |C_i - O_{(i-1)}| \leq \varepsilon \tag{3.42}$$

CDLONNECK

Concept: Similar to the In-Neck pattern, the second candle's body does not exceed the previous candle's body.

Equations:

$$min(O_{\mathrm{i}}, C_{\mathrm{i}}) \approx min(O_{(\mathrm{i}-1)}, C_{(\mathrm{i}-1)})\ (within\ \varepsilon) \tag{3.43}$$

CDLKICKING

Concept: Two-candle reversal pattern featuring a gap. In the bullish variant, a bearish candle is followed by a bullish candle with gaps in opposite directions.

Equations (Bullish variants)

$$O_{\mathrm{i}} < C_{(\mathrm{i}-1)}\ and\ C_{\mathrm{i}} > O_{(\mathrm{i}-1)} + \Delta, \tag{3.44}$$

where Δ is the threshold gap.

CDLLADDERBOTTOM

Concept: A bullish reversal pattern in which successive candles form progressively lower lows with shrinking bodies, suggesting a weakening of selling pressure.

Equations:

$$L_{(\mathrm{i}-2)} > L_{(\mathrm{i}-1)} > L_{\mathrm{i}}\ and\ RB_{(\mathrm{i}-1)} < RB_{(\mathrm{i}-2)}, RB_{\mathrm{i}} < RB_{(\mathrm{i}-1)} \tag{3.45}$$

CDLLONGLEGGEDDOJI

Concept: A doji with exceptionally long shadows on both sides, representing extreme indecision.

Equations:

$$|C_{\mathrm{i}} - O_{\mathrm{i}}| \leq \delta, US_{\mathrm{i}} >> RB_{\mathrm{i}}, and\ LS_{\mathrm{i}} >> RB_{\mathrm{i}} \tag{3.46}$$

CDLLONGLINECANDLE

Concept: A candle with a very long body and minimal shadows. In the bullish variant, strong upward movement was observed.

Equations (Bullish variants)

$$RB_{\mathrm{i}} >> AvgBody, US_{\mathrm{i}} \approx 0, LS_{\mathrm{i}} \approx 0, and\ C_{\mathrm{i}} > O_{\mathrm{i}} \tag{3.47}$$

CDLMARUBOZU

Concept: A candle with no shadows; a bullish marubozu is open at the low and close at the high, showing complete control by buyers.

Equation (Bullish):

$$O_{\mathrm{i}} \approx L_{\mathrm{i}}\ and\ C_{\mathrm{i}} \approx H_{\mathrm{i}} \tag{3.48}$$

CDLMATCHINGLOW

Concept: Two consecutive candles with nearly identical lows indicate a firm support level that may lead to bullish reversal.

Equations:

$$|L_{\mathrm{i}} - L_{(\mathrm{i}-1)}| \leq \varepsilon \tag{3.49}$$

CDLMATHOLD

Concept: A three-candle pattern in which a long bullish candle is followed by a consolidation phase (small bodies), then another bullish candle that resumes the trend; failure can signal a bearish reversal.

Equations (Bullish variants)

$$Candle\ i\!-2\!: C_{(\mathrm{i}-2)}\ >\ O_{(\mathrm{i}-2)}\ , Candles\ i\!-1\!: RB^{(_{\mathrm{i}}{}^{-1})} \leq\ \delta\ ,$$
$$Candle\ i\!: C_{\mathrm{i}}\ >\ O_{\mathrm{i}}\ and\ C_{\mathrm{i}}\ >\ \frac{(O_{(\mathrm{i}-2)} + C_{(\mathrm{i}-2)})}{2} \qquad (3.50)$$

CDLMORNINGDOJISTAR

Concept: A variant of the morning star pattern with a doji as the middle candle, indicating indecision before a strong bullish reversal.

Equations:

$$Candle\ i\!-2\!: C_{(\mathrm{i}-2)}\ <\ O_{(\mathrm{i}-2)}\ , Candle\ i\!-1\!: |C_{(\mathrm{i}-1)} - O_{(\mathrm{i}-1)}|\ \leq\ \delta\ ,$$
$$Candle\ i\!: C_{\mathrm{i}}\ >\ O_{\mathrm{i}}\ and\ C_{\mathrm{i}}\ \geq\ \frac{(O_{(\mathrm{i}-2)} + C_{(\mathrm{i}-2)})}{2} \qquad (3.51)$$

CDLMORNINGSTAR

Concept: A classic three-candle bullish reversal pattern, in which a long bearish candle is followed by a small candle and a long bullish candle that closes well into the first candle’s body.

Equations:

$$Candle\ i\!-2\!: C_{(\mathrm{i}-2)}\ <\ O_{(\mathrm{i}-2)} Candle\ i\!-1\!: RB_{(\mathrm{i}-1)}\ is\ small$$
$$Candle\ i\!: C_{\mathrm{i}}\ >\ O_{\mathrm{i}}\ and\ C_{\mathrm{i}}\ \geq\ \frac{(O_{(\mathrm{i}-2)} + C_{(\mathrm{i}-2)})}{2} \qquad (3.52)$$

CDLPIERCING

Concept: A bullish two-candle reversal pattern in which a bearish candle is followed by a bullish candle that opens below the previous low and closes above the midpoint of the body of the previous candle.

Equations:

$$C_{(\mathrm{i}-1)} \ < \ O_{(\mathrm{i}-1)}$$

$$O_{\mathrm{i}} \ < \ L_{(\mathrm{i}-1)} \ and \ C_{\mathrm{i}} \ > \ \frac{(O_{(\mathrm{i}-1)} + C_{(\mathrm{i}-1)})}{2} \tag{3.53}$$

CDLRICKSHAWMAN

Concept: A pattern with a long lower shadow and a very small body, interpreted as a bullish reversal when appearing after a downtrend.

Equations:

$$RB_{\mathrm{i}} \ < \ \delta, LS_{\mathrm{i}} \ >> \ RB_{\mathrm{i}}, and \ US_{\mathrm{i}} \ \approx \ 0 \tag{3.54}$$

CDLRISE3METHODS

Concept: A bullish continuation pattern: a long bullish candle is followed by three small candles within its range, and another long bullish candle resumes the trend.

Equations:

$$Candle \ i\text{–}\,3\text{: } C_{(\mathrm{i}-3)} \ > \ O_{(\mathrm{i}-3)}\,, Candles \ i\text{–}\,2 \ and \ i\text{–}\,1\text{: } max(O_{\mathrm{k}}, C_{\mathrm{k}})$$
$$\leq \ H_{(\mathrm{i}-3)} \ and \ min(O_{\mathrm{k}}, C_{\mathrm{k}}) \ \geq \ L_{(\mathrm{i}-3)}\,,$$
$$Candle \ i\text{: } C_{\mathrm{i}} \ > \ O_{\mathrm{i}} \ and \ C_{\mathrm{i}} \ > \ C_{(\mathrm{i}-3)} \tag{3.55}$$

CDLTakuri

Concept: Also known as the Dragonfly Doji with a long lower shadow; when it appears after a downtrend, it suggests a bullish reversal.

Equations:

$$|C_{\mathrm{i}} - O_{\mathrm{i}}| \ \leq \ \delta, US_{\mathrm{i}} \ \approx \ 0, and \ LS_{\mathrm{i}} \ >> \ RB_{\mathrm{i}} \tag{3.56}$$

CDLTASUKIGAP

Concept: Gap pattern. In its bearish variant, it confirms a gap down with subsequent bearish actions.

Equations (Bearish variants)

$$O_{\mathrm{i}} > H_{(\mathrm{i-1})} + \varepsilon \textit{ and } C_{\mathrm{i}} < \frac{(O_{(\mathrm{i-1})} + C_{(\mathrm{i-1})})}{2} \quad (3.57)$$

CDLTHRUSTING

Concept: A bearish reversal pattern in which a bullish candle is followed by a bearish candle that closes significantly lower than the body of the previous candle.

Equations:

$$C_{(\mathrm{i-1})} > O_{(\mathrm{i-1})}, O_{\mathrm{i}} > H_{(\mathrm{i-1})},$$

$$C_{\mathrm{i}} < \frac{(O_{(\mathrm{i-1})} + C_{(\mathrm{i-1})})}{2} \quad (3.58)$$

CDLTRISTAR

Concept: A three-candle pattern with a doji in the middle, signaling indecision that may precede a bearish move.

Equations:

$$|C_{(\mathrm{i-1})} - O_{(\mathrm{i-1})}| \leq \delta, with\ Candle\ i - 2\ bullish\ and\ Candle\ i\ bearish \quad (3.59)$$

CDLUNIQUE3RIVER

Concept: A complex three-candle pattern indicating decisive downwards movement after consolidation.

Equations:

The conditions involve relative body sizes and gap conditions, such that the final candle closes significantly lower than the first.

CDLUPSIDEGAP2CROWS

Concept: A gap followed by two bearish candles that confirm a downtrend after an up move.

Equations:

$$O_{\mathrm{i}} > C_{(\mathrm{i}-1)} + \varepsilon, and\ C_{\mathrm{i}} < O_{\mathrm{i}}\ with\ significant\ gap\ conditions. \tag{3.60}$$

CDLUPSIDEGAP3METHODS

Concept: A bearish continuation pattern with a gap followed by three candles that reinforce the downtrend.

Equations:

$$\begin{gathered} Candle\ i\text{–}\,2: Long\ bullishCandles\ i\text{–}\,1\ and\ i: Small\ and\ within\ the\ gap\ range \\ Final\ candle: Bearish, with\ C_{(\mathrm{i}+1)} < O_{(\mathrm{i}+1)} \end{gathered} \tag{3.61}$$

CDLPIERCING (Bearish interpretation)

Concept: Although it is typically bullish, if the conditions are reversed, it can be interpreted as bearish.

Equations:

$$If\ O_{\mathrm{i}} > H_{(\mathrm{i}-1)}\ and\ C_{\mathrm{i}} < \frac{(O_{(\mathrm{i}-1)} + C_{(\mathrm{i}-1)})}{2}, then\ it\ may\ signal\ bearish\ pressure. \tag{3.62}$$

CDLRICKSHAWMAN (Bearish interpretation)

Concept: When appearing in an uptrend, this variant of the rickshaw pattern can signal a bearish reversal.

Equations:

$$RB_{\mathrm{i}} < \delta, LS_{\mathrm{i}} >> RB_{\mathrm{i}}, US_{\mathrm{i}} \approx 0\ (in\ an\ uptrend\ context) \qquad (3.63)$$

CDLRISE3METHODS (Bearish Variant – Falling Three Methods)

Concept: The bearish counterpart to Rise Three Methods: A long bearish candle is followed by three small candles, and then another bearish candle confirming the trend.

Equations:

$$Candle\ i\text{–}\,3{:}\ C_{(\mathrm{i-3})} < O_{(\mathrm{i-3})}, Candles\ i\text{–}\,2\ and\ i\text{–}\,1{:}\ max(O_{\mathrm{k}}, C_{\mathrm{k}}) \leq H_{(\mathrm{i-3})}\ and\ min(O_{\mathrm{k}}, C_{\mathrm{k}}) \geq L_{(\mathrm{i-3})},$$

$$Candle\ i{:}\ C_{\mathrm{i}} < O_{\mathrm{i}}\ and\ C_{\mathrm{i}} < C_{(\mathrm{i-3})} \qquad (3.64)$$

CDLTakuri (Bearish interpretation)

Concept: Although typically bullish, if the Takuri pattern appears in an uptrend, it may be interpreted as bearish.

Equations:

$$|C_{\mathrm{i}} - O_{\mathrm{i}}| \leq \delta, US_{\mathrm{i}} \approx 0, LS_{\mathrm{i}} >> RB_{\mathrm{i}}, with\ context\ indicating\ reversal. \qquad (3.65)$$

CDLTASUKIGAP (Bearish variant)

Concept: A gap pattern that, in a bearish context, confirms a gap reduction with subsequent bearish actions.

Equations:

$$O_{\mathrm{i}} > H_{(\mathrm{i}-1)} + \varepsilon \; and \; C_{\mathrm{i}} < \frac{(O_{(\mathrm{i}-1)} + C_{(\mathrm{i}-1)})}{2} \quad (3.66)$$

CDLTHRUSTING

Concept: A bearish reversal pattern in which a bullish candle is followed by a bearish candle that closes significantly lower than the body of the previous candle.

Equations:

$$C_{(\mathrm{i}-1)} > O_{(\mathrm{i}-1)}$$

$$O_{\mathrm{i}} > H_{(\mathrm{i}-1)} \; and \; C_{\mathrm{i}} < \frac{(O_{(\mathrm{i}-1)} + C_{(\mathrm{i}-1)})}{2} \quad (3.67)$$

CDLTRISTAR (Bearish)

Concept: A three-candle pattern with the middle candle as a doji, signaling indecision before a bearish move.

Equations:

$$|C_{i-1} - O_{i-1}| \leq \delta, with \; C_{i-2} > O_{i-2} \; and \; C_i < O_i \quad (3.68)$$

CDLUNIQUE3RIVER (Bearish variant)

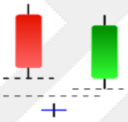
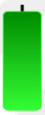
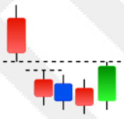
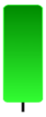
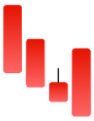
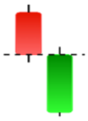
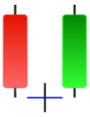
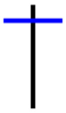
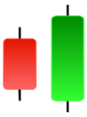
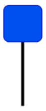
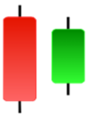
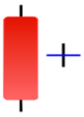
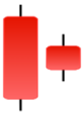
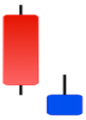
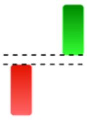
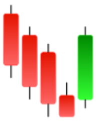
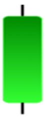
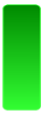
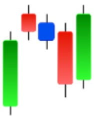
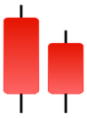
Concept: A complex three-candle pattern indicating decisive downwards movement after consolidation.

Equations:

$$\textit{Conditions involve relative body sizes and gaps across three candles,}$$
$$\textit{such that the final candle closes significantly lower than the first.}$$

The following table provides a visual representation of these formations, after presenting the definitions and equations that underpin each candlestick pattern. For instance, Table 3.1. presents a collection of Japanese Candlestick patterns that signal a bullish trend, while Table 3.2. shows cases that indicate a bearish market [51].

Table 3.1. Bullish candlestick patterns: structural examples and components of candle types.

Abandoned Baby	**Belt-hold**	**Breakaway**	**Closing Marubozu**
			
Concealing Baby Swallow	**Counterattack**	**Doji Star**	**Dragonfly Doji**
			
Engulfing	**Hammer**	**Harami**	**Harami Cross**
			
Homing Pigeon	**Inverted Hammer**	**Kicking**	**Ladder Bottom**
			
Long Line Candle	**Marubozu**	**Mat Hold**	**Matching Low**
			
Morning Doji Star	**Morning Star**	**Piercing Line**	**Rising Three Methods**

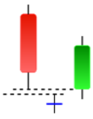
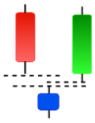
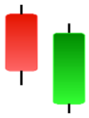
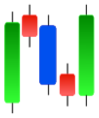
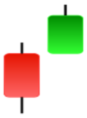
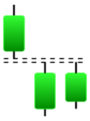
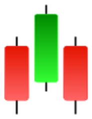
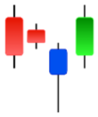
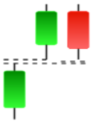
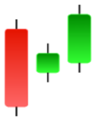
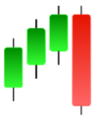
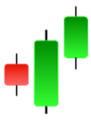
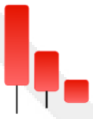
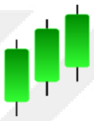
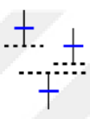
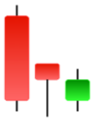
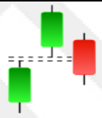
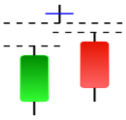
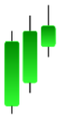
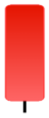
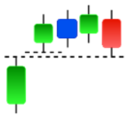
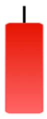
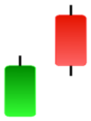
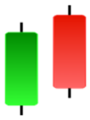
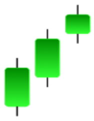
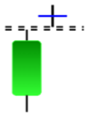
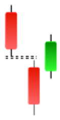
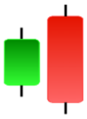
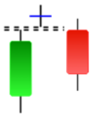
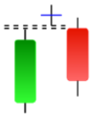
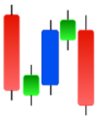
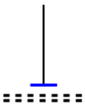
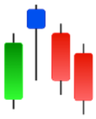
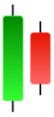
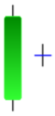
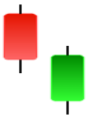
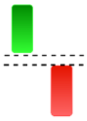
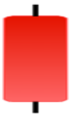
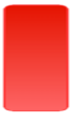
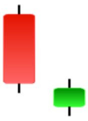
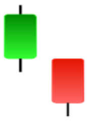
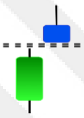
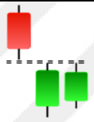
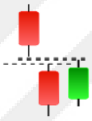
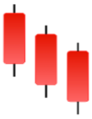
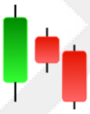
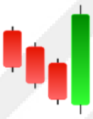
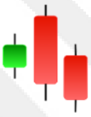
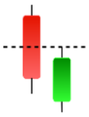
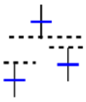
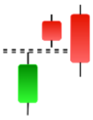
			
Separating Lines	**Side by Side White Lines**	**Stick Sandwich**	**Takuri**
			
Tasuki Gap	**Three Inside Up**	**Three Line Strike**	**Three Outside Up**
			
Three Stars in The South	**Three White Soldiers**	**Tri-Star**	**Unique Three-River**
			
Upside Gap Three Methods			
			

Table 3.2. Bearish candlestick patterns: structural examples and components of candle types.

Abandoned Baby	**Advance Block**	**Belt-hold**	**Breakaway**
			
Closing Marubozu	**Counterattack**	**Dark Cloud Cover**	**Deliberation**
			
Doji Star	**Downside Gap Three Methods**	**Engulfing**	**Evening Doji Star**
			
Evening Star	**Falling Three Methods**	**Gravestone Doji**	**Hanging Man**

			
Harami	**Harami Cross**	**In Neck**	**Kicking**
			
Long Line Candle	**Marubozu**	**On Neck**	**Separating Lines**
			
Shooting Star	**Side by Side White Lines**	**Tasuki Gap**	**Three Black Crows**
			
Three Inside Down	**Three Line Strike**	**Three Outside Down**	**Thrusting**
			
Tri-Star		**Upside Gap Two Crows**	
			

3.4. CNN MODEL ARCHITECTURE

CNNs have evolved significantly over the past few decades, driven by advancements in theory, computational power, and data availability. The earliest precursor to modern CNNs was proposed by **Kunihiko Fukushima** in the early 1980s, known as **Neocognitron**, which is a framework designed to recognize visual patterns through hierarchical layers of feature extraction. However, it was not until the work of **Yann LeCun** in the late 1980s and the early 1990s that CNNs gained traction, particularly with the development of **LeNet** for digit recognition using the MNIST dataset. LeNet demonstrated the effectiveness of local receptive fields and shared weights in reducing the computational costs of image processing.

The field witnessed a major breakthrough in 2012 when **AlexNet**, created by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, achieved a remarkable reduction in the error rate in the **ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)**. Several key innovations have underpinned AlexNet's success.

- **GPU Acceleration**: Leveraging parallel computation significantly sped up training.
- **ReLU Activation**: Mitigated vanishing gradients enabling the training of deeper networks.
- **Dropout Regularization**: Helped prevent overfitting by randomly disabling neuron connections.

Subsequent architectures, such as **VGG (2014)**, **GoogLeNet/Inception (2014–2015)**, and **ResNet (2015)**, further pushed the boundaries by increasing the network depth, introducing novel layer designs, and optimizing the parameter efficiency. These innovations have cemented CNNs as a cornerstone of deep learning, powering breakthroughs in image classification, object detection, and semantic segmentation.

3.4.1. Biological Inspiration: Similarities to the Human Brain

At the conceptual level, CNNs are inspired by the organization of neurons in the human brain, particularly in the visual cortex. In the early visual processing layers of the brain, neurons respond primarily to **local regions** of the visual field, similar to how **convolutional layers** in a CNN operate on localized patches of an image.

These neurons exhibit a **receptive field** and detect edges, lines, or simple patterns within their small regions. As signals propagate deeper into the visual cortex, neurons begin to respond to increasingly complex features (e.g., shapes or faces) constructed by combining simpler responses. This mirrors hierarchical feature extraction in CNNs, in which deeper layers integrate low-level features into higher-level concepts.

Furthermore, the neurons in the CNN were loosely analogous to the biological neurons as shown in figure 3.1.

- **Weighted Inputs**: Each CNN neuron computes a weighted sum of inputs similar to the synaptic connections in biological neurons.
- **Activation Function**: The nonlinear activation in a CNN neuron acts like a neuron firing mechanism when the summed input exceeds a certain threshold, it "fires."
- **Learning Through Adjustment of Weights**: Synaptic plasticity in the brain (where synapse strengths change over time) is loosely mirrored by backpropagation in CNNs, which updates the filter weights to minimize the loss function.

Although CNNs do not capture the full complexity of biological neural processes, the core principles of local receptive fields, hierarchical feature learning, and trainable connections reflect a simplified model of how the human visual cortex processes the visual information.

Convolutional Neural Networks (CNNs) are a powerful class of deep learning models designed to process **grid-structured data** such as images and time-series signals. Unlike fully connected neural networks, CNNs incorporate two key principles.

- **Local Connectivity** via convolutional layers
- **Spatial Pooling**

These design choices enable an efficient feature extraction and translational invariance.

A CNN transforms an input tensor (an image represented by height, width, and channels) through a sequence of layers, each with a specific function, culminating in an output (class probability). The lower layers detected simple patterns, whereas the deeper layers learned more complex abstract features.

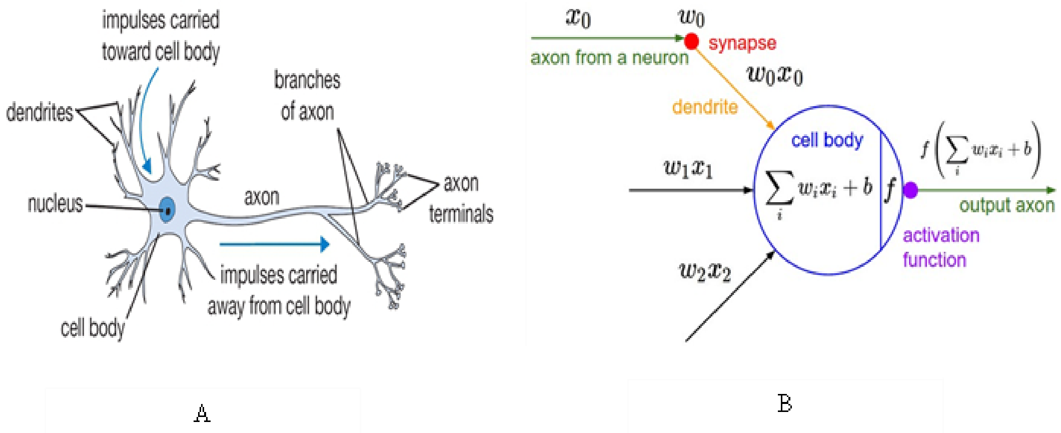


Figure 3.1. (A) Diagram of a biological neuron. (B) a perceptron.

3.4.2. CNN Layers:

3.4.2.1. Input Layer

The **input layer** serves as the interface between the raw input data and CNN. For images, the input is typically a three-dimensional array of shapes **(H, W, C)**, where

- **H** represents the height (number of pixel rows)
- **W** represents the width (number of pixel columns)
- **C** represents the number of channels (**C = 3** for RGB images)

For univariate time-series data, the input can be a two-dimensional array of shapes **(L, C)**, where

- **L** is the sequence length
- **C = 1** if dealing with a single variable, or higher if multiple features exist

The input layer does not perform computations but prepares the structured data tensor for the subsequent layers. Preprocessing techniques such as normalization (scaling values to [0, 1]) or standardization are often applied to enhance learning efficiency.

3.4.2.2. Convolutional Layers

Convolutional layers are the backbone of CNNs and are responsible for extracting features from input data. Each convolution layer applies multiple filters (kernels) to the input volume to detect spatial patterns such as edges, textures, and shapes.

Mathematical Representation

Given an input tensor $X^{(l-1)}$of shape (H, W, C_{in}), a convolutional layer with C_{out} filters, each of shape (k_h, k_w, C_{in}), computes the output feature map $Z^{(l)}$ as:

(Equation 1)

$$Z^{(l,k)}(p,q) = \sum_{c=1}^{C_{in}} \quad \sum_{u=0}^{K_h-1} \quad \sum_{v=0}^{K_w-1} \quad W_C^{(l,k)}(u,v) \cdot X_C^{(l-1)}(p+u, q+v) + b_k \quad (3.69)$$

where:

- WW represents the filter weights
- b is the bias term

Each filter spatially scans the input by applying the same weights throughout, thereby enforcing translational invariance. Stride and **padding** hyperparameters control the size of the output feature map as in figure 3.2.

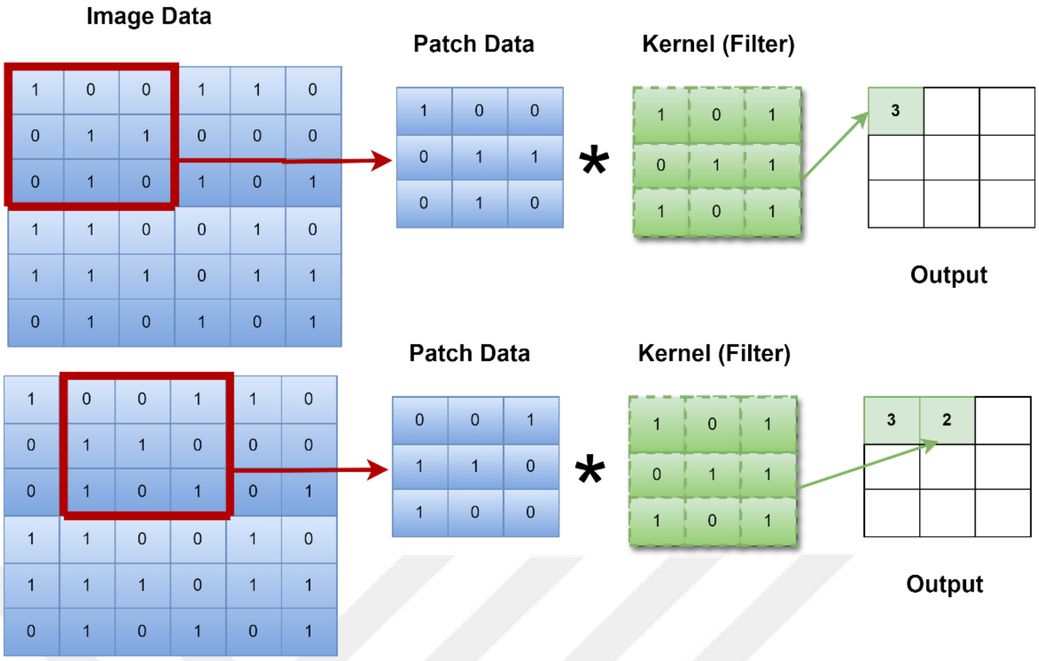


Figure 3.2. Illustration of convolutional padding.

Connection to Other Layers

- The convolutional layer outputs a set of feature maps that typically pass through the **activation function**.
- The result is then moved to a **pooling layer** to reduce spatial dimensions and computational complexity.

3.4.2.3. Activation Functions

Nonlinear **activation functions** were applied after convolutions to introduce complexity into learning representations. The most common type of activation is **a Rectified Linear Unit (ReLU)**.

(Equation 2)

$$ReLU(z) = max\ (0,\ z) \qquad (3.70)$$

This keeps the positive values unchanged while setting the negative values to zero, thereby mitigating the vanishing gradient problem.

Alternative activations include the following.

- **Leaky ReLU**: Allows small negative values, preventing neuron inactivation.
- **Sigmoid** and **Tanh**: Used in certain cases but less common due to saturation issues.
- **Figure 3.3.** illustration graphical activation’s function.

(Equation 3)

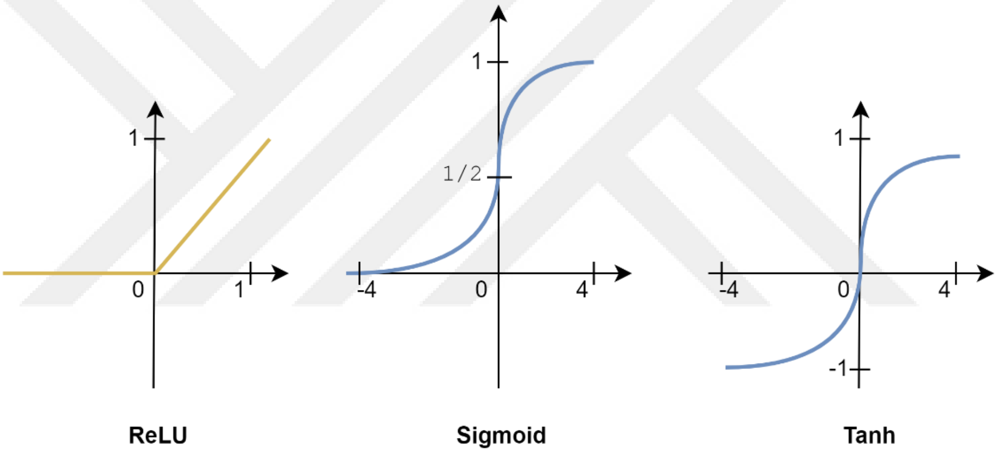


Figure 3.3. Illustration of relu sigmoid and tanh function.

3.4.2.4. Pooling Layers

Pooling layers downsample feature maps and reduce the spatial dimensions while retaining important information. The two primary types are as follows.

- **Max Pooling**: Retains the maximum value in each pooling window.
- **Average pooling**: Compute the average values within each pooling window.

Mathematical Formulation

For a pooling window of size $(m \times n)$ and stride s:

(Equation 4)

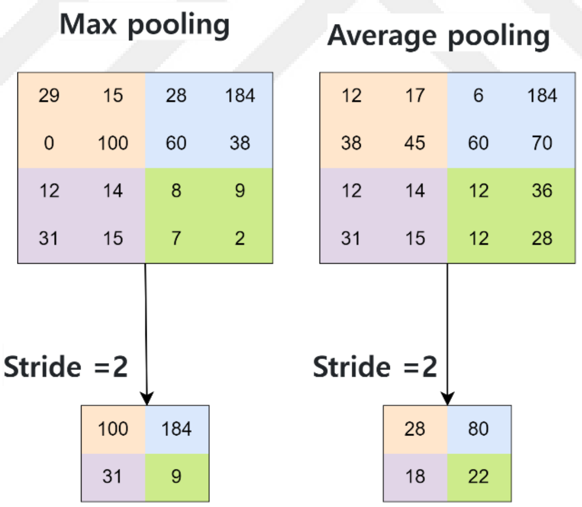
$$Y(i,j) = max_{0 \le u < m, 0 \le v < n} X(i.s + u, j.s + v) \tag{3.71}$$

for **max pooling**, and:

(Equation 5)

$$Y(i,j) = \frac{1}{mn} \sum_{u=0}^{m-1} \quad \sum_{v=0}^{n-1} \quad X(i\,.s + u\,, j\,.s + v) \tag{3.72}$$

for **average pooling**.



29	15	28	184
0	100	60	38
12	14	8	9
31	15	7	2

100	184
31	9

12	17	6	184
38	45	60	70
12	14	12	36
31	15	12	28

28	80
18	22

Figure 3.4. Illustrate how 2D max pooling and avg pooling work, with the following setting: kernel size [2, 2], stride [2, 2].

Connection to Other Layers

- Pooling layers typically follow **convolutional layers** to progressively reduce the feature map size.

- The output then proceeds to **the fully connected layers** for final classification or regression.

3.4.2.5. Fully Connected Layers

Fully connected (FC) layers combine extracted features and perform classification or regression. If the last pooling layer outputs feature maps of shape (H_f, W_f, C_f), they are flattened into a vector $X \in R^N$, where:

(Equation 6)

$$N = H_f \times W_f \times C_f \qquad (3.73)$$

An FC layer with MM output neurons computes

(Equation 7)

$$Z = W_x + b \qquad (3.74)$$

where:

- WW is a weight matrix of shape (M, N)
- b is a bias vector

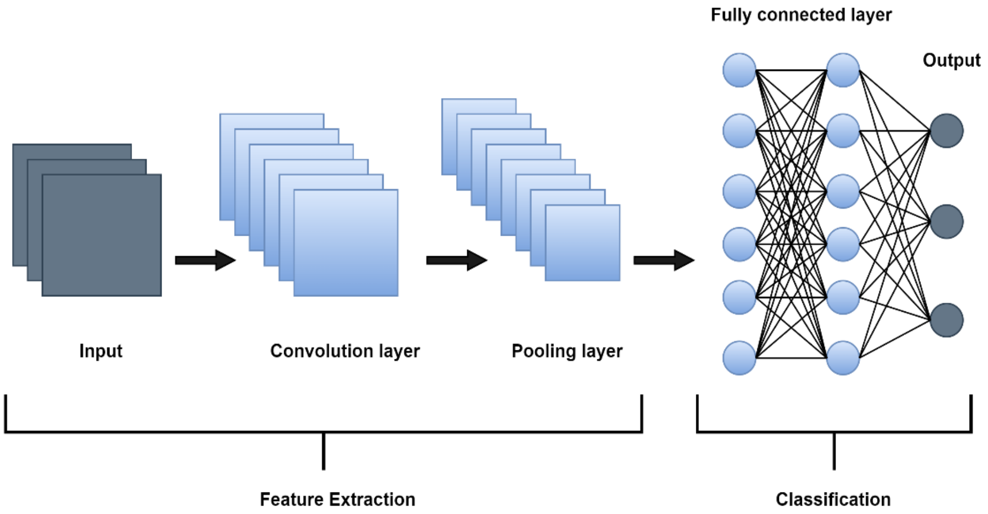


Figure 3.5. Fully connected (FC) layers.

3.4.2.6. Output Layer

The **output layer** depends on the following tasks:

- **Classification**: **Softmax activation** is used to convert raw scores into probabilities

(Equation 8)

$$softmax(z)_i = \frac{e^{zi}}{\sum_{j=1}^{K} \; e^{zj}} \qquad (3.75)$$

- **Regression** uses linear **activation** for continuous value prediction.

PART 4

THE PROPOSED SYSTEM DESIGN AND ALGORITHMS

INTRODUCTION

This chapter details the methodology used for candlestick chart pattern detection and next-candle direction prediction using Convolutional Neural Networks (CNNs). Figure 4.1. shows the entire system proposed in this study. It covers the process, from data collection and preparation to model design and evaluation. First, the data source and its characteristics are described, followed by preprocessing steps, such as cleaning and segmenting the time series into candlestick chart windows. Next, the approach for identifying candlestick patterns using a technical analysis library is explained, along with how these patterns and additional indicators are used to label each chart with a bullish or bearish trend indication. The CNN model architecture is then presented in depth, including its layers, activation functions, regularization through dropout, and image augmentation strategies applied to improve the generalization. Finally, this chapter outlines the training procedure and evaluation protocol, including cross-validation, and defines the performance metrics (accuracy, precision, recall, and F1-score) used to assess the model.

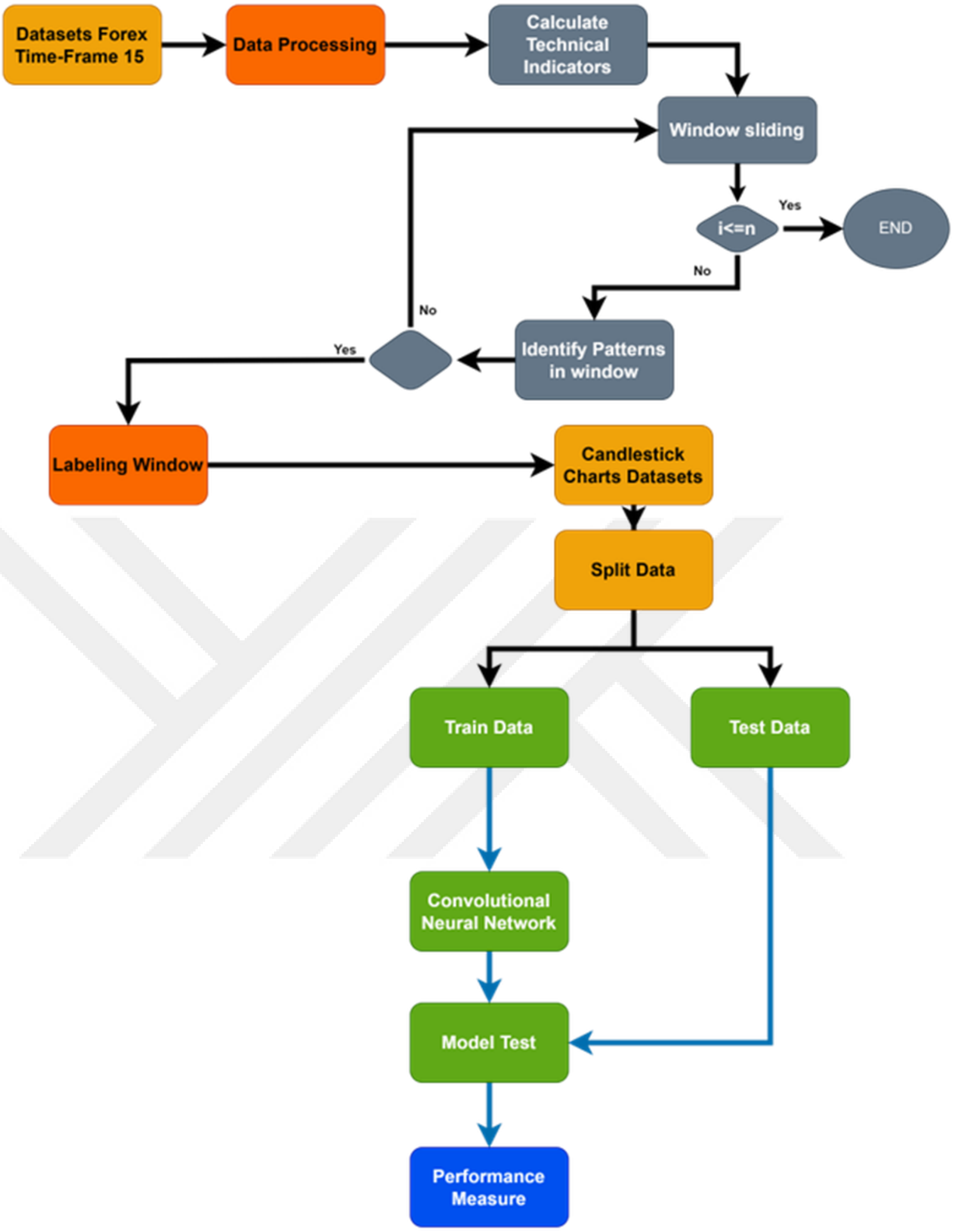

Datasets Forex Time-Frame 15
Data Processing
Calculate Technical Indicators
Window sliding
i<=n
Yes
END
No
Identify Patterns in window
No
Yes
Labeling Window
Candlestick Charts Datasets
Split Data
Train Data
Test Data
Convolutional Neural Network
Model Test
Performance Measure

Figure 4.1. The proposed system.

4.1. DATA COLLECTION

The dataset for this study consists of historical Foreign Exchange (Forex) market data obtained from the Forex Historical Data repository. Specifically, we utilize price data for the EUR/USD currency pair over a long continuous period (from March 2, 2007, to January 1, 2024) sampled at 15-minute intervals. This high-frequency dataset provides a rich sequence of candlestick data that captures intraday market dynamics over many years. Each data record contains the standard fields required for candlestick charting: the opening price, highest price, lowest price, and closing price (OHLC) within each 15-minute interval, along with the trading volume for that interval. The choice of the 15-minute timeframe balances the need for detailed pattern formation (finer than daily or hourly charts) with the practical limits of data size and model complexity. Using such an extensive and granular dataset ensures that the model is exposed to a wide variety of market conditions including different volatility regimes, trends, and outlier events, thereby improving the robustness of pattern detection and prediction.

Before using the data for pattern analysis and model training, initial quality checks were performed. The dataset was inspected for completeness and consistency to ensure that it accurately reflected the market behavior over the selected period. By selecting a well-established historical data source and a major currency pair, we aimed to minimize the issues of data quality and maximize the relevance of the patterns learned by the model to real-world trading scenarios.

4.1.1. Data Preprocessing

Raw financial time-series data often contain irregularities that must be addressed prior to analysis. In this study, data preprocessing was a crucial step in cleaning the historical Forex dataset and preparing it for candlestick pattern detection. Preprocessing focused on handling missing or corrupt records, filtering out anomalies, and then segmenting the continuous series into fixed-length candlestick chart samples suitable for model input. The key preprocessing steps are as follows:

- **Missing Data and Incomplete Records**: Any missing values in the OHLC or volume data were identified and handled to maintain continuity of the time series. For isolated missing entries (e.g., a single 15-minute record missing due to a data glitch), forward filling was applied; the missing price values were replaced with the last known values, under the assumption that a very short gap can be reasonably imputed by the previous state. However, for larger gaps or periods of missing data (e.g., if data transmission was interrupted, resulting in several consecutive missing intervals), the affected records were removed from the dataset to avoid introducing false information. Additionally, a filtering criterion ensured that each candlestick entry was complete; any record lacking any of the OHLC fields or having obviously invalid values was discarded. This step guarantees the integrity of each candlestick so that no malformed candlestick passes into the analysis (each interval must have a valid open, high, low, or close).

- **Candlestick Chart Segmentation (Sliding Window)**: After ensuring the time series was clean and regular, the data was segmented into smaller windows to create candlestick chart samples. We used a sliding window technique to generate segments. Each window spans a fixed number of consecutive 15-minute candlesticks and represents a subchart of the price series. A window size of 20 corresponds to a chart of 20 consecutive candlesticks (covering 20 × 15 minutes = 5 hours of price data). The window is then slid forward through the dataset in a certain step (scroll size) to create the next sample. In our approach, a moderate overlap between windows is allowed to increase the number of training samples while still providing new information. Specifically, we used a step size equal to half the window length (50% overlap) to balance the sample diversity with the efficient use of data. This means that if, say, a window covers time steps t=1 to t=20, the next window starts at t=11 (half overlap) and covers t=11 to t=30, and so on. The use of overlapping windows ensures that important patterns that may span across window boundaries can be captured in at least one window, as shown in Figure 4.2.

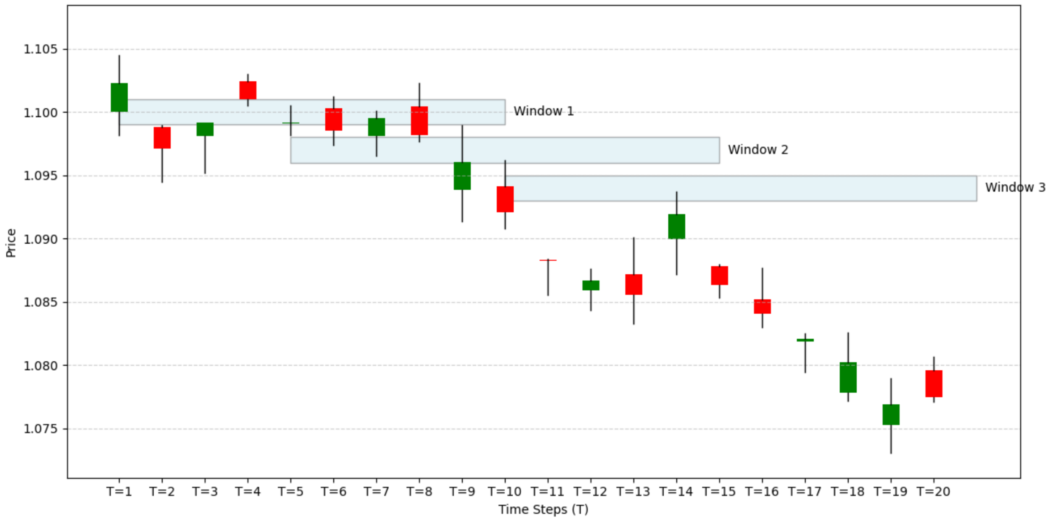
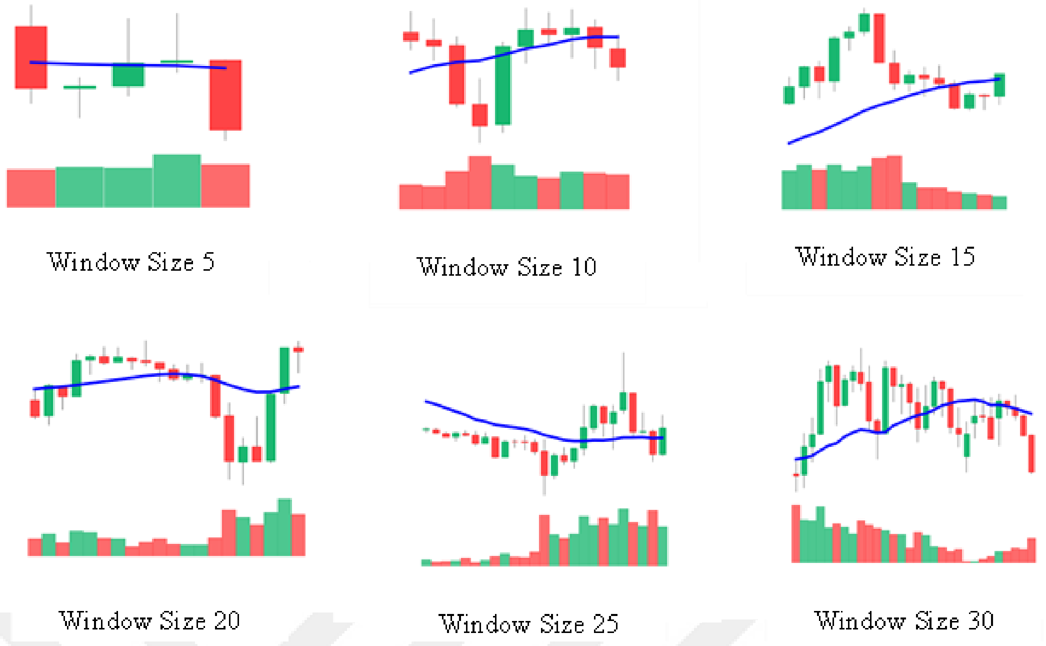


Figure 4.2. Sliding window for window =10 and shift size =5.

We experimented with multiple window sizes (5, 10, 15, 20, 25, and 30 candlesticks per window) to determine which length best captured the relevant patterns for prediction. Figure 4.3. shows the window sizes used in this search. These window sizes correspond to different trend durations (from short-term patterns lasting 1.25 hours up to longer patterns spanning 7.5 hours). By evaluating various window lengths, we ensured that the model was not unduly biased by an arbitrary timeframe choice. Instead, we c.an select an optimal window size that provides the highest predictive performance (this selection process is discussed in the results, but the methodology of testing various sizes is part of our approach). Once the window size was decided, all time-series data were segmented accordingly, yielding a large collection of candlestick chart samples ready for pattern identification and labeling.



Window Size 5

Window Size 10

Window Size 15

Window Size 20

Window Size 25

Window Size 30

Figure 4.3. Different window sizes with (5,10,15,20,25, and 30).

Each resulting window of data was then converted into a candlestick chart image. To do this, the OHLC values in the window were used to draw candlesticks (typically green/white for bullish candles and red/black for bearish candles) on a fixed-size image canvas. For consistency, all chart images were generated at the same resolution (in our case, 150×150 pixels with three color channels to mimic an RGB candlestick chart). This transformation from raw numerical data to images allows us to leverage image-based CNN techniques for pattern recognition. Visual encoding preserves the shape and sequence of candlesticks, enabling the CNN to learn from chart patterns in a manner similar to how a human trader might visually identify patterns.

4.1.2. Image Augmentation Strategies

When training image-based CNNs, it is beneficial to artificially augment the dataset to improve generalization. We employed real-time image augmentation on candlestick chart images using Keras ImageDataGenerator. Several augmentation techniques were randomly applied to each training image during the training epochs, including.

- **Rescaling**: As previously mentioned, all images were normalized by scaling the pixel intensities by 1/255. This is standard preprocessing rather than augmentation, but it ensures a uniform input range.
- **Random Shear**: A small shear transformation (skewing the image) was applied with a shear intensity up to 0.2 (20%). This can tilt the candlesticks slightly, which simulates minor variations in how the chart might be rendered or small timing misalignments, helping the model to not rely on exact symmetric shapes.
- **Random Zoom**: We applied random zoom-in or zoom-out of up to 20%. Zooming changes the scale of candlestick shapes in the image. By training on slightly zoomed versions, the model learns to recognize patterns at different scales (a pattern that is a bit larger or smaller in terms of pixel size should still be recognized as the same pattern).
- **Shifts/Flips**: The charts were not flipped horizontally because a mirror-image reversal in time would not represent a valid scenario (time cannot reverse), and vertical flips would invert upward moves downwards, which is not meaningful for pattern integrity. However, we allowed slight horizontal or vertical translations (shifting the image by a few pixels) as a part of the shear/zoom combination.

These augmentation techniques were applied on-the-fly during training, meaning that each epoch saw a new random variation in some charts. Augmentation increased the effective size of the training set and exposed the model to a broader set of input conditions, thereby reducing overfitting. By the end of the training, the CNN becomes more robust to variations in chart appearance that are not relevant to the pattern (for instance, slight changes in scale or alignment), focusing instead on the essential visual features of bullish or bearish patterns.

4.2. PATTERN IDENTIFICATION USING TA-LIB

With the candlestick chart segments prepared, the next step was to identify the known candlestick patterns within each segment. We utilized the Technical Analysis Library (TA-Lib), a widely used library for financial data analysis that includes a comprehensive set of functions for recognizing candlestick chart patterns. TA-Lib provides algorithms to detect **61 distinct classical candlestick patterns**, encompassing a broad range of single-day and multi-day formations commonly studied by technical analysts. These include patterns such as **Doji**, **Hammer**, **Shooting Star**, **Engulfing Pattern**, **Morning Star**, **Three White Soldiers**, and many others, as defined in the technical analysis literature.

For each candlestick window, the OHLC data were input into TA-Lib's pattern recognition functions of the TA-Lib. Each pattern function examined a specific configuration of one or more consecutive candlesticks. For example, the Hammer pattern function evaluates whether a given candlestick in the window has a long lower wick and short body positioned near the top (typical of a bullish hammer), whereas the engulfing pattern function looks at two consecutive candlesticks to see if the second candle’s body fully engulfs the previous one. TA-Lib functions typically return an indicator value (often an integer flag), signaling the presence of a pattern and sometimes the direction (bullish or bearish nature) of that pattern. By running through all relevant TA-Lib candlestick functions in each window, we automatically detected which, if any, of the 61 patterns were present in that sequence of candlesticks. In many cases, no known pattern might be present in a given window; in other cases, one or more patterns could be identified (for instance, a window might contain both a "Doji" and an "Engulfing" pattern, if those occur in overlapping fashion).

This programmatic pattern identification ensured objectivity and consistency in recognizing patterns compared to manual visual inspection, which could be error-prone or subjective. It also significantly expanded the scope of the patterns considered, as our methodology is not limited to a small subset of patterns; virtually all well-documented candlestick formations were included. The outcome of this step is an annotation for each window indicating which candlestick pattern(s) it contains and the nominal interpretation of those patterns (each pattern inherently has a bullish or bearish implication, as per technical analysis theory).

4.3. DATASET LABELING FOR TREND PREDICTION

Once the candlestick patterns were identified in each window, the next crucial step was to label each window with the outcome. We wanted the model to predict whether the pattern in that window would signal a bullish or bearish trend in the immediate future. The goal is to classify each candlestick chart (window) as either **bullish (expected uptrend)** or **bearish (expected downtrend)**, which corresponds to predicting the direction of the next candlestick or near-term price movement following that window.

Labeling was performed using a combination of pattern interpretation and technical indicator confirmation to increase the reliability.

4.3.1. Pattern-Based Trend Inference

Each detected candlestick pattern has a known, traditional interpretation. A Hammer or a Morning Star pattern is typically considered a bullish reversal signal (anticipating an upward move), whereas a Shooting Star or a Dark Cloud Cover is considered bearish. Initially, we assigned a tentative label to the window based on the identified primary pattern. If multiple patterns were found in the window, the most significant or latest pattern (often the one formed by the last few candlesticks in the window) was used for trend indication. This gives an initial bullish/bearish label derived from classical technical analysis.

4.3.2. Technical Indicator Confirmation

Relying solely on patterns can be misleading, because candlestick patterns do not always correctly predict market movements in isolation. To improve labeling accuracy, we incorporated additional technical indicators as confirmation signals for the pattern indication. In our methodology, we use a Simple Moving Average (SMA) for key confirmation. The SMA has a smoothed average of closing prices over a recent period and helps indicate short-term trend momentum. For each window, we computed a short-term SMA (e.g., a moving average spanning a small multiple of the window length or another relevant period). If the pattern was bullish, we checked whether the price relative to this SMA or the slope of the SMA supported an upward momentum (for instance, the closing price moving above the SMA or the SMA curve turning upward). Similarly, for a bearish pattern, we verified whether the SMA indicated downwards momentum (price breaking below the average or sloping downwards). These indicators serve as a **confirmatory filter**; if a recognized pattern suggests a trend, but the indicators contradict it, the trend signal is deemed weak or possibly false. In such cases, the window might be labeled according to the indicator (or even excluded if uncertain), whereas when both the pattern and indicators align, the label is assigned high confidence.

Through this combined approach, each candlestick window was classified into one of two classes: **bullish** (meaning that the next candle or short-term movement is expected to go up) or **bearish** (meaning that a downwards movement is expected). By using technical indicators to guide labeling, we ensured that the model's training data reflected scenarios in which a pattern’s implication was corroborated by market momentum, thereby focusing the learning on more reliable pattern outcomes. This is important because candlestick patterns can occasionally appear without resulting in the expected trend change; including these cases as positive examples could confuse the model. Our labeling strategy mitigates this by providing the model with labels that have a higher probability of being correct, given additional confirmation.

The final labeled dataset consisted of thousands of candlestick chart images, each with an associated binary label (uptrend or downtrend). These labeled data serve as the ground truth for training the CNN model to recognize patterns and predict the next-candle direction. Figure 4.4. shows the labeling process using a simple average indicator.

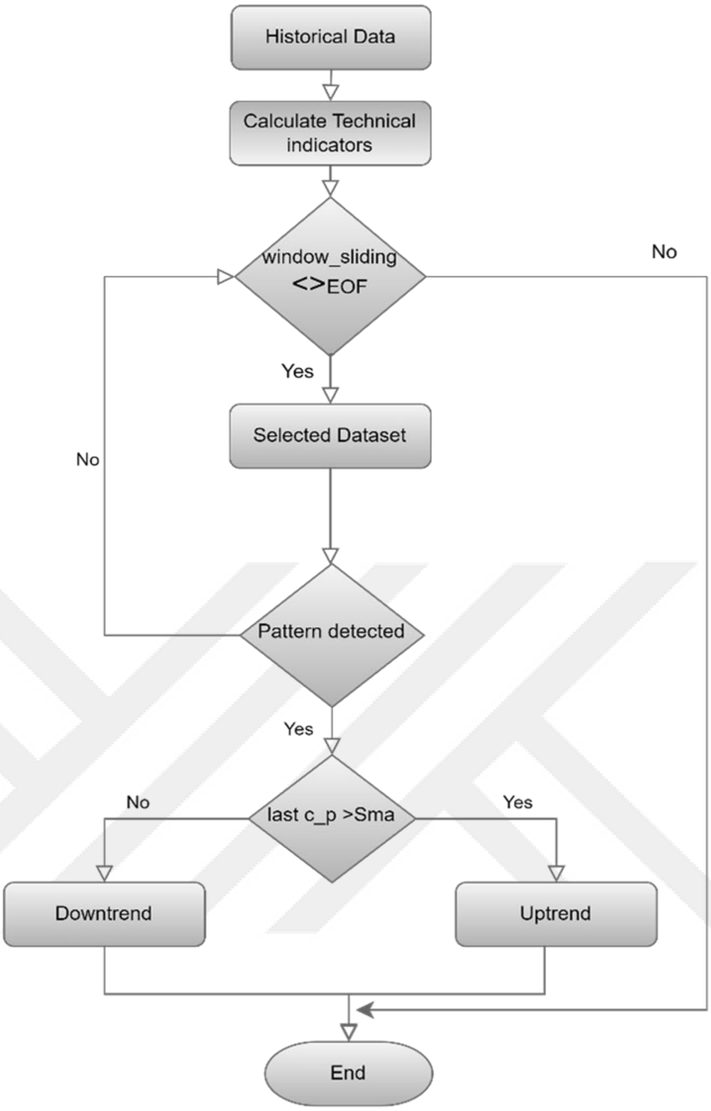


Figure 4.4. The labeling process.

4.4. CNN MODEL ARCHITECTURE

To automatically classify candlestick chart images as bullish or bearish, we designed a Convolutional Neural that is tailored for image-based pattern recognition. The CNN model uses the generated candlestick chart images as input and learns to output a prediction of the trend class. The architecture was implemented using TensorFlow and Keras deep learning frameworks, and it comprises several layers that progressively extract visual features from the candlestick charts and then perform binary classification, figure 4.5. present The CNN architecture representing the sequential processing of an input image. In constructing the CNN architecture, a series of carefully curated layers was incorporated, each serving a distinct function while encompassing a specific set of parameters. The layout of these layers is comprehensively depicted in Table 4.1., which details each layer's type, resultant output shape, number of trainable parameters, and overall contribution to the functionality of the model.

The key components of the model architecture are described in detail below:

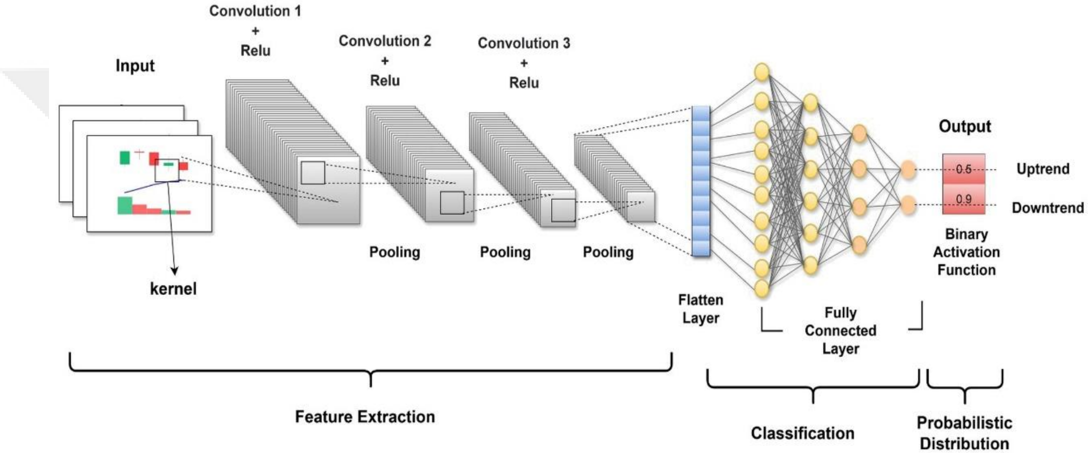


Figure 4.5. CNN architecture representing sequential processing of the input image.

- **Input Layer**: The input to the model is an image of a candlestick chart with a fixed size of 150×150 pixels and three-color channels (RGB). Each image encodes the sequence of candlesticks in the window, and is drawn in a standard candlestick chart format. Before feeding the images into the network, pixel values were normalized (rescaled) to the range [0, 1] to facilitate faster and more stable training (original pixel intensities 0–255 were divided by 255).
- **Convolutional layers**: CNN use multiple convolutional layers to automatically learn features from candlestick images. The first convolutional layer applied a set of learnable filters (kernels) to the input image. In our architecture, this layer uses 32 filters of size 3×3, with each filter scanning across a 150×150 image. As the filter slides over the image, it performs convolution operations (dot products between the filter weights and local patch of the image), producing a feature map

as the output. Each filter can learn to detect basic visual features such as vertical or horizontal edges, shapes of candlestick bodies/wicks, or texture in the chart. The **rectified linear unit (ReLU)** activation function was applied to the convolution output, introducing nonlinearity by eliminating negative responses. This helps the network model complex patterns and interactions between pixels, beyond linear combinations.

Following the first convolution, a **max-pooling** layer was used. We apply 2×2 max pooling, which reduces the spatial dimensions of the feature maps by taking the maximum value in each 2×2 region. In effect, the 150×150 feature map is down-sampled (pooled) to roughly 75×75 (if using 'same' padding, exact dimensions may be 75 or 74 depending on rounding). Pooling serves two main purposes: it reduces the computational load for subsequent layers by decreasing the number of activations and it provides a form of translation invariance (small shifts in the input candle position will not drastically change the pooled representation). It also helps generalize by abstracting the features.

After pooling, we stacked additional convolutional layers to learn higher-level representations. The second convolutional layer might use 64 filters (also of size 3×3) applied to the pooled feature maps from layer one. This layer can capture more complex combinations of features detected by the first layer (for example, it may recognize parts of candlestick formations or edges of patterns). Again, ReLU activation is applied, and another pooling layer reduces the output (e.g., down to ~37×37). A third convolutional layer with an even larger number of filters, such as 128 filters of size 3×3, can capture even more abstract patterns (potentially learning to recognize entire candlestick shapes or arrangements corresponding to specific patterns). We included three convolutional layers in our model (with 32, 64, and 128 filters), each followed by a max-pooling layer of size 2×2. At the end of this convolutional stack, the image was transformed into a set of distilled feature maps. The progressive increase in filter count allows the model to learn a richer set of features at each stage, while pooling gradually reduces the spatial size, focusing the model on *what* patterns are present rather than their exact position in the image. This is useful because, for example, a hammer pattern can occur at different vertical positions in the chart, depending on the price scale, and we want the model to recognize it regardless of the exact pixel location.

- **Flattening and Fully Connected Layers**: After last pooling layer, the feature maps were "flattened" into a one-dimensional vector. This flattening converts spatially organized features into a format that can be fed into dense (fully connected) layers. The first dense layer in the architecture contained 512 neurons. Each neuron in this layer receives input from all the features of the previous layer, allowing it to combine them in arbitrary ways. We also use ReLU activation on this layer, which enables the network to learn nonlinear combinations of convolutional features. In essence, this layer learns to interpret high-level features (extracted by the convolutional layers) in the context of the classification task. For instance, it might learn neurons that activate strongly when certain combinations of edges and shapes corresponding to a "bullish engulfing" pattern are present.
- **Dropout Regularization**: To prevent overfitting (where the model memorizes training examples rather than generalizing them), we incorporate a dropout layer after the first dense layer. The dropout randomly sets a fraction of the neurons of the layer to zero for each training batch. We used a dropout rate of 0.5 (50%), that is, half of the neurons in the dense layer were dropped at each training step. This forces the network to not rely too heavily on any single feature or a co-dependent set of features, improving its ability to generalize to unseen charts. Essentially, dropout simulates an ensemble of many smaller networks and helps to ensure that the learned features are robust and not specific to particular training samples.
- **Output Layer**: The final layer of the CNN is a dense output layer that produces a prediction. Because this is a binary classification (bullish vs. bearish), we used a single neuron in the output layer with a **sigmoid activation** function. The sigmoid squashes the output to a value between 0 and 1, which can be interpreted as the probability of the input image belonging to the "Bullish" class (for example). A threshold (typically 0.5) would then be used to decide the class label: outputs >= 0.5 are categorized as bullish, and outputs < 0.5 as bearish. (Alternatively, the network could be designed with two output neurons

and a softmax activation for the two classes, but the single sigmoid is a simpler equivalent approach for binary outcomes.) The goal of the model during training is to adjust the weights such that these output probabilities align with the true labels of the training images.

Table 4. 1. Architectural details of CNN model.

Layers	Output Shape	N. Parameters	Description and Functionality
Input Layer	(None, 150, 150, 3)	0	Accepts input images with dimensions 150×150 pixels and three-color channels (RGB).
Convolutional Layer 1	(None, 148, 148, 32)	896	32 filters of size (3×3), Activation: ReLU. Extracts basic edge features.
MaxPooling Layer 1	(None, 74, 74, 32)	0	Pool Size: (2×2). Reduces spatial dimensions by half.
Convolutional Layer 2	(None, 72, 72, 64)	18,496	64 filters of size (3×3), Activation: ReLU. Captures higher-level features.
MaxPooling Layer 2	(None, 36, 36, 64)	0	Pool Size: (2×2). Further downsampling for computational efficiency.
Convolutional Layer 3	(None, 34, 34, 128)	73,856	128 filters of size (3×3), Activation: ReLU. Detects complex patterns.
MaxPooling Layer 3	(None, 17, 17, 128)	0	Pool Size: (2×2). Compresses feature maps while retaining important information.
Flatten Layer	(None, 36,992)	0	Convert 3D feature maps into a 1D vector for fully connected layers.
Fully Connected Layer 1	(None, 512)	18,940,416	Dense Layer: 512 units, Activation: ReLU. Learning high-level abstract features.
Dropout Layer	(None, 512)	0	Dropout Rate: 0.5. Mitigates overfitting by randomly deactivating nodes.
Output Layer	(None, 1)	513	Dense Layer: 1 unit, Activation: Sigmoid. Produces binary classification output.

Total Parameters:	19,034,177 (72.61 MB)
Trainable Parameters:	19,034,177 (72.61 MB)
Non-trainable Parameters:	0 (0.00 Byte)

4.5. TRAINING AND EVALUATION PROCEDURE

Furthermore, Table 4.2 delineates the key training and compilation parameters that guided model development. These parameters encompass data preprocessing techniques such as rescaling and data augmentation alongside the chosen optimizer, loss function, batch size, and performance metric. In concert, they form the experimental framework, ensuring reproducibility and bolstering the robustness of the training process. When considered alongside the information provided in the preceding table, these details collectively offer a comprehensive picture of the CNN’s architecture and the methodological foundation for its training and evaluation.

Following the construction and compilation of the CNN model, the next phase involved training on a curated dataset of candlestick chart images labeled as bullish or bearish. This stage was meticulously organized to both optimize the model’s performance and thoroughly gauge its predictive capabilities. Several pivotal aspects of the training workflow and validation strategies are summarized below.

- **Training/Validation/Test Split**: The labeled dataset was divided into three subsets to enable unbiased evaluation of the model. We used a typical split, where the majority of the data were used for training and smaller portions were reserved for validation during training and final testing. Approximately 70% of the candlestick chart samples were used for training, 10% for validation, and 20% were used as the test set. The **training set** was used by the model to learn the patterns (the model weights were updated by running through these data). The **validation set** is not used to update the model weights; rather, it monitors the model's performance on unseen data during training, which helps in tuning the hyperparameters and deciding when to stop training (if the validation accuracy stops improving, it may indicate that the model is starting to overfit the training data). Finally, the **test set** was kept completely untouched during training and was used to evaluate how well the trained model generalizes entirely new data.

- **Model Compilation and Hyperparameters**: CNN was compiled using an appropriate loss function and optimizer for binary classification. Binary Cross-Entropy was used as the loss function, which is the standard for probabilistic binary output models. This loss function measures the divergence between the predicted probabilities and the actual binary labels, penalizing incorrect predictions as they become confident and wrong. Minimizing this loss guides the model to output probabilities close to 1 for bullish charts that are truly bullish and close to 0 for bearish charts, and vice versa. We employed the **Adaptive Moment Estimation (Adam**) optimization algorithm with a carefully chosen learning rate as the optimizer. Adam is well-suited for this task as it adapts the learning rate for each parameter and generally converges faster and more reliably than basic stochastic gradient descent. Initially, we tried the default learning rate (0.001) but found that a slightly lower learning rate (such as 0.0003) provided more stable learning for our model, likely due to the complexity of the model and the risk of overshooting the minimum with too large steps. The training was performed in mini-batches (we set a batch size of 64 images per gradient update), which balances noise in the gradient estimation with computational efficiency (a batch of 64 was a good fit for the hardware memory and provided stable convergence). We trained the model for a fixed number of epochs (e.g., 20 epochs initially) while monitoring performance. The number of epochs was determined based on the observations when the validation loss/accuracy leveled off. In some runs, we also employed an **early stopping** strategy: if the validation performance did not improve for a certain number of consecutive epochs, the training was halted to prevent overfitting and save time.
- **Hyperparameter Tuning**: Several hyperparameters (parameters of the training process or model architecture that are not learned from the data but set by the researcher) were tuned to achieve the best performance. These include the learning rate, batch size, number of training epochs, and architecture

choices, such as the number of layers or neurons in the dense layer. We performed a combination of manual and systematic search. For instance, we tried batch sizes of 32 and 64 and found that 64 gave slightly better generalization. We experimented with learning rates in the range of 1e-2 to 1e-4 and discovered that too high a learning rate caused the training to diverge or oscillate, while too low made convergence very slow; hence, the choice of 3e-4 as a middle ground yielded smooth learning curves. We also validated the choice of the 512-neuron dense layer by trying a smaller dense layer (256 neurons) and noted a minor drop in accuracy, indicating that 512 was appropriate for capturing the complexity of the patterns. The window size for candlestick charts, as mentioned earlier, was also a critical hyperparameter, and our experiments with different window lengths on a validation set guided the selection of an optimal window size that the model used for the final training. All of these decisions were made based on the model’s performance on the validation set to avoid biasing the model to the test data.

Table 4.2. Training hyperparameters and compilation parameters used in the CNN model.

Parameter	Value
Input image dimensions	150 × 150
Batch size	64
Image rescaling factor	1/255
Data augmentation	Shear = 0.2, Zoom = 0.2
Optimizer	Adam
Loss function	Binary Crossentropy
Performance metric	Accuracy
Number of epochs	20
Learning rate	0.0003

4.6. K-FOLD CROSS-VALIDATION

In addition to the single hold-out validation set, k-fold cross-validation was employed to verify the robustness of the model further. Specifically, we used a 5-fold cross-validation approach for the training dataset. In 5-fold cross-validation, the training data were split into five equally sized folds (subsets). The model is trained five separate times, each time using a different fold as a validation set and the remaining four folds as the training data. As shown in Figure 4.6., this represents a K-fold. After these five training runs, we obtained five validation performance scores (accuracy and F1-score for each fold). We computed the average and standard deviation of these scores to assess the sensitivity of the model to a particular composition of the training data. This technique ensures that the performance of our model is consistent and not a fluke of a particular train validation split. It also aids in hyperparameter tuning; if a certain hyperparameter setting yields consistently high validation performance across all folds, it is likely a robust choice. In our methodology, the cross-validation results provided additional confidence in the chosen model configuration and insight into the variance. For instance, a small variance in performance across folds indicates that the model generalizes well, whereas a high variance might have suggested overfitting to some data patterns or the need for more data/regularization.

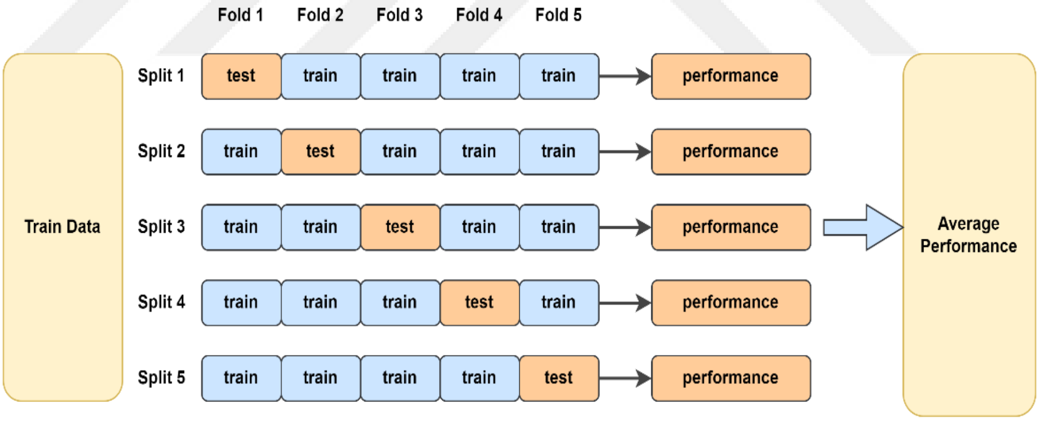


Figure 4.6. k-fold data partitioning.

- **Training Process**: During training, at each epoch, the CNN processed all training batches and updated their weights via backpropagation to minimize the loss. After each epoch, the model was evaluated using the validation set. We plotted the learning curves of training versus validation loss and accuracy to monitor the training progress. Typically, we observed that training accuracy increases over epochs and eventually plateaus, whereas validation accuracy follows a similar trend up to a point where it might start to diverge (indicating overfitting). We used validation metrics to determine the final model (for instance, we could roll back to the model weights from the epoch with the best validation accuracy if early stopping was not triggered automatically).

- **Model Selection and Testing**: After training, we selected the model with the best validation performance (either the final epoch if no overfitting was observed or an earlier epoch if the early stopping criterion was triggered) as the final model. This model was then evaluated on an independent test set that was previously set aside. Importantly, the test set consists of chart samples that the model has never seen during training or validation. We used this set to compute the final performance metrics, providing an unbiased assessment of how the model would perform on new real-world data. This step simulates the scenario of deploying the model for live predictions and tests whether the model can correctly classify unseen candlestick charts as bullish or bearish based on learned patterns.

Throughout the training and evaluation, careful measures were taken to ensure that the results were reliable and not due to random chance or overfitting. The combination of a separate test set and cross-validation provided a comprehensive view of the performance of the model.

4.7. PERFORMANCE METRICS

To evaluate the effectiveness of the candlestick pattern prediction model, we measured several performance metrics commonly used in binary classifications. These metrics provide insight into the different aspects of a model’s predictive performance.

- **Accuracy**: This metric measures the overall correctness of the model and is defined as the number of correct predictions divided by the total number of predictions. In terms of confusion matrix components (True Positives, True Negatives, False Positives, False Negatives), the accuracy is (TP + TN) / (TP + TN + FP + FN). An accuracy of, say, 0.90 indicates that 90% of all candlestick chart instances (bullish or bearish) were correctly classified by the model. While accuracy provides a quick summary of performance, it can be misleading if classes are imbalanced. In our dataset, we tried to maintain a balance between bullish and bearish examples through the data collection and labeling process, but we still considered additional metrics for a nuanced evaluation.

- **Precision**: Precision focuses on the quality of positive predictions (for instance, we can arbitrarily define “bullish” predictions as the positive class). It is calculated as TP / (TP + FP). Precision answers the question: "Of all the instances that the model predicted as bullish, what fraction were actually bullish?" High precision means that when the model predicts an uptrend, it is usually correct, that is, there are few false alarms. This is particularly important in the trading context because a false positive (predicting a bullish trend when none occurs) might lead to poor trade. For example, if our model identifies 100 chart patterns as bullish and 90 of them truly lead to upward movement, the precision is 0.90, indicating a fairly reliable bullish signal.
- **Recall**: Recall (also known as sensitivity or true positive rate) looks at how well the model captures actual positive instances. It was computed as TP / (TP + FN). Recall answers: "Of all the truly bullish instances (patterns that did lead to an uptrend), what fraction did the model successfully detect as bullish?" A high recall indicates that the model misses very few true bullish signals (i.e., few false negatives). In market terms, a false negative would be missing a profitable bullish pattern (the model predicts bearish or fails to act when an uptrend is coming). For instance, if there were 100 actual bullish cases, and the model caught 80 of them, the recall was 0.80. There is often a trade-off between precision and recall (capturing more positives can lower precision, and vice versa), so both are important to consider.
- **F1-Score**: The F1-score is the harmonic mean of Precision * Recall, calculated as 2 × (Precision × Recall) / (Precision + Recall). It provides a single metric that balances the precision and recall. The F1-score is particularly useful when the class distribution is uneven or when one wants to seek a balance between avoiding false alarms and missing true signals. The F1-score reaches its best value at 1 and the worst at 0. For instance, if the precision is 0.9, and recall is 0.8, the F1-score is 2*(0.9*0.8)/ (0.9+0.8) = 0.847. We considered the F1-score

as a primary metric for model selection because it ensures that the model performs well in identifying bullish patterns without too many false positives, and does not miss a large portion of them. In other words, it summarizes the ability of the model to make accurate and comprehensive predictions for the positive class. (For binary classification, we can also compute precision, recall, and F1 for the negative class if needed, but usually one focuses on the "interesting" class, here bullish, or reports metrics for both classes separately. In our case, because bullish vs bearish were roughly balanced and equally important, we report the overall precision/recall which is effectively the same for one class if we treat the other as negative.)

In addition, we also examined the confusion matrix to see the breakdown of predicted versus actual classes and monitored if there was any bias toward predicting one class over the other. By analyzing precision and recall together, we ensured that our model was not only accurate overall but also behaved well in terms of trade-off. For instance, if the model had very high accuracy but low recall, it would mean it was very conservative in predicting bullish (perhaps only calling very obvious patterns bullish and missing many subtle ones). We aimed for a model with both high precision and high recall, leading to a high F1 score, indicating a reliable and consistent performance.

Overall, these performance metrics provided a comprehensive evaluation of the CNN model. In the following chapter, we will present the results obtained using these metrics and discuss how well the model performed in predicting next-candle directions from candlestick chart patterns, as well as compare it with benchmarks or other approaches, if applicable. The methodological rigor outlined in this chapter, from data preparation to validation, is intended to ensure that the results are credible and that the model can be trusted for practical use in analyzing financial market trends.

PART 5

THE RESULT AND DISCUSSION

5.1. THE RESULT

In the methodology section, we explain how subcharts containing JCs patterns were generated to enhance the analysis. Our initial objective was to identify the most suitable window size for training our CNN model because the window size is among the most critical hyperparameters for capturing meaningful time-series patterns.

We evaluated multiple window sizes, specifically 5, 10, 15, 20, 25, and 30, to find the optimal choice. The type of data modeled by each window size differs; a smaller window size may capture finer short-term trends, whereas a larger window size can reflect more general long-term patterns. In addition, we fixed the transformation size to half the window size across all configurations, creating a 50% overlap between consecutive windows. This overlap is crucial for preserving data continuity, allowing the model to shift from one window to another without discarding essential information. Such an approach not only bolsters the model’s ability to generalize across various data segments but also avoids abrupt changes that can negatively affect learning.

We constructed our dataset meticulously, emphasizing the chosen window sizes and strategic overlapping to best represent the underlying temporal dynamics of the JCs charts. These intricate patterns are central to the accurate classification or prediction in our binary tasks. This comprehensive approach laid the groundwork for the successful training and evaluation of the CNN models. Once a window with recognized candlestick patterns was identified, the Technical Analysis Library (Ta-lib) helped determine whether any of the 61 predefined patterns were present. If so, the window was progressed for further investigation. The next phase entailed classifying the window as bullish or bearish. Market trend classification is supported by an array of technical indicators that mainly focus on the price of the final candle relative to the moving average. We tested moving averages of 20, 50, and 200 periods but ultimately selected a 20-period moving average for this study. If the last candle price was higher than the moving average, the window indicated an upward trend, whereas a lower price signaled a downward trend. This framework integrates both candlestick pattern detection and pivotal technical indicators to analyze trends. In the Forex market, trading patterns evolve continuously, with each pattern emerging differently. This ever-shifting pattern landscape challenges traders and analysts, complicating any attempt to track them. The ongoing nature of pattern mining has highlighted the complexity and fluidity of the market.

Table 5.1. demonstrates how TA-Lib contributed to the pattern analysis, revealing how often each pattern in the library appeared. The table displays the names of each pattern along with the frequency with which it emerged in our dataset, offering a direct view of which patterns are dominant and possibly indicative of specific trends. Notably, some patterns became more frequent, whereas others diminished.

Table 5.1. Frequency of various candlestick patterns within a 15-min timeframe.

Pattern	Occurrences	Pattern	Occurrences
Spinning Top	19,678	Stalled Pattern	346
Long Line Candle	19,058	Evening Star	340
Belt-Hold	17,318	Three-Line Strike	337
Short Line Candle	15,741	Identical Three Crows	250
Closing Marubozu	14,445	Morning Doji Star	106
Doji	14,133	Evening Doji Star	104
Hikkake Pattern	11,894	Modified Hikkake Pattern	119
High-Wave Candle	11,640	Thrusting Pattern	117
Rickshaw Man	9,553	Three Advancing White Soldiers	86
Engulfing Pattern	7,990	Piercing Pattern	76
Marubozu	5,659	Dark Cloud Cover	68
Harami attern	4,370	Homing Pigeon	47

Three Outside Up/Down	3,854	On-Neck Pattern	32
Hammer	2,776	Stick Sandwich	32
Gravestone Doji	2,047	In-Neck Pattern	27
Dragonfly Doji	1,942	Tristar Pattern	24
Takuri (Dragonfly Doji with long lower shadow)	1,906	Three Black Crows	24
Hanging Man	1,537	Tasuki Gap	19
Matching Low	1,474	Unique 3 River	9
Harami Cross Pattern	1,082	Ladder Bottom	7
Doji Star	967	Breakaway	4
Three Inside Up/Down	793	Abandoned Baby	1
Upside/Downside Gap Three Methods	719	Two Crows	1
Shooting Star	699	Counterattack	1
Separating Lines	624		

Within our 15-minute historical forex dataset, a few patterns did not appear at all, such as “Kicking,” “Kicking-bull/bear determined by the longer marubozu,” “Three Stars in the South,” “Concealing Baby Swallow,” “Mat Hold,” and “Upside Gap Two Crows.” Their absence suggests that neither the market conditions nor the timeframe fulfilled the requirements for their formation. Some patterns occur rarely (as seen in Table 5.1.), which, while uncommon during normal market activity, may become essential indicators for significant market shifts.

To understand how varying window sizes and shifts affect the image dataset creation process for CNN-based binary classification, we tested six unique window size and shift configurations. Our goal was to determine how each combination modified a time-series dataset.

- **Window Size 5, Shift 2**: Yielded 6,292 images across two classes. These were partitioned into 4,403 for training, 628 for validation, and 1,261 for testing.
- **Window Size 10, shift 5**: produced 4,970 images, with 3,478 for training, 496 for validation, and 996 for testing.

- **Window Size 15, Shift 7**: 11, 130 images were created, with 7,790 images for training, 1,112 for validation, and 2,228 for testing.
- **Window Size 20, Shift 10**: This resulted in 3,675 images, split into 2,572 for training, 366 for validation, and 737 for testing.
- **Window Size 25, Shift 12**: Generated 6,541 images, allocated as 4,578 for training, 653 for validation, and 1,310 for testing.
- **Window Size 30, shift 15**: Created 5,181 images, with 3,626 for training, 517 for validation, and 1,038 for testing.

Each configuration helped assess the CNN's ability to recognize and classify pattern changes with different window sizes and overlaps. These datasets informed the training, validation, and testing processes factors that affect how well the model generalizes to new, unseen data. This extensive exploration of window sizes and shifts informs the best practices in data preparation for CNNs, particularly in tasks that rely on detecting subtle temporal patterns. We also compiled a table detailing the resolution of each window (see Table 5.2.), which adds further insight into how window dimensions can shape the overall model performance and offer future directions for optimal configurations. Figure 5.1. highlights the CNN model's training performance, illustrating how accuracy and loss evolve over successive epochs for both the training and validation sets. Figure 5.2. presents the receiver operating characteristic (ROC) curves, which reveal the classification model's effectiveness. In nearly all scenarios, the ROC curves approached an area under the curve (AUC) of 1.00, indicating an almost ideal classifier that achieves a high true positive rate across multiple thresholds while maintaining a negligible false positive rate.

Table 5.2. The performance results according to common metrics obtained from the implemented model

Window size	Shift size	Precision	Recall	F1-Score	Accuracy
5	2	0.993	0.993	0.993	0.993
10	5	0.977	0.977	0.977	0.977
15	7	0.988	0.988	0.988	0.988
20	10	0.969	0.969	0.969	0.969
25	12	0.982	0.982	0.982	0.982

30	15	0.974	0.974	0.974	0.974

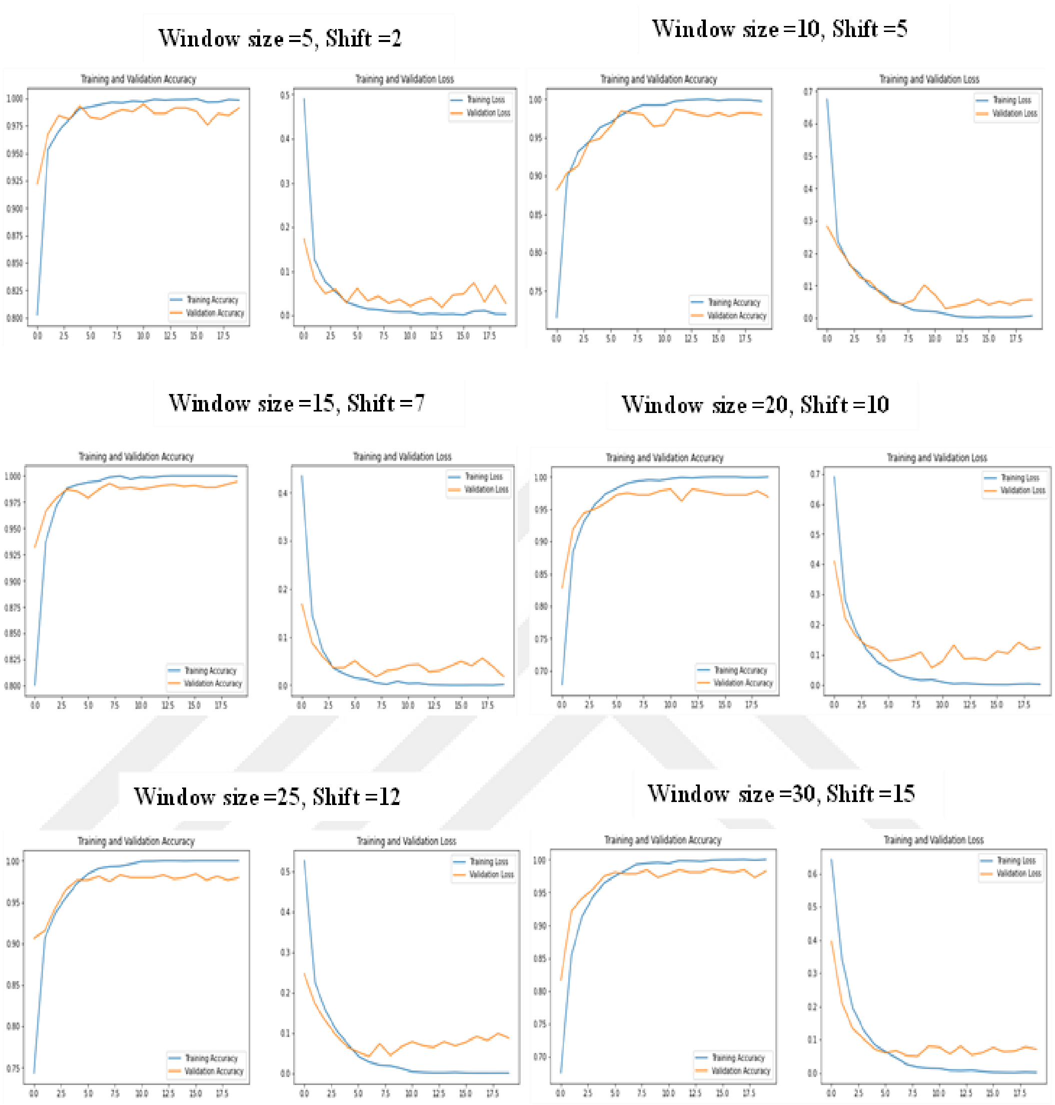


Figure 5.1. Illustrates how the CNN was performed during training, charting both accuracy and loss over the course of the training and validation epochs.

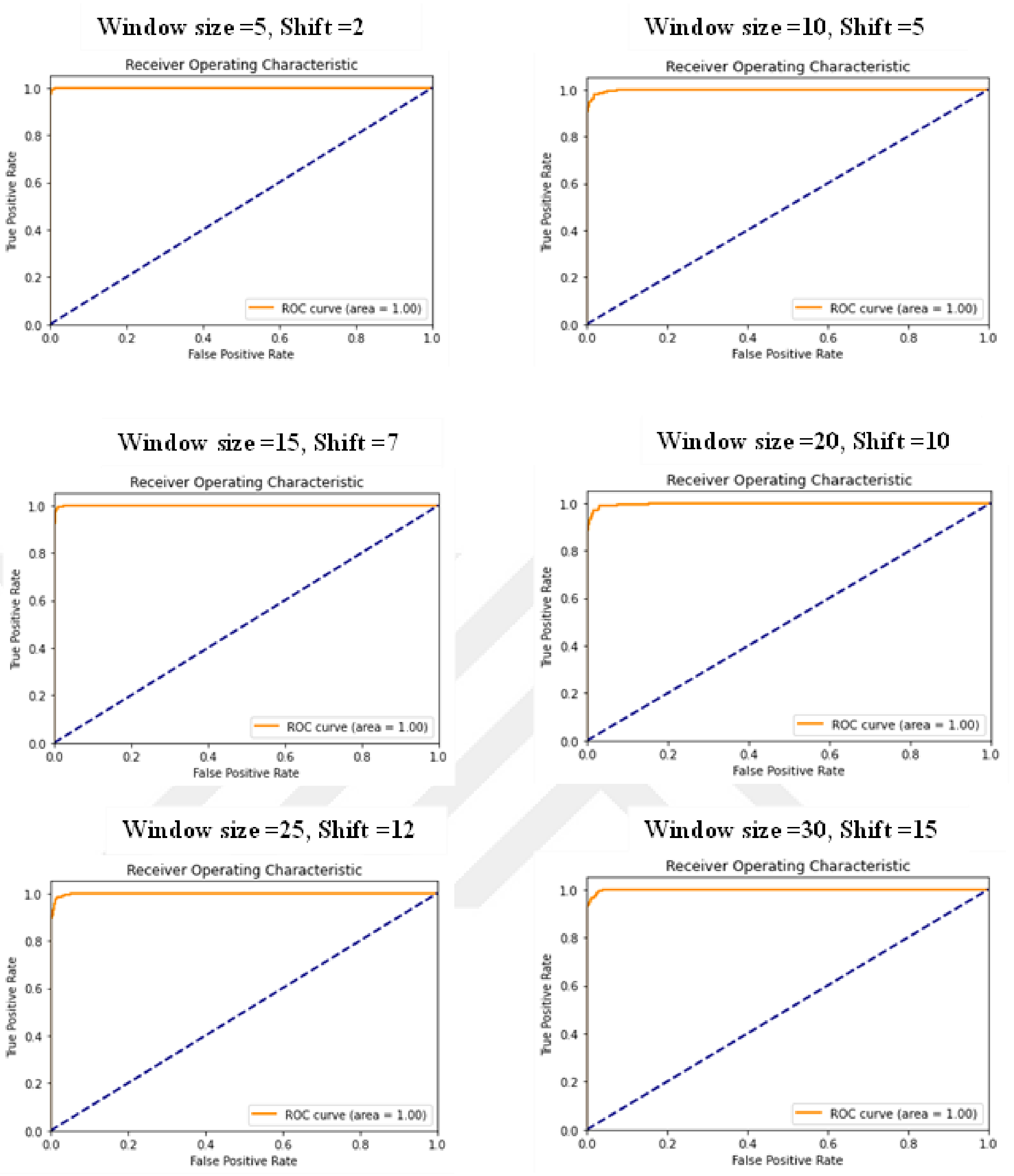


Figure 5.2. Presents the ROC curves indicating classification effectiveness. In almost every scenario, the ROC curves approached a perfect AUC of 1.00, suggesting an outstanding classifier that displayed minimal false positives.

Regarding the model performance, we further investigated the effect of changing window sizes and shifts by applying a cross-validation binary classification setup. We conducted experiments on six configurations of window size and shift in a time-series dataset, implementing stratified k-fold cross-validation (k = 5) to reduce the overestimation of outcomes. **Table 5.3.** provides a broad comparison of the model performance across these six configurations, each with distinct (W) window sizes and (S) Shift parameters.

Table 5.3. Performance evaluation of loss and accuracy across multiple configurations and cross-validation folds.

Fold	**Config 1** *(W=5, S=2)*		**Config 2** *(W=10, S=5)*		**Config 3** *(W=15, S=7)*		**Config 4** *(W=20, S=10)*		**Config 5** *(W=25, S=12)*		**Config 6** *(W=30, S=15)*	
	Loss	**ACC**	**Loss**	**ACC**	**Loss**	**ACC**	**Loss**	**ACC**	**Loss**	**ACC**	**Loss**	**ACC**
1	0.0293	0.9929	0.0284	0.9879	0.017	0.9933	0.1002	0.9633	0.0481	0.9816	0.0385	0.9875
2	0.0124	0.9944	0.0604	0.9829	0.1131	0.9816	0.077	0.9714	0.0536	0.9906	0.0658	0.9817
3	0.0374	0.9873	0.0461	0.9889	0.0428	0.9906	0.024	0.9932	0.0753	0.9924	0.0845	0.9788
4	0.0339	0.9913	0.0601	0.9789	0.026	0.9924	0.0642	0.9796	0.0466	0.9933	0.099	0.9624
5	0.014	0.9936	0.0659	0.9819	0.0256	0.9933	0.0503	0.985	0.0643	0.985	0.0636	0.9826
Average	**0.0254**	**0.9919**	**0.0522**	**0.9841**	**0.0449**	**0.9902**	**0.0631**	**0.9785**	**0.0576**	**0.9823**	**0.0703**	**0.9786**

Our findings showed different levels of effectiveness. Configuration 1 (W = 5, S = 2) showed outstanding metrics, with the best average accuracy (0.9919) and lowest mean loss (0.0254). It also showed remarkable stability across all five folds (accuracy ranged from 0.9873 to 0.9944). Examination of Configuration 1 revealed strong consistency, with fold 2 displaying the highest accuracy (0.9944) and lowest loss (0.0124). Meanwhile, Fold 3 posted a slightly lower accuracy (0.9873) and a marginally higher loss (0.0374). This pronounced steadiness in a variety of data subsets suggests excellent generalization and robust model behavior under different data conditions. In comparison, configuration 2 (W = 10, S = 5) experienced a slight decline in performance (mean accuracy ~ 0.9841, mean loss ~ 0.0522). Configuration 3 (W = 15, S = 7) also remained strong, with a mean accuracy of 0.9902 and mean loss of 0.0449. As the window size increased, the performance slowly decreased: configuration 4 (W = 20, S = 10) reached 0.9785 accuracy; configuration 5 (W = 25, S = 12) attained 0.9823 accuracy; and configuration 6 (W = 30, S = 15) finished at 0.9786 accuracy with a higher mean loss (0.0703). This pattern suggests that smaller windows and proportionally smaller shifts tend to capture the relevant temporal signals more effectively, thereby sustaining strong classification scores. Regardless, all six configurations managed to exceed 0.96 in accuracy, although larger windows also corresponded to higher loss and larger variations between folds. Configuration 1's low standard deviation for both accuracy and loss further demonstrated its superiority in this specific context.

Table 5.4. summarizes the performance of the model for each configuration. Configuration 1 (window = 5, shift = 2) again showed the highest accuracy (0.9919) and the lowest loss (0.0254). Configuration 3 (window size = 15, shift = 7) was runner-up (accuracy = 0.9902, loss = 0.0449). Meanwhile, window sizes of 20, 25, and 30 still achieved high accuracies (> 0.97), but with marginally higher loss values, culminating in Configuration 6's 0.0703 as the peak loss.

Table 5.4. The summary of average accuracy and loss for each configuration.

Configuration	Window size	Shift	Average loss	Average accuracy
Config 1	5	2	0.0254	0.9919
Config 2	10	5	0.0522	0.9841
Config 3	15	7	0.0449	0.9902
Config 4	20	10	0.0631	0.9785
Config 5	25	12	0.0576	0.9823
Config 6	30	15	0.0703	0.786

The graphs in Figure 5.3. illustrate the evolution of the training and validation performance metrics (accuracy and loss) over time for the CNN model, which was tested experimentally using the 5-fold cross-validation technique. Each graph corresponds to a specific training set-up or configuration.

To assess the classification performance of various CNN model configurations, Receiver Operating Characteristic (ROC) curves were plotted, as shown in Figure 5.4.. The analysis, conducted through 5-fold cross-validation on time-series data, indicated strong discriminative power across all configurations. Each configuration's ROC curve approached the ideal classification point (top-left corner), reflecting high sensitivity and specificity. Additionally, the performance metrics remain consistent across different windows and shift parameters, demonstrating a stable model behavior.

The configurations consistently achieved high true-positive rates while maintaining low false-positive rates, indicating robust binary-classification capabilities. These findings, visualized through ROC curves for six distinct configurations (with window sizes from 5 to 30 and corresponding shifts from 2 to 15), offer a thorough perspective of the model's classification efficacy across varying temporal scales.

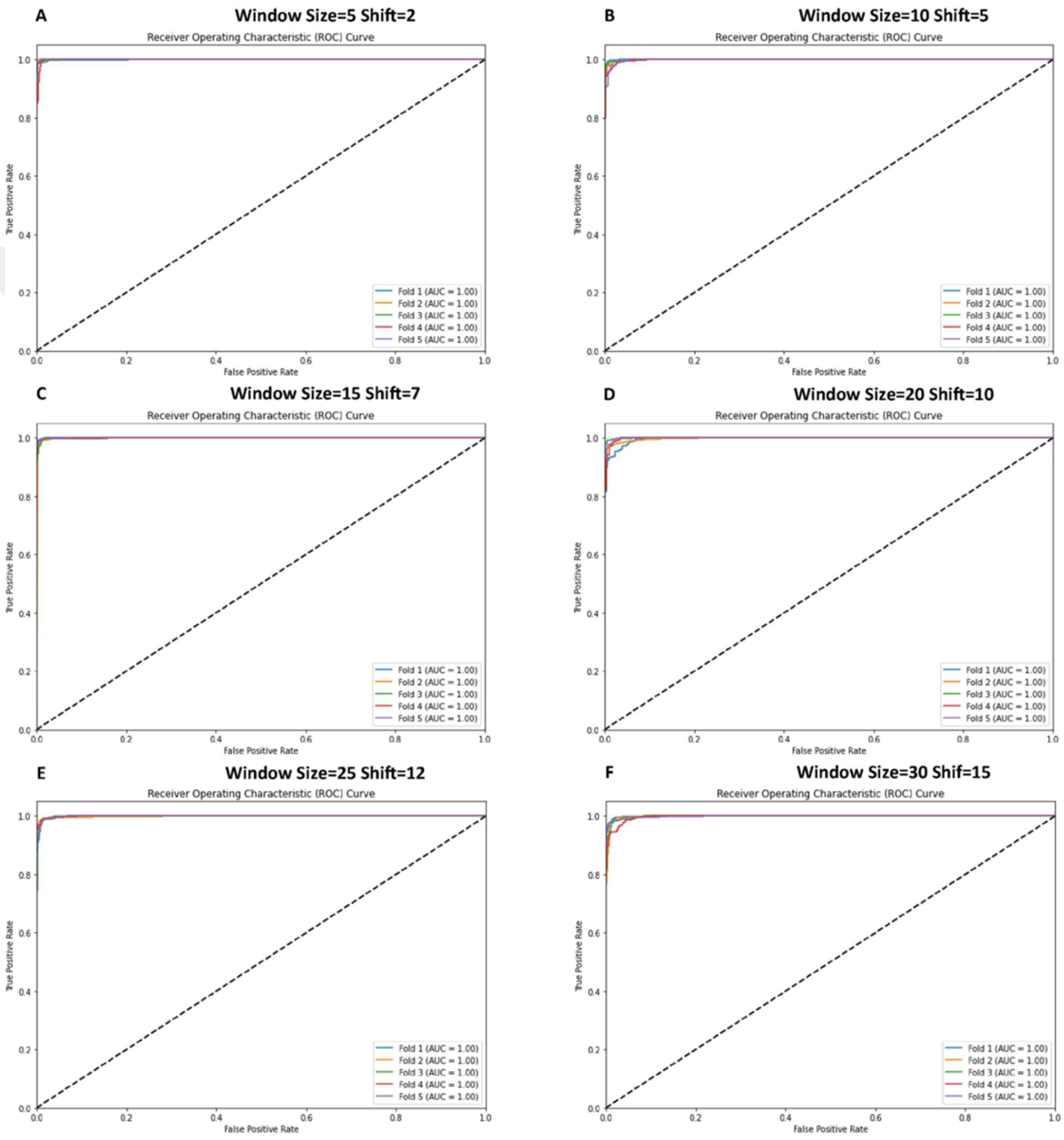


Figure 5.3. Shows the training and validation accuracy/loss metrics at different points during the experimental run with 5-fold cross-validation, with each panel indicating a particular configuration.

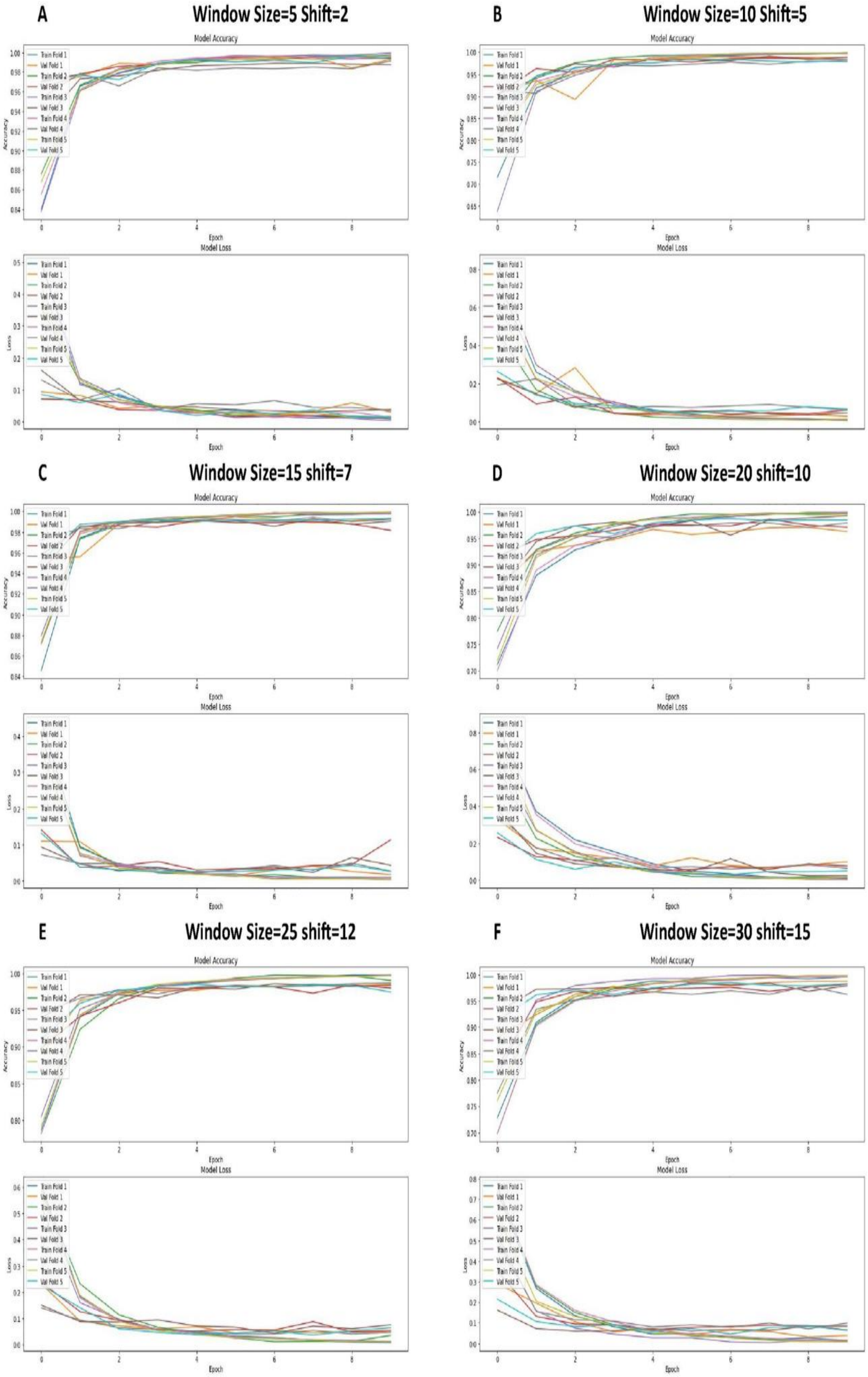


Figure 5.4. ROC curve graphs for various window and shift sizes after using the cross-validation.

Table 5.5. expands on the comparison by contrasting nine pre-trained architectures along the metrics of precision, recall, F1-score, and accuracy. Among these, MobileNet stands out, securing the best numbers across all measures (precision: 0.935; recall: 0.933; F1-score: 0.933; accuracy: 0.933). VGG19 also recorded good results (accuracy: 0.925), followed by VGG16 (accuracy: 0.925). EfficientNetB0 consistently exceeded 0.920 for its metrics, whereas ResNet50 and MobileNetV2 hovered around 0.907. DenseNet121 displayed a moderate performance (accuracy 0.812), and the inception-based models demonstrated significantly lower accuracies (~0.509 for InceptionResNetV2, ~0.586 for InceptionV3). Hence, MobileNet was the best among the tested pre-trained models, but the task-specific CNN described here outperformed these baselines overall.

Table 5.5. Performance metrics of pre-trained CNN models.

Model	Precision	Recall	F1-Score	Accuracy
VGG19	0.927	0.926	0.925	0.925
VGG16	0.925	0.925	0.925	0.925
ResNet50	0.912	0.908	0.907	0.907
MobileNet	0.935	0.933	0.933	0.933
EfficientNetB0	0.922	0.921	0.920	0.920
InceptionResNetV2	0.255	0.500	0.337	0.509
MobileNetV2	0.907	0.907	0.907	0.907
DenseNet121	0.863	0.809	0.804	0.812
InceptionV3	0.690	0.592	0.528	0.586

5.2. DISCUSSION

This section dives deeper into the candlestick chart forecasting approach and highlights its distinguishing features, computational complexity, and practical applicability. This discussion is divided into three subsections: **Comparative Analysis**, **Model Complexity Evaluation**, and **Real-Time Market Implementation**.

5.2.1. Comparative Analysis With Existing Literature

Comparing this candlestick chart forecasting methodology to the established literature reveals multiple key differences (see Table 5.6.). In this study, 61 candlestick patterns were used in tandem with a CNN model, diverging from conventional studies that rely on ensemble machine learning, other deep networks, graph-based models, or optimization algorithms. For example, Lin et al. (2021) employed a combination of random forest, GBDT, LR, KNN, SVM, and LSTM, whereas Chen and Tsai (2022) adopted YOLO for dynamic pattern recognition.

The main methodological distinction here is the targeted 15-minute Forex data interval, unlike other studies featuring daily or hourly data from places like China’s Stock Market (CSI 300, CSI 500), the Taiwan Stock Exchange, the Nikkei 225, or social media sources. Frequent sampling in our data yields more granular insights, in contrast to the daily/hourly intervals in Lin et al. (2021), Behar & Sharma (2022), and Wang et al. (2022), the 1-minute intervals from the Dynamic Deep Convolutional Candlestick Learner, or the 4-hour windows for SVM-based forex forecasting. Moreover, unlike many previous studies that explored multiple models (such as CNN-autoencoders, RNN, ensemble learners, or advanced optimization schemes), this study focuses exclusively on CNNs. Although other studies use an assortment of data types (historical prices, sentiment, or graph embeddings), ours centers specifically on OHLC candlestick information. Critically, the approach introduced here excels in accuracy (~99.3%), far surpassing the 56–91.51% range typically reported. By merging a systematic window and shift strategy with a single strong CNN model and a comprehensive analysis of 61 patterns, this study breaks the new ground in the literature.

Table 5.6. The comprehensive comparison of the related studies using candlestick charts.

Authors	**Stock market**	**Timeframe**	**Model used**	**Input dataset**	**Classification**	**Accuracy**
Lin et al. [52]	China’s	Daily	RF, GBDT, LR, KNN, SVM, LSTM	Historical stock prices, technical indicators	Categorical	0.6
Hung & Chen [18]	Taiwan and Tokyo	Daily	CNN-autoencoder, RNN	Candlestick Charts	Binary	0.84

Ho & Huang [53]	Apple, Tesla, IBM, Amazon, and Google	Daily	1-D CNN and 2-D CNN	Candlestick charts + Twitter Text	Binary	0.7538
Ardiyanti, Palupi & Indwiarti [54]	IDX	Daily	ANN+K-Fold Cross-Validation	Candlestick Pattern data	Binary	0.8596
Chen & Tsai [15]	Foreign exchange (EUR/USD)	1-min	YOLO model	GAF encoded candlestick charts	Binary	0.8835
Liang et al.[55]	China's	Daily	K-line	K-line patterns	Binary	56.04% and 55.56%
Santur [16]	11 world indices	Daily	Ensemble Learning-Xgboost	Candlestick Chart	Binary	0.538
Wang et al. [56]	China's (CSI 300)	Daily	Graph Neural Network	Candlestick is represented by graph embedding	Categorical	-
Behar & Sharma [57]	Indian(BSE and NIFTY 50) and US (S&P500 and DJIA)	Daily	KNN	Candlestick charts	Binary	0.614
Ramadhan, Palupi & Wahyudi [39]	Nasdaq100	Hourly	CNN-LSTM	GAF encoded candlestick charts	Binary	90% and 93%
Puteri et al. [58]	Forex (GBP/USD)	4-h	SVM	OHLC candlestick data	Binary	0.9072
Ruixun Zhang & Lin [59]	Exchange-traded funds (ETF)	Daily	Channel and Spatial-Attention CNN (CS-ACNN)	Candlestick charts	Binary	Sharpe ratios between 1.57 and 3.03
Vijayababu , Bennur & Vijayababu [60]	Ahihi Dataset	Daily	VGG16, ResNet50, AlexNet, GoogleNet, YOLOv8	OHLC candlestick pattern	Binary	0.9151
Chen, Hu & Xue [40]	Chinese	Daily	Bidirectional GRU with Candlestick Patterns and Sparrow Search Algorithm (SSA-CPBiGRU)	OHLC candlestick data	Categorical	-
Huang, Wang & Wang [61]	Chinese (Kweichow Moutai, CSI 100, and 50 ETF)	Daily	Vector auto-regression (VAR), Vector error correction model (VECM)	OHLC candlestick data	Binary	-

Proposed model	Forex (EUR/USD)	15-min	CNN	Candlestick charts	Binary	0.993

5.2.2. Model Complexity Evaluation

Table 5.7. provides a robust examination of the complexity of a CNN. The architecture has a total of 19,034,177 parameters and requires ~221 million Floating Point Operations (FLOPs), all of which consuming approximately 72.65 MB of memory. Composed of five layers, the architecture balances depth and computational demands. It includes three sequential Conv2D layers, each paired with a MaxPooling2D layer, followed by flattening and two dense layers, the latter separated by a dropout layer to curb overfitting. A closer look reveals that the second Conv2D layer has the highest FLOPs (95,883,264), whereas the largest param count (18,940,416) occurs in the first dense layer. Despite these significant demands, MaxPooling2D substantially reduces dimensionality while retaining vital features, boosting overall efficiency. Benchmarking with inference times underscores real-time feasibility, yielding a mean latency of ~570 ms and standard deviation of ~84 ms, which is reasonable in practical trading scenarios.

Table 5.7. CNN model complexity analysis.

Model overview			
Total Parameters:	19,034,177		
Total FLOPs:	221,008,385		
Parameter Memory:	72.61 MB		
Model Size:	72.65 MB		
Model Deth:	5 Layers		
Layer-wise Analysis			
Layer type	**Parameters**	**FLOPs**	**Output shape**
Conv2D	896	19,625,984	(64,148,148,32)
MaxPooling2D	0	700,928	(64,72,72,32)
Conv2D	18,496	95,883,264	(64,72,72,64)

MaxPooling2D	0	331,776	(64,36,36,64)
Conv2D	73,856	85,377,536	(64,34,34,128)
MaxPooling2D	0	147,968	(64,17,17,128)
Flatten	0	0	(64,36,992)
Dense	18,940,416	18,940,416	(64,512)
Dropout	0	0	(64,512)
Dense	513	513	(64,1)
Inference time analysis			
Mean Inference Time:		570.20 ms	
Std Inference Time:		84.14 ms	
Min Inference Time:		397.73 ms	
Max Inference Time:		741.64 ms	

5.2.3. Real-Time Market Analysis and Results

Figure 5.5. illustrates an algorithm’s performance in a real trading environment, specifically on the EUR/USD pair from October 28 (12:00) to November 1 (00:00), 2024. Markers “U” (Up) and “D” (Down) denote trend reversals. Throughout these 84 hours, prices oscillated between 1.078 and 1.088. A sharp drop occurred on October 29 (at approximately 12:00), bottoming near 1.078 before rebounding. The pair then embarked on an upward track, testing 1.086–1.088. These observations show that the trend detection algorithm works effectively in live, volatile conditions, thus carrying risk-management implications for institutional investors. The algorithm’s higher sensitivity to turning points compared to some standard technical tools suggests an advancement in AI-driven trading systems. Future investigations could extend these tests to other timespans and instruments to improve the adaptability of the model.

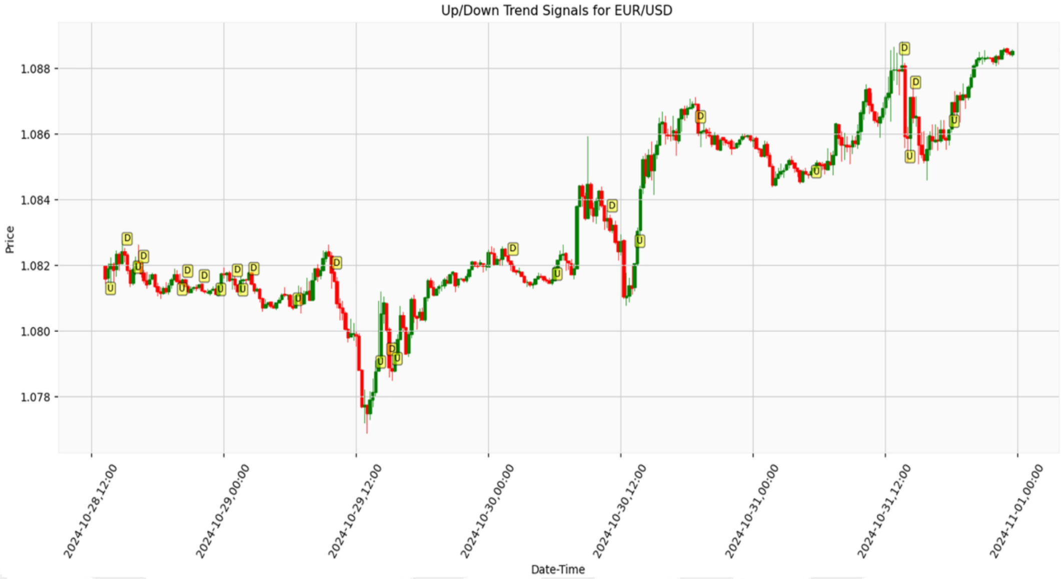


Figure 5.5. Algorithmic trend detection signals on EUR/USD parity over 84-h period with 15-min intervals (October 28–November 1, 2024).

PART 6

CONCLUSION AND FUTURE WORKS

6.1. CONCLUSION

This study integrates historical Japanese candlestick (JC) knowledge from the 17th century with a modern CNN, thereby enhancing the prediction of financial market behavior. The comprehensive three-stage approach we adopt establishes a novel path for market analysis, especially in the EUR/USD Forex domain. First, we introduced an innovative sliding window strategy that allows systematic time-series exploration, enabling the creation of detailed subgraphs and identifying 61 separate candlestick patterns. This step broadens conventional approaches and offers a more refined perspective on how markets evolve. Second, automated pattern detection via the Ta-lib library was combined with the selection of technical indicators and an adaptable window-shift mechanism (six intervals). Third, our sophisticated CNN architecture, comprising ~19 million parameters and ~221 million FLOPs, achieves remarkable efficiency, averaging an inference time of ~ 570 ms.

Our experiments confirm the efficacy of this combined approach, reaching ~99.3% accuracy, a figure substantially above that of existing methods that usually fall between 56% and 91.51%. This impressive rate was demonstrated through a rigorous 5-fold cross-validation on 15-minute data and validated in a real-time context (October 28–November 1, 2024). The model’s robust performance on novel data cements its suitability for scenarios demanding a rapid response, such as high-frequency trading, where classic technical analysis can falter amid rapid shifts.

The key contributions of this study include (1) a pattern-recognition framework encapsulating 61 candlestick formations, (2) an adaptive window-shift mechanism tailored to high-frequency data, and (3) quantitative trend detection using SMAs. These enhancements are significant for automated software trading and risk management methods.

6.2. FUTURE WORKS

Directions for future work include several aspects.

First, testing the model on different instruments (cryptocurrencies and commodities) and diverse markets (volatile or crisis periods) can foster broader applicability.

Second, further synergy with technical or fundamental signals could refine the model, especially if macroeconomic or sentiment-based data are integrated.

Third, advanced optimization (e.g., neural architecture search or pruning) can preserve or enhance the accuracy while reducing the computational overhead. **Fourth**, unsupervised or semi-supervised methods may uncover heretofore undocumented patterns, amplifying the model's insight.

Fifth, to strengthen real-world validation, performance evaluations should address more risk metrics, transaction costs, and additional market states.

Sixth, exploring longer time horizons or multiple time frames expands the scope of the model.

Lastly, advanced adaptive learning that recalibrates shifting market conditions might ensure sustained relevance over time.

In summary, this study merges legacy technical techniques with state-of-the-art AI to significantly improve the accuracy of financial forecasting. The resulting system holds promise for automated trading and real-time analytics, enhancing both practical outcomes and the broader financial modeling field. By merging historical expertise with modern deep learning, we set a stage for robust next-generation market-prediction frameworks.

REFERENCES

[1] Lin, Y., Liu, S., Yang, H., Wu, H., & Jiang, B. "Improving stock trading decisions based on pattern recognition using machine learning technology." ***PloS one*** 16.8: e0255558. (2021).

[2] Thammakesorn, S., & Sornil, O. "Generating trading strategies based on candlestick chart pattern characteristics." ***Journal of Physics: Conference Series***. Vol. 1195. No. 1. IOP Publishing, (2019).

[3] Udagawa, Y. "Mining stock price changes for profitable trade using candlestick chart patterns." ***Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services***. (2019).

[4] Jearanaitanakij, K., & Passaya, B. "Predicting short trend of stocks by using convolutional neural network and candlestick patterns." ***2019 4th International Conference on Information Technology (InCIT)***. IEEE, (2019).

[5] Pan, W., Li, J. ve Li, X. "Derin öğrenmeye dayalı portföy öğrenimi." ***Future Internet*** 12.11 202. (2020).

[6] Lin, Y., Liu, S., Yang, H., & Wu, H. "Stock trend prediction using candlestick charting and ensemble machine learning techniques with a novelty feature engineering scheme." *IEEE Access* 9: 101433-101446. (2021).

[7] Du, B., Fernandez-Reyes, D., & Barucca, P. "Image processing tools for financial time series classification." *arXiv preprint arXiv:2008.06042* (2020).

[8] Liang, M., Wu, S., Wang, X., & Chen, Q. "A stock time series forecasting approach incorporating candlestick patterns and sequence similarity." *Expert Systems with Applications* 205: 117595. (2022).

[9] Hung, C. C., Chen, Y. J., Guo, S. J., & Hsu, F. C. "Predicting the price movement from candlestick charts: a CNN-based approach." *International Journal of Ad Hoc and Ubiquitous Computing* 34.2: 111-120. (2020).

[10] Lin, Y., Liu, S., Yang, H., & Wu, H. "Stock trend prediction using candlestick charting and ensemble machine learning techniques with a novelty feature engineering scheme." *IEEE Access* 9: 101433-101446. (2021).

[11] Dakalbab, F., Talib, MA, Nasir, Q., & Saroufil, T.. "Finansal ticarette yapay zeka teknikleri: Sistematik bir literatür incelemesi." *Kral Suud Üniversitesi-Bilgisayar ve Bilişim Bilimleri Dergisi* 36.3 102015. (2024.

[12] Chen, J. H., & Tsai, Y. C. "Dynamic deep convolutional candlestick learner." *arXiv preprint arXiv:2201.08669* (2022).

[13] Mersal, E. R., & Kutucu, H. "TECHNIQUES USED TO EXTRACT FEATURES FROM CANDLESTICK CHARTS IN THE STOCK MARKET A SYSTEMATIC REVIEW." *Current Trends in Computing* 1.2 104-121. (2024).

[14] Nguyen, D. T., Tran, B. Q., Tran, A. D., Than, D. T., & Tran, D. Q. "Object Detection Approach for Stock Chart Patterns Recognition in Financial Markets." *Proceedings of the 2023 12th International Conference on Software and Computer Applications*. 2023.

[15] Chen, J. H., & Tsai, Y. C. "Dynamic deep convolutional candlestick learner." *arXiv preprint arXiv:2201.08669* (2022).

[16] Santur, Y. "Candlestick chart based trading system using ensemble learning for financial assets." *Sigma Journal of Engineering and Natural Sciences* 40.2 370-379. (2022).

[17] Brim, A., & Flann, N. S. "Deep reinforcement learning stock market trading, utilizing a CNN with candlestick images." *Plos one* 17.2 e0263181. (2022).

[18] Hung, C. C., & Chen, Y. J."DPP: Deep predictor for price movement from candlestick charts." *Plos one* 16.6 e0252404. (2021).

[19] Ramadhan, A., Palupi, I., & Wahyudi, B. A. "Candlestick patterns recognition using CNN-LSTM model to predict financial trading position in stock market." *Journal of Computer System and Informatics (JoSYC)* 3.4 339-47. (2022)

[20] Liang, Y., & Unwin, J. "Covid-19 forecasts via stock market indicators." *Scientific Reports* 12.1 13197. (2022).

[21] Orte, F., Mira, J., Sánchez, M. J., & Solana, P. "A random forest-based model for crypto asset forecasts in futures markets with out-of-sample prediction." *Research in International Business and Finance* 64 101829. (2023).

[22] Naik, N., & Mohan, B. R. "Intraday stock prediction based on deep neural network." *National Academy Science Letters* 43.3 241-246. (2020).

[23] Barra, S., Carta, S. M., Corriga, A., Podda, A. S., & Recupero, D. R. "Deep learning and time series-to-image encoding for financial forecasting." *IEEE/CAA Journal of Automatica Sinica* 7.3 683-692. (2020).

[24] Xu, R., Liu, X., Wan, H., Pan, X., & Li, J. "A feature extraction and classification method to forecast the PM2. 5 variation trend using candlestick and visual geometry group model." *Atmosphere* 12.5: 570. (2021).

[25] Chen, J. H., & Tsai, Y. C. "Dynamic deep convolutional candlestick learner." *arXiv preprint arXiv:2201.08669* (2022).

[26] Andriyanto, A., Wibowo, A., & Abidin, N. Z. "Sectoral stock prediction using convolutional neural networks with candlestick patterns as input images." *International Journal* 8.6: 2249-2252. (2020).

[27] Birogul, S., Temür, G., & Kose, U. "YOLO object recognition algorithm and “buy-sell decision” model over 2D candlestick charts." *IEEE access* 8 (2020): 91894-91915.

[28] Cagliero, L., Fior, J., & Garza, P. "Shortlisting machine learning-based stock trading recommendations using candlestick pattern recognition." *Expert Systems with Applications* 216 119493. (2023).

[29] Jun-Hao, C., & Yun-Cheng, T. "Encoding candlesticks as images for pattern classification using convolutional neural networks." *Financial Innovation* 6.1 (2020).

[30] Chen, X., Hu, W., & Xue, L. "Stock price prediction using candlestick patterns and sparrow search algorithm." *Electronics* 13.4 771. (2024).

[31] Heinz, A., Jamaloodeen, M., Saxena, A., & Pollacia, L. "Bullish and Bearish Engulfing Japanese Candlestick patterns: A statistical analysis on the S&P 500 index." The Quarterly Review of Economics and Finance 79 221-244. (2021)

[32] Karmelia, M. E., & Widjaja, M. "Candlestick Pattern Classification Using Feedforward Neural Network." *International Journal of Advances in Soft Computing & Its Applications* 14.2 (2022).

[33] Kusuma, R. M. I., Ho, T. T., Kao, W. C., Ou, Y. Y., & Hua, K. L. "Using deep learning neural networks and candlestick chart representation to predict stock market." *arXiv preprint arXiv:1903.12258* (2019).

[34] Lee, J., Kim, R., Koh, Y., & Kang, J. "Global stock market prediction based on stock chart images using deep Q-network." *IEEE Access* 7: 167260-167277. (2019).

[35] Madbouly, M. M., Elkholy, M., Gharib, Y. M., & Darwish, S. M. "Predicting stock market trends for Japanese candlestick using cloud model." *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)*. Springer International Publishing, 2020.

[36] Nakayama, A., Izumi, K., Sakaji, H., Matsushima, H., Shimada, T., & Yamada, K. "Short-term stock price prediction by analysis of order pattern images." *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. IEEE, 2019.

[37] Orquin-Serrano, I. "Predictive power of adaptive candlestick patterns in forex market. eurusd case." *Mathematics* 8.5: 802. (2020).

[38] Pan, W., Li, J., & Li, X. "Portfolio learning based on deep learning." *Future Internet* 12.11: 202. (2020).

[39] Ramadhan, A., Palupi, I., & Wahyudi, B. A. "Candlestick patterns recognition using CNN-LSTM model to predict financial trading position in stock market." *Journal of Computer System and Informatics (JoSYC)* 3.4: 339-47. (2022).

[40] Chen, X., Hu, W., & Xue, L. "Stock price prediction using candlestick patterns and sparrow search algorithm." *Electronics* 13.4: 771. (2024).

[41] Cohen, G. "Best candlesticks pattern to trade stocks." *International Journal of Economics and Financial Issues* 10.2: 256. (2020).

[42] Fengqian, D., & Chao, L. "An adaptive financial trading system using deep reinforcement learning with candlestick decomposing features." *IEEE Access* 8: 63666-63678. (2020).

[43] Wilson, G. T. "Time Series Analysis: Forecasting and Control, by George EP Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1.": 709-711. (2016).

[44] Hyndman, R. J., & Athanasopoulos, G. *Forecasting: principles and practice*. OTexts, 2018.

[45] Montgomery, D. C., Peck, E. A., & Vining, G. G. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[46] Guo, X., Li, W. J., & Qiao, J. F. "A self-organizing modular neural network based on empirical mode decomposition with sliding window for time series prediction." *Applied Soft Computing* 145: 110559. (2023)

[47] Chu, C. S. J. "Time series segmentation: A sliding window approach." *Information Sciences* 85.1-3: 147-173. (1995)

[48] Sadouk, L. "CNN Approaches for Time Series." *Time Series Analysis: Data, Methods, and Applications* : 57. (2019).

[49] Hota, H. S., Handa, R., & Shrivas, A. K. "Time series data prediction using sliding window based RBF neural network." *International Journal of Computational Intelligence Research* 13.5: 1145-1156. (2017).

[50] İnternet : TA-Lib Development Team, “Pattern Recognition Functions”. [Online]. Disponível. https://ta-lib.github.io/ta-lib-python/func_groups/pattern_recognition.html

[51] İnternet : https://chartschool.stockcharts.com/, “ChartSchool”.

RESUME

Edrees Ramadan MERSAL MORCELI completed his Bachelor's degree in Computer Science at Teachers College. Between 2008-2009, I worked as Control and Archiving Officer and Transportation Manager. Between 2009-2010 I worked as Site Manager and Supervisor. Between 2012-2014, I worked as a teacher at Manara School and Comprehensive Vocational High School. I continued his master's degree in Computer Engineering at Altınbaş University in Istanbul and completed it in 2018. In 2021, I started his PhD studies in Computer Engineering at Karabük University.