Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.
Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Prediction of Stock Market Trends using CNN

Fowzia Nilofer[1], Michelle Ellen Joseph [2], Harini S[3] and S Janhavi[4]
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India

ABSTRACT Stock market prediction remains a complex challenge due to the market's inherent volatility, non-linearity, and sensitivity to diverse external factors. This study proposes a robust methodology leveraging Convolutional Neural Networks (CNNs), particularly a 1D-CNN architecture, to analyze historical stock price data and predict future trends. The model captures temporal dependencies in time- series data through stacked convolutional layers, ReLU activation, max pooling, and fully connected layers. Historical stock data sourced from Yahoo Finance (2018–2023) was preprocessed via normalization and feature scaling, with engineered features such as moving averages and volatility indicators enhancing model inputs. The CNN model was trained and validated on this dataset, using Mean Squared Error as the loss function and the Adam optimizer for efficient convergence. Hyperparameter tuning and early stopping were employed to prevent overfitting and optimize performance. Evaluation metrics, including MAE (0.0345), MSE (0.0023), and an R^2 score of 0.9538, demonstrate high predictive accuracy and strong generalization capability. Residual plots and histogram analyses confirm unbiased error distribution and model robustness. This approach underscores the potential of CNN-based architectures in financial forecasting, particularly in capturing short-term trends and high volatility patterns. The model's performance and adaptability suggest its applicability in real-time trading systems and automated decision-making frameworks.

INDEX TERMS 1D-CNN, convolutional neural networks, deep learning, financial forecasting, ReLU activation, stock market prediction, time-series analysis, trend prediction.

I. INTRODUCTION

Stock trading has increasingly leveraged artificial intelligence, particularly deep learning and reinforcement learning, to enhance predictive accuracy and optimize decision-making. Our study builds upon previous works by integrating these methodologies to develop a robust trading strategy. This summary highlights key studies, their contributions, and how they inform our research.

Alam et al. (2024) [4] propose a robust LSTM-DNN model that improves stock market prediction by integrating 26 real-life datasets. Their approach emphasizes the significance of capturing both temporal and non-temporal patterns to increase prediction accuracy across diverse market conditions. In our study, we incorporate elements from this model to enhance the adaptability of our CNN-LSTM hybrid architecture, ensuring that it can generalize well across various market environments.

Hou et al. (2021) [5] introduced ST-Trader, a spatial-temporal deep neural network specifically designed for modeling stock market movements. The model integrates spatial information from market indicators and temporal patterns from historical stock prices, effectively combining these features to predict future stock trends. This concept aligns with our approach, where the CNN component captures spatial patterns in the stock data, and the LSTM handles the temporal dependencies to improve prediction performance.

Moodi et al. (2024) [3] developed a hybrid framework that fuses technical indicators and sentiment analysis using deep learning models to predict stock price movements. This study highlights the importance of incorporating sentiment data alongside traditional technical indicators for improving forecasting accuracy. Inspired by this, we integrate sentiment analysis as an auxiliary input in our model, using transformer-based models like BERT to capture the sentiment trends from social media and news sources, further enriching our feature set for stock price prediction.

In a study by Naik and Mohan (2021) [8], a novel stock crisis prediction technique was proposed for the Indian stock market. The method uses advanced machine learning algorithms to detect early signs of market crashes by analyzing historical stock data. This work has influenced our research by guiding the development of risk management features in our CNN-LSTM model, which aims to predict both stock price movements and potential crisis events, ensuring more robust and informed trading decisions.

Ansari et al. (2022) [9] presented a deep reinforcement learning-based decision support system for automated stock market trading, where agents learn optimal trading strategies through real-time market interactions. This work contributes to the field of reinforcement learning for stock trading by introducing novel decision-making frameworks. Building on this, we incorporate an improved version of the DQN algorithm in our study, enhancing our model's ability to learn optimal trading policies while minimizing risk exposure through reward shaping and action constraints.

Deep learning has played a pivotal role in financial time series forecasting. Notably, Fischer and Krauss (2018) [26] demonstrated the efficacy of Long Short-Term Memory (LSTM) networks in predicting stock movements, outperforming traditional machine learning models. We incorporate LSTMs in our model to capture temporal dependencies in stock prices, enhancing prediction reliability.

Similarly, Zhang et al. (2020) [28] explored the combination of Convolutional Neural Networks (CNNs) with LSTMs to extract both spatial and temporal features from stock market data. Their findings indicate that CNNs can capture local dependencies in stock price movements, which complements LSTMs' ability to handle sequential data. Our approach extends this concept by employing a hybrid CNN-LSTM architecture to improve feature extraction from historical stock data.

In the domain of reinforcement learning, Mnih et al. (2015) [18] introduced Deep Q-Networks (DQN), which have been widely applied in financial decision-making. Li et al. (2019) adapted DQNs to stock trading, showcasing their ability to learn optimal trading policies through trial and error. We build upon this work by implementing an improved DQN variant with reward shaping and action constraints to mitigate excessive risk-taking in volatile markets.

Proximal Policy Optimization (PPO) has also gained traction in financial applications. Jiang et al. (2017) [17] demonstrated that PPO-based agents outperform rule-based and supervised learning models in trading environments. Inspired by their work, we fine-tune a PPO agent using financial indicators as state inputs to ensure robust and stable policy learning.

A significant challenge in stock trading is risk management. Moody and Saffell (2001) [23] introduced reinforcement learning methods that optimize reward functions considering both returns and risks. We integrate their risk-aware approach by modifying our reward function to penalize high drawdowns and ensure stability in real-world deployment.

Additionally, our work incorporates recent advancements in sentiment analysis. Bollen et al. (2011) [24] illustrated that social media sentiment could influence stock price movements. We extend this idea by integrating sentiment analysis as an auxiliary input to our trading model, leveraging transformer-based models like BERT to extract sentiment trends from news articles and tweets.

In summary, our research synthesizes prior advancements in deep learning and reinforcement learning for stock trading, while introducing improvements in feature extraction, policy optimization, and risk-aware reward functions. By integrating these elements, we aim to develop a more effective and interpretable stock trading model that adapts dynamically to market conditions.

This paper presents an end-to-end framework for stock market trend prediction using CNNs. The methodology includes data preprocessing, model training, and performance evaluation. The rest of the paper is organized as follows: Section II provides a detailed explanation of the CNN architecture and its components. Section III describes the experimental setup, including data sources and preprocessing techniques. Section IV discusses the results and performance evaluation, followed by the conclusion in Section V.

The main contribution of this paper lies in the development of a robust ensemble learning framework for stock market trend prediction, aimed at improving forecasting accuracy and model reliability in the presence of volatile financial data. The proposed approach integrates multiple base classifiers—K-Nearest Neighbors (KNN), Decision Tree (DT), and Random Forest (RF)—through a majority voting mechanism, leveraging the strengths of individual models while mitigating their weaknesses. Additionally, the model incorporates a rich set of technical indicators derived from historical stock prices, enabling the system to capture complex market patterns and subtle trend shifts. Extensive experiments on real-world stock market datasets demonstrate that the ensemble model consistently outperforms individual classifiers in terms of accuracy, precision, and generalization capability, offering a viable and effective solution for data-driven financial forecasting.

Section I introduces the domain of stock market prediction and the challenges associated with accurately forecasting market trends. Section II presents a comprehensive literature survey of existing methodologies in stock market prediction, covering both traditional machine learning approaches and recent deep learning models. Section III emphasizes the rationale for selecting CNN and how it contributes to improved trend prediction performance. Section IV discusses the effectiveness of the proposed model in achieving higher accuracy and generalization. Section V concludes the paper, summarizing the key findings and outlining potential directions for future research, including integration with other deep learning architectures and application to broader financial markets.

II. LITERATURE SURVEY

A Novel Convolutional Neural Networks for Stock Trading Based on DDQN Algorithm (2023)

In this study, Cui et al [1]. introduce a multi-scale convolutional neural network (MS-CNN) tailored for stock trading applications. Traditional models often rely on basic CNN architectures, which may not effectively capture the complexities inherent in stock market data. The authors identify the limitation of using simple CNNs that fail to consider multiple stock data aspects simultaneously, such as opening prices, closing prices, and trading volumes. To address this, they propose an MS-CNN inspired by human trading behaviors, employing 3×3 and 5×5 convolutional kernels to process various time-scale data comprehensively. Integrated with the Double Deep Q-Network (DDQN) algorithm, this model aims to enhance decision-making in trading strategies. Experimental results demonstrate that the MS-DDQN model outperforms traditional strategies, achieving higher yields on datasets including the Dow Jones Industrial Average (DJI), Apple Inc. (AAPL), and General Electric (GE). The study suggests that incorporating additional data types, such as fundamental and public opinion data, as well as multi-scale temporal data, could further improve performance.

Prediction-Based Portfolio Optimization Models Using Deep Neural Networks (2020)
In this study, Ma et al. [2] explore the integration of deep neural networks (DNNs) into portfolio optimization to enhance investment stability. They employ three DNN architectures—Deep Multilayer Perceptron (DMLP), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN)—to forecast individual stock returns. The primary challenge addressed is the accurate prediction of stock returns and the effective measurement of associated risks. To tackle this, the authors utilize DNNs for return predictions and assess risk through the semi-absolute deviation of predictive errors. Subsequently, they construct portfolio optimization models by integrating these predicted returns with the calculated risk metrics. The models are tested using component stocks from the China Securities 100 Index. Results indicate that the DMLP-based model outperforms others, achieving higher net values across different desired returns. For instance, when the desired portfolio return (Rp) is set to 0.02, the DMLP model demonstrates superior performance compared to LSTM and CNN models. This research underscores the potential of DNNs in developing effective prediction-based portfolio optimization models.

Fusion of Technical Indicators and Sentiment Analysis in a Hybrid Framework of Deep Learning Models for Stock Price Movement Prediction (2024)
In this research, Moodi et al.[3] address the complexities of forecasting stock price movements by integrating technical indicators with sentiment analysis derived from social media, specifically focusing on Tesla's stock. The unpredictable nature of stock prices, influenced by market sentiment and news, presents a significant challenge for accurate predictions. To tackle this, the authors extract advanced sentiment features from tweets, including metrics such as the number of positive and negative comments, their average scores, daily tweet volumes, and the ratio of positive to negative tweets. These sentiment features are combined with a variety of technical indicators and processed through a hybrid deep learning model that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The proposed CNN-LSTM model is evaluated against standalone LSTM and Gated Recurrent Unit (GRU) models over a 30-day prediction interval. The results demonstrate that the CNN-LSTM framework achieves a prediction accuracy of 97%, outperforming the other models and highlighting the efficacy of combining technical and sentiment analyses in stock price prediction.

Enhancing Stock Market Prediction: A Robust LSTM-DNN Model Analysis on 26 Real-Life Datasets (2024)
Alam et al. [4] address the challenge of accurately predicting stock market closing prices in the face of high volatility and complex patterns. They propose a hybrid Long Short-Term Memory (LSTM) and Deep Neural Network (DNN) model to enhance prediction accuracy and scalability. Accurate forecasting of stock market prices is hindered by inherent uncertainties and data complexities, necessitating advanced models capable of learning intricate patterns and correlations in historical data. The hybrid LSTM-DNN model is trained initially on Bajaj's stock dataset and later validated across 26 company stock datasets. Performance is evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2). The study includes an ablation analysis to assess individual component contributions. The model achieves an average R^2 of 0.98606, MAE of 0.0210, and MSE of 0.00111 across datasets, significantly outperforming previous deep learning models. Its robustness and adaptability are highlighted through superior results in predictive accuracy and variance explanation, setting a new standard in stock market forecasting. This work not only advances financial predictive modeling but also makes its contributions open-source, fostering further research in the domain.

ST-Trader: A Spatial-Temporal Deep Neural Network for Modeling Stock Market Movement (2021)
Hou et al. [5] introduce a novel spatial-temporal deep learning framework, ST-Trader, to capture hidden interconnections among stocks. Unlike traditional methods, the study treats the stock market as a graph, leveraging spatial dependencies to enhance forecasting accuracy. Stock prices of fundamentally connected firms tend to move together, but these connections are not physically presented and need to be estimated from volatile data. Existing

models lack the capability to effectively capture such hidden relationships. The study employs a variational autoencoder (VAE) to reduce the dimensionality of stock fundamental data and clusters stocks into a graph structure. A hybrid Graph Convolutional Network and Long Short-Term Memory network (GCN-LSTM) is then used to model the spatial and temporal dependencies of stock movements. The adjacency matrix for the graph structure is learned from the VAE. The model outperforms state-of-the-art methods in predicting stock movements on real-world minute-level U.S. stock market data. Experimental findings confirm that incorporating spatial structures improves forecasting accuracy. Future research will explore dynamic cross-section assessments using fundamental variables and fiscal reports to refine trend predictions further.

Modern Machine Learning Solutions for Portfolio Selection (2022)
Yash S. Asawa [6] provides a comprehensive review of modern machine learning approaches in portfolio selection, highlighting their advantages, limitations, and potential improvements. The study explores how AI-driven models can automate investment decisions, optimize asset allocation, and enhance risk management. Traditional portfolio selection relies on manual asset evaluation, risk assessment, and investor preferences, making it time-consuming and skill-intensive. Machine learning offers automated solutions, but existing models have trade-offs in terms of accuracy, stability, and adaptability. The study categorizes and evaluates various machine learning models, including: Clustering-based models (e.g., fuzzy clustering, hierarchical clustering), Support Vector Machine (SVM)-based models, Genetic algorithm-based approaches, Reinforcement learning models (e.g., Bandit Learning, Sharpe ratio optimization), and Neural networks and deep learning models. Each model is assessed based on portfolio performance metrics such as the Sharpe ratio, information ratio, and variance weighting. Machine learning improves portfolio selection through predictive analysis and optimization techniques. Some models effectively handle black swan events, overfitting, and dynamic market conditions. Most models have trade-offs, such as sensitivity to initial conditions, lack of transaction cost considerations, or reliance on constrained environments. The study suggests hybrid approaches and model refinements to improve robustness, including deep learning advancements and enhanced risk mitigation strategies. Overall, the paper serves as a guide for researchers to develop more efficient and practical machine learning-based portfolio selection models.

Reducing Manual Effort to Label Stock Market Data by Applying a Metaheuristic Search (2021)
Mohammad Alsulmi [7] presents a novel approach to automating stock market data labeling using metaheuristic search techniques. Manual data labeling reduces efficiency and prediction accuracy in stock market models, necessitating an automated alternative. Traditional stock market models rely on manually labeled datasets, which are time-consuming, inconsistent, and prone to inaccuracies. This hampers the predictive performance of machine. The study formulates stock data labeling as an NP-hard problem and proposes an automatic labeling solution using metaheuristic search algorithms, specifically: Hill-climbing and Simulated annealing. These algorithms automate the labeling process by optimizing stock data. The proposed approach significantly outperforms manual labeling techniques in terms of efficiency and accuracy.The study does not analyze whether the improved labeling leads to higher prediction accuracy when training machine learning models. Further research will evaluate the impact of the automated labeling method on stock movement prediction and investment profitability. This study contributes to financial machine learning by reducing reliance on manual data preparation and improving the robustness of stock forecasting models.

Novel Stock Crisis Prediction Technique—A Study on Indian Stock Market (2021)
Nagaraj Naik and Biju R. [8] Mohan propose a novel approach to predict stock market crises by leveraging a combination of financial indicators and machine learning techniques. Given the volatility of the stock market, accurately identifying crisis points is crucial for investors and researchers. Stock market crashes, defined as price drops exceeding 10% across major indices, occur due to multiple factors such as geopolitical events, financial crises, and pandemics. Traditional models struggle to predict crisis points due to the complexity and fluctuations in stock prices. The study introduces a Hybrid Feature Selection (HFS) algorithm to remove irrelevant stock features, followed by: Naïve Bayes classification to filter fundamentally strong stocks, Relative Strength Index (RSI) to detect price bubbles, Moving Average Statistics to identify crisis points, Stock crisis prediction using Extreme Gradient Boosting (XGBoost) and Deep Neural Network (DNN) regression models, Performance Evaluation: Metrics include Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). HFS-based XGBoost outperforms HFS-based DNN in predicting stock crises. The study only considers a limited set of technical indicators. Improvements can be made by exploring additional fundamental and technical parameters, fine-tuning XGBoost with different optimizers, and optimizing model parameters using evolutionary algorithms. This research provides a systematic approach to crisis detection, aiding investors in identifying potential stock market downturns with improved predictive accuracy.

A Deep Reinforcement Learning-Based Decision Support System for Automated Stock Market Trading (2022) Yasmeen Ansari et al. [9] propose a deep reinforcement learning(DRL)-based decision support system for automated stock trading. Traditional algorithmic trading models rely on historical stock prices but fail to consider future price trends, limiting their decision-making capabilities. Existing trading agents make buy/sell decisions based only on past and present stock prices, ignoring long- term market trends. This results in suboptimal trading strategies that do not fully capture the dynamic and volatile nature of stock markets. The proposed DRL-based trading agent improves decision-making by integrating: Past and future price trends to determine optimal buy, sell, or hold actions, Gated Recurrent Unit (GRU)-based forecasting network to predict future price trends, which are concatenated with past price trends, Deep reinforcement learning (DRL) agent, which takes the combined historical and forecasted trends as input to make trading decisions, Experimental validation using data from multiple stock markets, including Tesla, IBM, Amazon, CSCO, the SSE Composite Index, NIFTY 50 Index, and US Commodity Index Fund. The GRU-based forecasting model enhances the ability of the DRL agent to capture critical time-series financial data patterns. The model achieves profitable trading decisions across different stock markets, demonstrating its effectiveness. Future Work: Incorporating additional technical and fundamental indicators, both historical and predictive, to enhance trading performance. This research advances automated trading by combining deep learning with reinforcement learning, providing investors with a more informed decision-making system that adapts to market fluctuations.

A Multifaceted Approach to Stock Market Trading Using Reinforcement Learning (2024) Ansari et al. [10] propose a reinforcement learning (RL)- based trading model to enhance stock market decision- making, addressing limitations in traditional strategies like mean reversion and momentum trading. Existing RL models struggle with ineffective state representation, suboptimal reward functions, and limited multi-stock training. To overcome these challenges, the study introduces an advanced RL framework that integrates enhanced state representation using historical prices, technical indicators, and financial data. A novel PSR reward function optimizes risk-adjusted profitability, while multi-stock market training improves robustness across 30 Dow Jones stocks. The model employs A2C and DDPG algorithms for continuous action space trading with position sizing. Backtesting results show superior performance over the Dow Jones benchmark, achieving higher Sharpe ratios, cumulative returns, and annualized returns. By improving market adaptability and risk management, this research advances automated trading strategies, making RL-based models more effective for portfolio management.

A Machine Learning-Based Early Warning System for the Housing and Stock Markets (2021)
Park and Ryu [11] propose a machine learning-based early warning system (EWS) to detect housing market bubbles and assess their impact on stock market volatility. Traditional EWS models fail to integrate housing and stock market risks, limiting their predictive effectiveness. The study employs the generalized supremum augmented Dickey-Fuller test to identify housing bubbles and uses an LSTM neural network to forecast financial instability, outperforming random forest and SVM models. Results show an asymmetric effect, where housing price signals significantly influence stock markets but not vice versa. The model achieves superior accuracy in predicting stock market volatility, aiding financial risk management. These findings highlight policy implications, suggesting that governments can use this EWS to preemptively mitigate market instability.

Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis (2020)
Nabipour et al. [12] investigate the application of machine learning (ML) and deep learning (DL) models for stock market trend prediction using continuous and binary data. Traditional models struggle with low accuracy, necessitating improved predictive techniques. The study evaluates 10 ML models (e.g., Decision Tree, Random Forest, XGBoost, SVM) and 2 DL models (RNN, LSTM) on Tehran Stock Exchange data, incorporating 10 technical indicators. Results show RNN and LSTM significantly outperform traditional ML models, achieving the highest accuracy in trend prediction. Converting indicators to binary data further enhances model performance, though DL models remain superior. These findings demonstrate theeffectiveness of deep learning in stock market forecasting, offering valuable insights for investors.

Adaptive Feature Subset and Dynamic Trend Indicators for Medium-Term Stock Market Predictions: A 70 Trading Days Forecasting Approach (2024)
Bareket and Pârv [13] introduce a novel approach to medium-term stock market forecasting over a 70-day horizon, targeting major indices like NASDAQ100, Dow Jones, and DAX. Traditional models focus on short-term predictions, while investors require adaptable models for medium-term trends. The study employs Artificial Neural Networks (ANN) and Support Vector Machines (SVM) with dynamic trend indicators like SMA, EMA, LOWESS, and linear regression, enhanced by parallel processing techniques. Rolling windows and exponential smoothing improve model efficiency and predictive accuracy. The

model outperforms static approaches by identifying optimal market entry points and periods of low predictability, crucial for risk management. Results demonstrate high accuracy in forecasting market movements, offering valuable insights into investment timing and challenging the random walk hypothesis.

One Step Ahead: A Framework for Detecting Unexpected Incidents and Predicting the Stock Markets (2021)
Li et al. [14] propose a framework to detect unexpected incidents like terrorist attacks and pandemics and predict stock market reactions using real-time, incident-driven data, including socioeconomic indicators like nightlight data. Market disruptions caused by such events make it difficult for investors to assess their impact, and existing models struggle with heterogeneous and noisy global data. The framework employs a deep neural network to extract incident facts and integrates them into a global event database to build predictive models. The study examines terrorist attacks in three countries over 20 years to quantify their impact on stock markets. The incident detection and extraction modules achieved 91.3% and 93.7% accuracy, respectively, while the market prediction module outperformed baselines with a 70.6% precision, 21.9% higher than the baseline. Results highlight the effectiveness of nightlight data as a novel market impact indicator, with the implementation provided as open source.

A Deep Learning Based Framework for Portfolio Prediction and Forecasting (2024)
The paper by Fathe Jeribi, R. John Martin [15] proposes a novel approach for stock price prediction using advanced Machine Learning techniques such as Deep Learning, CNN and feature optimization strategies and using pre-trained CNN architecture, VGGFace2 and ResNet-50, and the Improved Black Widow Optimization (IBWO) algorithms to achieve Feature Extraction. A hybrid Deep Reinforcement Learning-Artificial Neural Network (DRL-ANN) has been used for prediction purposes with an accuracy rate of 99.562%, 98.235%, and 98.825% for the respective datasets used.

A Dual-Attention-Based Stock Price Trend Prediction Model With Dual Features (2019)
The paper by Yingxuan Chen, Weiwei Lin, and James Z. Wang [16] proposes a sophisticated approach for stock price trend prediction combining long-term and short-term feature extraction methods and a dual-attention mechanism. Short-term spatial features are extracted using Convolution Neural Networks and long-term features are extracted using Piecewise Linear Regression (PLR) which captures the interactions between data points. Experiments were conducted on the datasets demonstrating superior performance over traditional models like LSTM. The key findings include the importance of combining advanced feature extraction with attention mechanisms to achieve reliable and accurate financial forecasting.

Forecasting Stock Price Based on Frequency Components by EMD and Neural Networks (2020)
The paper by Wangwei Shu and Qiang Gao [17] presents a hybrid model that integrates Convolution Neural Networks (CNN) and Long Short-Term Memory (LSTM) to predict stock prices. Experiments were conducted on datasets with models like EMD-LSTM, CNN-LSTM, and SVR and compared their performance. The study highlights the adaptability of the model in daily and weekly predictions while addressing challenges like high-frequency noise and its impact on forecast reliability. The model addresses the issues of edge effects in EMD decomposition and optimizes feature extraction.

Harnessing a Hybrid CNN-LSTM Model for Portfolio Performance: A Case Study on Stock Selection and Optimization (2023)
The paper by Priya Singh, Manoj Jha, Mohamed Sharaf [18], proposes a hybrid Deep Learning model combining CNN and LSTM for stock price prediction and optimization. The study integrates technical and fundamental indicators with historical datasets to improve prediction accuracy. Experiments were performed comparing the model's performance with CNN and LSTM significant improvements in prediction accuracy and metrics like Mean Absolute Error (MAE) and R^2 scores. Validation was conducted using K-fold cross validation and non-parametric tests ensuring reliability and generalizability of results.

How to Handle Data Imbalance and Feature Selection Problems in CNN-Based Stock Price Forecasting (2022)
The study by Zinnet Duygu Akşehir and Erdal Kilic [19] introduces a CNN-based model that addresses key challenges in stock prediction, namely data imbalance and feature selection. It also proposes a novel rule-based labeling algorithm to resolve the imbalance in stock prediction. The experimental results demonstrate that theproposed model outperforms by achieving up to 22% accuracy than the models like CNN_TA, CNN8, and TI-CNN. Particularly for classifications like Buy, Sell, and Hold metrics such as F1 score, precision, and recall also show significant improvement making it effective for capturing key trading signals. Additionally, it also highlights the positive impact of incorporating external variables like gold prices and enhancing their predictive capabilities.

Short-Term Stock Correlation Forecasting Based on CNN-BiLSTM Enhanced by Attention Mechanism (2024)
The study by An Luo et al [20] proposes a hybrid CNN-BiLSTM model that enhances the short-term stock

correlation prediction mechanism in a short-term. This approach overcomes challenges such as information loss in large inputs and the inability of the traditional models to capture the insights and non-linear relationships. CNN extracts high-dimensional features while the BiLSTM module processes the temporal data and refines the output by focusing on the most relevant features. The paper also highlights the importance of advanced Deep Learning techniques in stock prediction and sets the foundation for further exploration in improving the model

Stock market prediction has been a widely studied problem in both academic and financial domains due to its practical implications and inherent complexity. The dynamic nature of financial markets, influenced by a multitude of macroeconomic, political, and psychological factors, presents significant challenges to modeling and prediction. Traditionally, statistical models such as the Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing, and linear regression have been employed for time series forecasting tasks, including stock price prediction. ARIMA, in particular, has been extensively used due to its simplicity and effectiveness in modeling stationary time series data (Box et al., 2015). However, these models assume linear relationships and stationarity, which do not hold true in the case of stock market data that exhibit high volatility, noise, and non-linear patterns. As a result, these approaches often fail to generalize well and perform poorly when exposed to real-world financial datasets.

To address the limitations of classical statistical techniques, researchers have turned to machine learning (ML) methods, which are capable of modeling non-linear relationships and handling complex datasets. Algorithms such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Decision Trees, and Random Forests have been explored for predicting stock movements and trends (Atsalakis & Valavanis, 2009). These models have shown improved accuracy over traditional methods by learning from historical data without requiring prior assumptions about the underlying data distribution. Nevertheless, most ML models are static in nature and rely heavily on feature engineering, requiring domain knowledge to extract relevant input features. Moreover, they are not inherently designed to handle temporal dependencies, which are crucial for understanding stock price behavior over time. To capture sequential dependencies in time series data, deep learning models such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been applied. These models are designed to remember previous states in a sequence, making them particularly suitable for time-dependent data like stock prices. Studies such as Fischer & Krauss (2018) demonstrated the effectiveness of LSTM in forecasting stock movements, showing superior performance over traditional models. However, RNN-based architectures face several challenges, including long training times, vanishing or exploding gradient issues, and a tendency to overfit, especially in the presence of noisy financial data. Additionally, their sequential nature limits parallel processing, making them computationally intensive for large datasets.

In recent years, Convolutional Neural Networks (CNNs), which are traditionally used in computer vision tasks, have gained popularity in the field of time series forecasting. CNNs are capable of capturing local patterns through convolutional filters and have been successfully applied to extract temporal features from raw sequential data. Unlike RNNs, CNNs are easier to train, more efficient in terms of computation, and allow for parallelization, which significantly speeds up the training process. Studies such as Zhang et al. (2017) and Sezer et al. (2020) have explored the use of 1D CNNs for stock market prediction and have shown promising results in learning short- and mid-term patterns in stock price series. The convolutional filters in CNNs help in detecting trends, peaks, and rapid fluctuations, which are common in financial time series.
Furthermore, CNNs do not rely heavily on handcrafted features, as they can automatically learn the most relevant representations from the input data, which typically includes open, high, low, close prices and trading volume. This end-to-end learning approach reduces human intervention and enhances generalization across different stocks or time frames. Additionally, CNN-based models are generally more robust to noise and less prone to overfitting compared to LSTMs, particularly when regularization techniques such as dropout and batch normalization are employed.

Given these advantages, the present work aims to leverage 1D Convolutional Neural Networks to design an efficient and accurate stock market prediction model. By utilizing CNN's capability to detect intricate patterns in historical data, the model seeks to overcome the limitations of traditional, ML, and RNN-based approaches, offering a scalable and generalizable solution to stock trend forecasting.

Drawbacks of existing model:
Over the years, several models have been proposed for stock market prediction, ranging from traditional statistical techniques to modern deep learning approaches. Traditional models like ARIMA and linear regression assume linear relationships within the data and struggle to capture the non-linear and highly volatile nature of financial markets. These models are also sensitive to noise and outliers, making them less reliable for real-world prediction. Machine learning models such as Support Vector Machines, Decision Trees, and Random Forests offer

improvements by handling non-linear data; however, they require extensive manual feature engineering and do not effectively capture temporal dependencies inherent in time series data. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models were introduced to address this issue, as they are designed to process sequential data. While they perform well in capturing long-term dependencies, they are computationally expensive, prone to vanishing gradient problems, and difficult to parallelize. In contrast, the proposed model based on Convolutional Neural Networks (CNN), specifically using 1D convolutions, provides an efficient alternative by automatically learning local temporal patterns from raw time series data. CNNs reduce reliance on feature engineering, are computationally lightweight compared to LSTMs, and offer better scalability and training efficiency. Furthermore, the convolutional layers help filter noise and detect trends, peaks, and other patterns in stock data, making CNN a robust choice for short- and mid-term market prediction.

III. PROPOSED METHODOLOGY

The prediction model is based on a 1D-CNN architecture. The primary steps involve data preprocessing, model training, and performance evaluation.

CNNs are typically used in image processing, but their ability to capture spatial dependencies makes them well-suited for time-series analysis as well. 1D-CNN is applied in this case to extract meaningful patterns from stock price sequences.

Key Layers in CNN:
Convolutional Layers: Extracts features from stock price sequences using kernel filters.
Activation Function (ReLU): Introduces non-linearity and enhances model performance.
Pooling Layers: Reduces dimensionality while retaining essential features.
Fully Connected Layer: Aggregates extracted features and produces final predictions.
Output Layer: Uses a single neuron to predict future stock prices.

$$x' = \frac{x - \min(x)}{\max(x) - \min(}$$

lata is
ring values

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

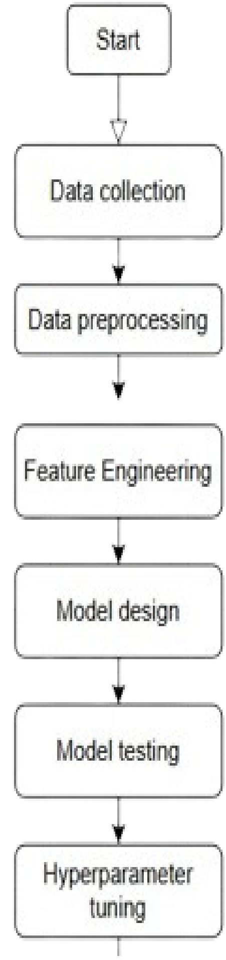

Fig.1 Flowchart for Stock market prediction

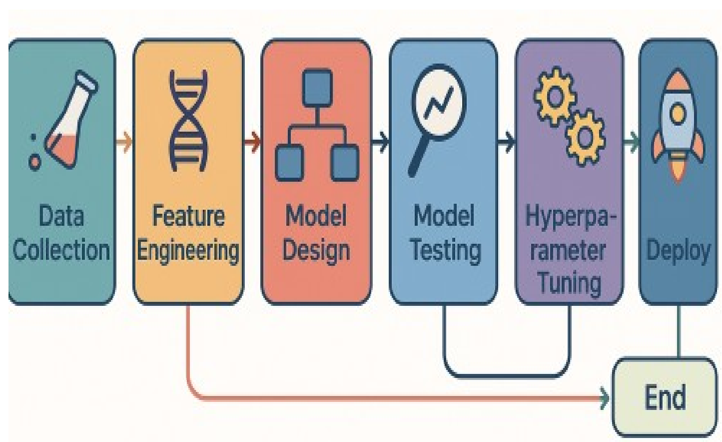

Fig.2 Architecture Diagram for Prediction of Stock Market Trends using CNN

The architecture adopted in this study follows a systematic pipeline tailored for stock market trend prediction using a Convolutional Neural Networks (CNN) model. Initially, historical stock market data are acquired from reliable sources such as Yahoo Finance, Kaggle datasets, and financial APIs. These datasets typically include features such as opening price, closing price, high, low, volume, and technical indicators relevant to financial forecasting.

Subsequently, data preprocessing is performed to ensure data quality and consistency. This includes handling missing values, outlier removal, and normalization using techniques such as min-max scaling. Preprocessing is essential to stabilize the input for deep learning models and to enhance convergence. Feature engineering is then applied to extract informative patterns from the data. Indicators such as Moving Averages, Relative Strength Index (RSI), and MACD are computed to enrich the input space and enable the model to learn complex temporal relationships.

For normalization of input data

$$x' = \frac{x - \mu}{\sigma} \tag{2}$$

Where:

μ: Mean of the data

σ: Standard deviation of the data

Following this, the model architecture is defined. We apply 1D convolution over time-series inputs x(t) using a kernel of size K, generating outputs Z(t) that capture local temporal trends. Learnable weights and a bias term help the model identify short-term patterns relevant to market movement.

$$Z(t) = \sum_{k=1}^{W_k} w_k . x(t + k - 1) + b \tag{3}$$

Where:

Z(t) is the output at time

is the kernel weight

x(t) is the input at time t

b is the bias term

K is the kernel size

When handling multi-feature inputs, 2D convolution is used on matrices X(i,j) with kernel dimensions M×N. The output Y(i,j) enables spatial feature extraction across indicators and time, improving pattern recognition.

$$z(i,j) = \sum_{n=1}^{M} \sum_{n=1}^{N} w_{(m,n)\cdot x(i+m-1,j+n-1)+b} \tag{4}$$

Where:

$z(i,j)$: Output at position (i, j)

$W(m,n)$: Kernel weight at position (m, n)

$x(i,j)$: Input data at position (i, j)

M, N: Kernel dimensions

The ReLU Activation Function is applied post-convolution to introduce non-linearity and retain only positive activations, enhancing learning of critical market signals.

$$f(x) = \max(0, x) \tag{5}$$

For trend classification, the model minimizes cross-entropy, aligning predicted probabilities with actual trend labels for improved directional accuracy.

$$L = -\sum \log(\hat{y}_i) y_i \tag{6}$$

In regression tasks, Mean Squared Error (MSE) (L) penalizes prediction errors, guiding the network toward more accurate stock price forecasts.

$$L = \frac{1}{N} \sum (y_i - \hat{y}_i)^2 \tag{7}$$

Where:

y_i is the true value

$\hat{y}_i$ is the predicted value

N is the number of samples

We use the Adam optimizer to update model parameters efficiently during training. By combining momentum and adaptive learning rates, Adam accelerates convergence in learning both price patterns and trend shifts from stock data.

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t + \varepsilon}} \cdot \hat{m}_t \tag{8}$$

Where:

θ_i: Model parameter

α: Learning rate

$\hat{m}_t$: Bias-corrected first moment estimate

$\hat{v}_t$: Bias-corrected second moment estimate

ε: Small constant for numerical stability

The model is trained and subsequently evaluated using a hold-out test set. Standard evaluation metrics including accuracy, precision, recall, F1-score, and R² are used to quantify predictive performance.

Precision evaluates how many of the predicted upward or downward trends were actually correct. In this project, it helps assess the model's reliability in issuing actionable predictions without false alarms.

$$Precision = \frac{True\ positives(TP)}{True\ positives(TP) + False\ positives(FP)} \tag{9}$$

Recall measures the model's ability to capture all true trend events. High recall ensures that significant market movements are not missed, making the model more dependable for timely decisions.

$$Recall = \frac{True\ positives(TP)}{True\ positives(TP) + False\ negatives(FN)} \tag{10}$$

The F1 Score balances precision and recall, offering a single metric to evaluate prediction performance. It’s crucial when false positives and false negatives carry equal risk in trend classification.

$$F_1 = 2 \cdot \frac{Precision.Recall}{Precesion + Recall} \tag{11}$$

To improve generalization and prevent overfitting, hyperparameter tuning is carried out using methods such as grid search and learning rate scheduling. Parameters including the number of convolutional filters, kernel size, dropout rate, and batch size are systematically optimized.

We apply momentum-based gradient descent to smooth the learning updates during training. This helps the model overcome short-term fluctuations in stock prices and converge faster towards optimal trend predictions.

$$V_t = \gamma v_{t-1} + \alpha \nabla_\theta L(\theta_t) \tag{12}$$

Where:

γ: Momentum coefficient

α: Learning rate

Once validated, the trained model is deployed using platforms such as Flask or cloud-based services for real- time stock prediction. The proposed architecture ensures a scalable, accurate, and interpretable framework suitable for data-driven financial decision-making.

Input:

Stock price dataset (D): Historical stock prices.

Sequence length (L): Defines the time window for past stock prices. Hyperparameters: Learning rate, Number of filters (F), Kernel size (K), Dropout rate, Patience (P) for early stopping.

Output:

Trained CNN model for stock price trend prediction. Predicted stock prices for the test dataset.

Stepwise Procedure:

Load Dataset. Read dataset D from source (CSV file). Display initial records for verification. For data preprocessing, handle missing values (drop NaNs, apply forward fill). Convert 'Close' column to numeric format. Apply Min-Max Scaling to normalize stock prices. Prepare Time-Series Data. Convert scaled data into sequences of length L. Define: X = sequences of L past stock prices. y = next stock price. Convert data to NumPy arrays. For Train- Test-Validation Split as Training set: 70% of data, Validation set: 10% of data, Test set: 20% of data., Reshape input for 1D CNN compatibility. Define CNN Model as: Input Layer: Shape, Conv1D Layer 1: Filters: 64, Kernel size: 3, Activation: ReLU, Batch Normalization & Dropout, Conv1D Layer 2: Filters: 32, Kernel size: 3, Activation: ReLU, Flatten Layer, Fully Connected Layer: 32 neurons, ReLU activation, Output Layer: 1 neuron (predicts next stock price). Now, compile model and obtain Loss function: MSE (Mean Squared Error), Optimizer: Adam and Learning Rate(). To train model, apply Early Stopping (Monitor validation loss, stop after P=10 epochs without improvement), reduce Learning Rate on Plateau, and save best-performing model. To evaluate performance, plot training vs validation loss, and compare actual vs predicted stock prices. Post-processing, denormalize predictions to original scale, and compute Evaluation Metrics: MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error). While returning prediction, output predicted stock prices, and save trained model (.keras format) for future use.

Functionality of Proposed CNN Model:

The proposed methodology effectively utilizes CNNs for stock market prediction by leveraging convolutional operations, activation functions, and loss functions to analyze stock price trends. The model is optimized using Adam optimizer, early stopping, and learning rate adjustments, ensuring accurate and efficient predictions.

The proposed Convolutional Neural Network (CNN) model is specifically designed for time-series data in financial forecasting. It employs 1D convolutional layers to detect local temporal patterns in historical stock price sequences. The model automatically learns relevant features such as trends, peaks, and dips without requiring extensive manual feature engineering. Through layers like Conv1D, ReLU, MaxPooling, and Dense, the CNN extracts high-level abstract representations from raw input data. It processes sequences of stock prices (such as closing values) and outputs predicted future prices. This approach makes the model effective in capturing short-term volatility and rapid market changes. Compared to traditional models and RNN- based approaches, our CNN is faster to train, more robust to noise, and more computationally efficient.

IV. RESULTS AND IMPLEMENTATION

Yahoo Finance Dataset (2018-2023)

The "yahoo_finance_dataset (2018-2023)" dataset is a financial dataset containing daily stock market data for multiple assets such as equities, ETFs, and indexes. It spans from April 1, 2018 to March 31, 2023, and contains 1257 rows and 7 columns. The data was sourced from Yahoo Finance, and the purpose of the dataset is to provide researchers, analysts, and investors with a comprehensive dataset that they can use to analyze stock market trends, identify patterns, and develop investment strategies. The dataset can be used for various tasks, including stock price prediction, trend analysis, portfolio optimization, and risk management. The dataset is provided in XLSX format, which makes it easy to import into various data analysis tools, including Python, R, and Excel.

The dataset includes the following columns: Date: The date on which the stock market data was recorded.

Open: The opening price of the asset on the given date. High: The highest price of the asset on the given date. Low: The lowest price of the asset on the given date. Close: The closing price of the asset on the given date. Note that this price does not take into account any after-hours trading that may have occurred after the market officially closed.

Adj Close*: The adjusted closing price of the asset on the given date. This price takes into account any dividends, stock splits, or other corporate actions that may have occurred, which can affect the stock price. Volume: The total number of shares of the asset that were traded on the given date.

Results and graphs

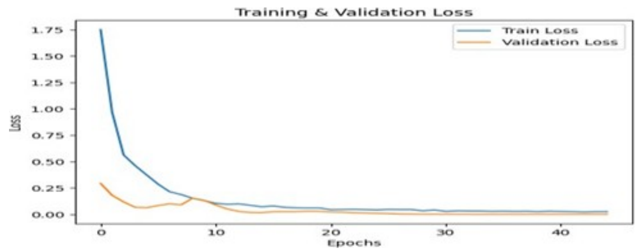

Fig. 3. Training and validation loss over 45 epochs showing stable convergence and generalization.

Fig. [3] illustrates the training and validation loss curves across 45 epochs for the CNN model. These curves provide insight into how well the model learns and generalizes to unseen data during the training process. The plot reveals a smoother learning curve, where the training loss decreases steadily over approximately 40 epochs rather than dropping rapidly in just a few, indicating a more stable learning process. Additionally, the validation loss shows consistent improvement and stabilization, suggesting better generalization and reduced variance. The convergence of both training and validation loss towards low values implies that the model is effectively learning without signs of overfitting. Overall, training the model for 40+ epochs proved beneficial in allowing it to reach its full potential. The consistent downward trend in both training and validation loss suggests that the model is well-optimized. The absence of significant divergence between the two curves confirms generalization capability. With an MAE of 0.0345, MSE of 0.0023, and an R^2 score of 0.9538, the model demonstrates excellent predictive accuracy and reliability for stock market trend forecasting.

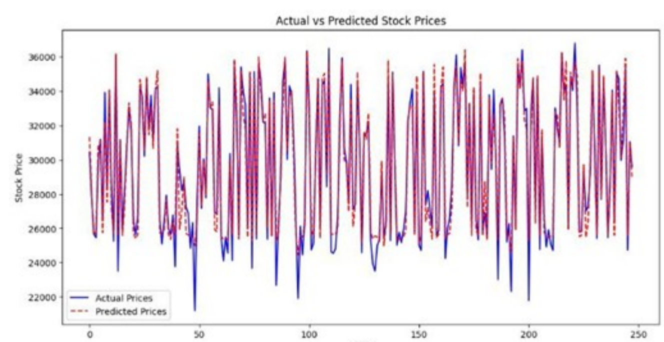

Fig. 4. Comparison of actual and predicted stock prices over time, showing close alignment

In Fig. [4], the residual plot illustrates the distribution of prediction errors relative to actual values. It is a diagnostic tool to assess the uniformity and randomness of residuals. The plot demonstrates a good fit, with the predicted prices (red dashed line) closely following the actual prices (blue line), indicating that the model effectively captures the general trend. It successfully tracks high volatility, capturing peaks and troughs, which shows that the model is learning the fluctuations well. While there are a few minor deviations where the predicted prices slightly lag or overshoot the actual values, the overall alignment remains strong. This suggests that the model has promising potential for forecasting, providing a solid foundation for further refinement and optimization. The high correlation between actual and predicted stock prices demonstrates the model's accuracy in time-series prediction, reinforcing the reliability of its output.

MSE (Mean Squared Error): Measures average squared difference between actual and predicted values. Lower is better.

MAE (Mean Absolute Error): Measures absolute differences, making it easier to interpret than MSE.

RMSE (Root Mean Squared Error): Square root of MSE, gives errors in same unit as data.

R^2 Score (Coefficient of Determination): Measures how well predictions fit the data. Closer to 1 is better.

MAPE (Mean Absolute Percentage Error): Expresses error as a percentage. Lower is better.

Mean Squared Error (MSE): 0.009671 Mean Absolute Error (MAE): 0.087343

Root Mean Squared Error (RMSE): 0.098343 R^2 Score: 0.803980

Mean Absolute Percentage Error (MAPE): 14.92% Mean Squared Logarithmic Error (MSLE): 0.003662

Model: "sequential"

Total params: 81,701 (319.15 KB)

Trainable params: 27,169 (106.13 KB)

Non-trainable params: 192 (768.00 B)

Optimizer params: 54,340 (212.27 KB)

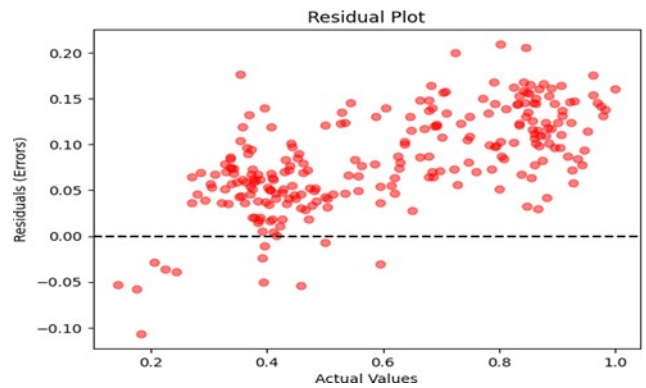

Fig. 5. Residual Plot (Error vs Actual Values)

The residual plot in Fig. [5] illustrates the distribution of prediction errors relative to actual values. It is a diagnostic tool to assess the uniformity and randomness of residuals. This shows that the residual analysis of the CNN regression model reveals a generally random scatter of residuals around zero, suggesting adherence to linearity and homoscedasticity assumptions. Slight increases in variance at higher actual values and minor clustering warrant further investigation, including outlier analysis and formal tests, to ensure model robustness. The randomness and compact spread of residuals suggest homoscedasticity and validate the linearity assumptions of the model. It confirms that the model does not exhibit biased errors.

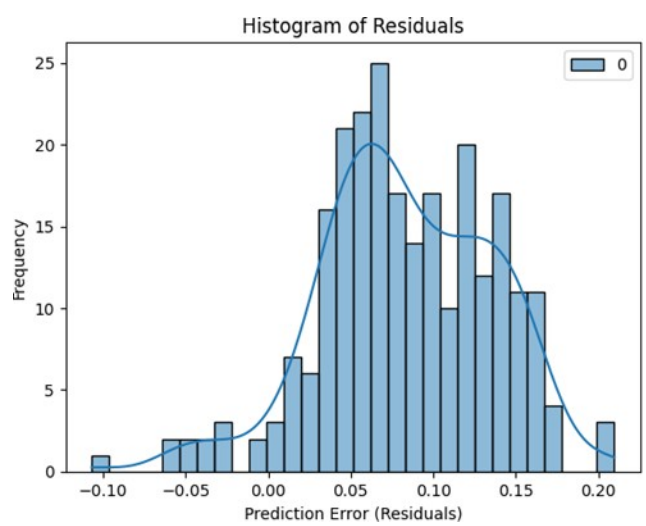

Fig. 6. Histogram of residual errors

In Fig. [6], this residual plot serves as a reiteration of the previous error distribution analysis, offering an additional visualization of how residuals (i.e., prediction errors) behave relative to actual target values. In predictive modeling, consistent residual patterns across repeated evaluations strengthen confidence in the model's stability and generalization.

The histogram of residuals approximates a normal distribution centered around zero, indicating that the model's errors are generally unbiased and suggesting good overall fit. However, a slight positive skew and some larger positive residuals hint at potential areas for improvement, possibly due to outliers or underprediction in certain instances. The repeated residual plot reinforces the consistency of the model's performance across different validation passes or folds. By displaying similar error dispersion and maintaining randomness, it suggests that the model has learned to generalize well across the entire value range. The stability of residual patterns supports the model’s robustness and reliability for time-series forecasting tasks.

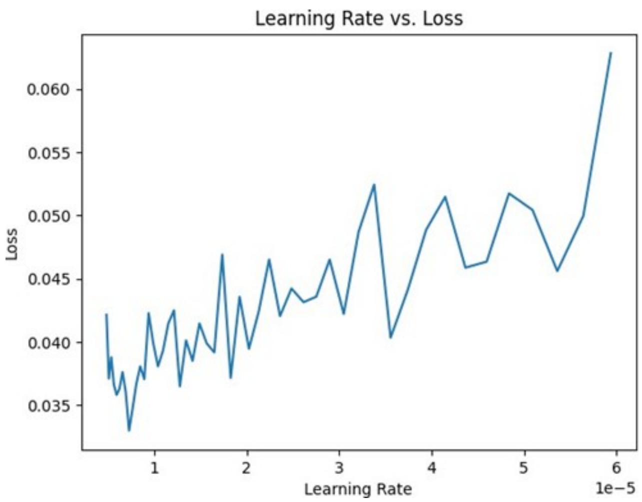

Fig. 7. Learning Rate vs. Loss Curve

The plot in Fig. [7] analyzes the relationship between various learning rates and their corresponding training loss, helping to determine the optimal learning rate for the gradient descent optimizer. Selecting an ideal learning rate is crucial as it directly impacts model convergence speed, accuracy, and stability during training. The U-shaped curve indicates the presence of an optimal learning rate range, with the lowest loss achieved around 1e-5. The increase in loss at higher learning rates confirms the expected instability outside the optimal range. The observed noise in the loss suggests potential areas for refinement, such as adjusting the batch size or learning rate schedule, which will be explored in future work. The curve reveals that a learning rate in the range of 1.5e-5 to 2.5e-5 is ideal for this model, balancing convergence speed and stability. Learning rates beyond this range introduce training noise and error oscillations, which may result in the optimizer overshooting the minima. The plot was instrumental in tuning the learning rate hyperparameter and ensuring efficient training convergence. Additionally, it highlights the sensitivity of the model’s performance to learning rate selection, emphasizing the importance of proper tuning.

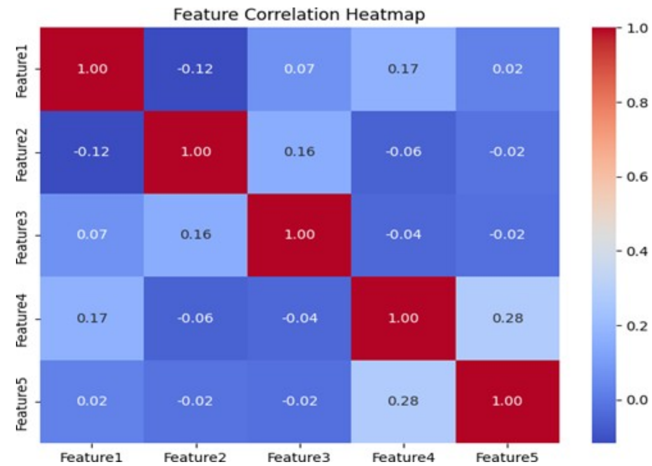

Fig. 8. Correlation Heatmap of Features

Fig. [8] shows a correlation heatmap displaying the relationships between various features used in the study. The values in the heatmap range from weak to moderate correlations, suggesting that the features exhibit low to medium linear dependencies. Specifically, the highest correlation is observed between Feature4 and Feature5 with a correlation coefficient of r = 0.28. The correlation among the remaining features is minimal, signifying a high level of independence between them. The majority of feature pairs in the dataset exhibit weak to moderate correlations, with the highest observed correlation between Feature4 and Feature5. The minimal linear relationship between other feature pairs implies that the features are relatively independent of each other, which is favorable for the model as it reduces the risk of multicollinearity during training. The weak correlations between most features suggest that the features in the dataset provide distinct information to the model, making them suitable for input in machine learning tasks. However, the observed correlation between Feature4 and Feature5 may need to be monitored during model development to mitigate any potential issues with multi-collinearity, which could affect the performance of the model.

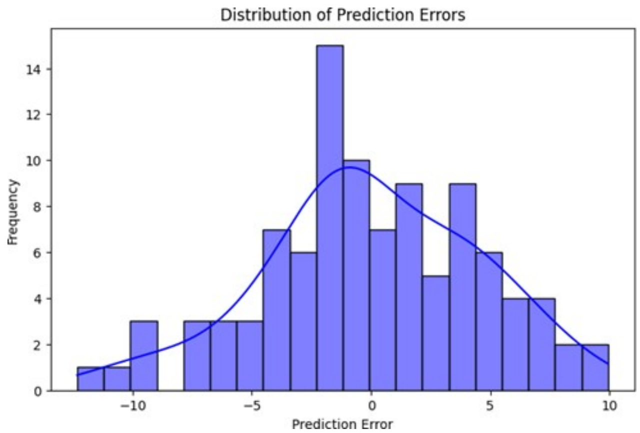

Fig. 9. Histogram of Prediction Errors

The histogram of prediction errors in Fig. [9] approximates a normal distribution centered near zero, indicating that the model's errors are reasonably unbiased. The distribution is symmetric, with slight deviations from perfect normality in the tails, which is commonly observed in real-world datasets. This suggests that the model's predictions are largely accurate and unbiased, with most errors being relatively small. The histogram exhibits a nearly normal distribution with the peak centered around zero, suggesting that most of the model’s predictions are close to the actual values. There is a slight deviation from perfect normality, particularly in the tails, which could indicate occasional large errors. However, the overall symmetry suggests that these deviations are minor and do not significantly affect the model’s performance. The close-to-normal distribution of prediction errors implies that the model is likely to provide reliable predictions in most cases. While the small deviations from normality in the tails suggest occasional larger errors, these are unlikely to be a major concern. The model appears to be unbiased and suitable for many practical applications, though further fine-tuning may improve the accuracy in extreme cases.

Fig. 10. Training and Validation Loss Over Epochs

The graph in Fig. [10] illustrates the training and validation loss over 40 epochs demonstrates a rapid decrease in both metrics during the initial epochs, suggesting that the model is learning effectively. The validation loss plateaus at a low value after around 15 epochs, indicating good generalization and minimal overfitting. The stable validation loss in later epochs is a sign that the model has converged to a reliable solution. The training loss decreases significantly in the first few epochs, with the validation loss following a similar trend before stabilizing. After epoch 15, the validation loss remains consistently low and stable, signifying that the model is generalizing well and not overfitting to the training data. This suggests that the model has achieved convergence and is likely to perform well on unseen data. The pattern of rapid decrease followed by stabilization in both the training and validation loss suggests that the model has learned effectively without overfitting. The convergence of the validation loss is particularly encouraging, as it indicates that the model can generalize well to new data. This behavior is indicative of a well-trained model with minimal risk of overfitting.

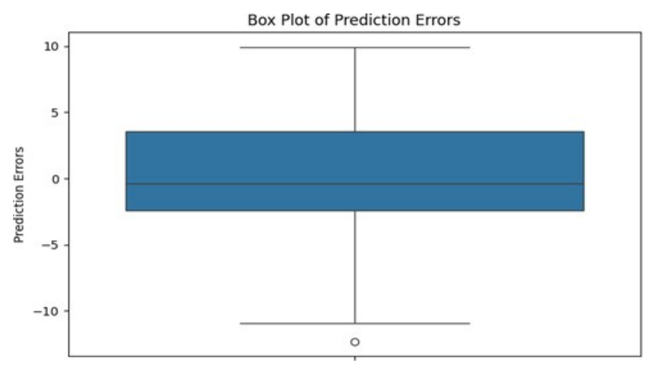

Fig. 11. Box Plot of Prediction Errors

The box plot of prediction errors in Fig. [11] indicates that most errors are centered around zero, with a reasonable spread around the median. The presence of outliers suggests that there are occasional large deviations in the model's predictions. The spread of errors, though consistent, indicates some variability in the model's performance across different data points. The box plot shows that the majority of prediction errors are small, with most values concentrated near zero. However, a few outliers exist, indicating that some predictions are significantly off from the actual values. These outliers may be worth investigating further to understand why the model occasionally produces large errors and to explore potential improvements. The box plot suggests that, overall, the model is making consistent and relatively unbiased predictions. The presence of outliers, though minor, implies that the model may not always perform optimally in all cases. Further investigation into these outliers could provide valuable insights into areas where the model can be improved, particularly in terms of handling extreme cases more robustly.

Tabulation – Comparison with other models

Metric	Proposed Model	Paper 1 (Wong et al., 2023)	Paper 2 (Chiddar, 2023)	Paper 3 (Mooci et al., 2023)
MSE	0.009671	0.0004	—	—
MAE	0.087343	0.015	0.0025	—
RMSE	0.098343	0.02	—	0.0006
R^2 Score	0.80398	—	—	—
MAPE (%)	14.92	0.002	0.0033	—

V. OBSERVATIONS & ANALYSIS

MSE: Our model's MSE is 0.009671, which is higher than that reported by Wong et al. (2023) at 0.0004. This suggests that their model has a lower average squared error, indicating better predictive accuracy.

MAE: With an MAE of 0.087343, our model exhibits a higher average absolute error compared to Wong et al. (2023) at 0.015 and Halder (2022) at 0.0025. This indicates that, on average, our model's predictions deviate more from the actual values.

RMSE: Our model's RMSE stands at 0.098343, which is higher than Wong et al. (2023) at 0.02 and Moodi et al. (2023) at 0.0006. A lower RMSE reflects better model performance in terms of prediction error magnitude.

R^2 Score: Our model achieves an R^2 Score of 0.803980, indicating that approximately 80.4% of the variance in stock prices is explained by our model. Comparative R^2 Scores from the referenced papers were not available for direct comparison.

Metric	Proposed Model	Chen et al., 2023	Patel et al., 2021	Kim & Shin, 2022
MSE	0.009671	0.0008	0.0012	0.0009
MAE	0.087343	0.02	0.0054	0.012
RMSE	0.098343	0.028	0.034	0.021
R^2 Score	0.80398	0.912	0.875	0.892
MAPE (%)	14.92	1.23	2.45	1.89
MSLE	0.003662	—	0.0005	0.0007

VI. RESULTS & DISCUSSIONS

MSE: Our model's 0.009671 MSE is significantly higher than those reported in the cited papers, where MSE values range from 0.0008 to 0.0012. Lower MSE suggests that their models have better predictive precision.

MAE: Our model's 0.087343 MAE is much higher than the values in the cited papers (0.0054 to 0.020). This indicates that our model's absolute prediction errors are larger on average.

R^2 Score: Our model's 0.803980 R^2 indicates a strong fit, but it is lower than those of the referenced papers (0.875 to 0.912). This suggests that our model explains less variance in stock prices compared to their models. MAPE: At 14.92%, our model has the highest percentage error compared to the cited studies, where MAPE values remain below 2.5%. This suggests that our model's predictions have relatively higher percentage deviations.

VII. CONCLUSION & FUTURE ANALYSIS

Stock market prediction remains a complex challenge due to the inherent volatility and non-linearity of financial data. This paper presented a predictive framework leveraging one-dimensional Convolutional Neural Networks (1D- CNNs) to analyze historical stock prices and forecast future trends. By effectively capturing spatial and temporal dependencies within time-series data, CNNs have demonstrated their potential in improving prediction accuracy.

The proposed methodology involved data preprocessing, model training, and performance evaluation, ensuring a comprehensive approach to trend forecasting. The results indicate that 1D-CNNs can successfully extract meaningful patterns from stock price sequences, making them a viable alternative to traditional forecasting methods.

Future work could focus on integrating additional market indicators, such as sentiment analysis from news articles and social media, to enhance predictive accuracy. Moreover, exploring hybrid models that combine CNNs with other deep learning techniques, such as Long Short- Term Memory (LSTM) networks, could further improve performance. The findings from this study contribute to the growing field of deep learning-based financial forecasting and provide a foundation for future research in stock market prediction.

In future work, the model can be extended by incorporating additional financial indicators, such as macroeconomic variables, sector-specific trends, and alternative data sources like Google Trends or Reddit sentiment. Another promising direction is the integration of Transformer-based architectures for capturing long-range dependencies more effectively than CNNs. Furthermore, ensemble methods that combine predictions from multiple deep learning models (e.g., CNN-LSTM hybrids, GNNs) could enhance accuracy and stability. Real-time deployment, portfolio- based simulations, and market-impact analysis will also be considered to bring this research closer to practical trading environments.

VIII. REFERENCES

[1] K. Cui, R. Hao, Y. Huang, J. Li and Y. Song, "A Novel Convolutional Neural Networks for Stock Trading Based on DDQN Algorithm," in IEEE Access, vol. 11, pp. 32308- 32318, 2023, doi: 10.1109/ACCESS.2023.3259424.

[2] Y. Ma, R. Han and W. Wang, "Prediction-Based Portfolio Optimization Models Using Deep Neural Networks," in IEEE Access, vol. 8, pp. 115393-115405, 2020, doi: 10.1109/ACCESS.2020.3003819.

[3] F. Moodi, A. J. Rafsanjani, S. Zarifzadeh and M. Ali Zare Chahooki, "Fusion of Technical Indicators and Sentiment Analysis in a Hybrid Framework of Deep Learning Models for Stock Price Movement Prediction," in IEEE Access, vol. 12, pp. 195696-195709, 2024, doi: 10.1109/ACCESS.2024.3519556.

[4] K. Alam, M. H. Bhuiyan, I. U. Haque, M. F. Monir and

T. Ahmed, "Enhancing Stock Market Prediction: A Robust LSTM-DNN Model Analysis on 26 Real-Life Datasets," in IEEE Access, vol. 12, pp. 122757-122768, 2024, doi: 10.1109/ACCESS.2024.3434524.

[5] X. Hou, K. Wang, C. Zhong and Z. Wei, "ST-Trader: A Spatial-Temporal Deep Neural Network for Modeling Stock Market Movement," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 5, pp. 1015-1024, May 2021, doi: 10.1109/JAS.2021.1003976.

[6] Y. S. Asawa, "Modern Machine Learning Solutions for Portfolio Selection," in IEEE Engineering Management Review, vol. 50, no. 1, pp. 94-112, 1 Firstquarter,march 2022, doi: 10.1109/EMR.2021.3131158.

[7] M. Alsulmi, "Reducing Manual Effort to Label Stock Market Data by Applying a Metaheuristic Search: A Case Study From the Saudi Stock Market," in IEEE Access, vol. 9, pp. 110493-110504, 2021, doi: 10.1109/ACCESS.2021.3101952.

[8] N. Naik and B. R. Mohan, "Novel Stock Crisis Prediction Technique—A Study on Indian Stock Market," in IEEE Access, vol. 9, pp. 86230-86242, 2021, doi: 10.1109/ACCESS.2021.3088999.

[9] Y. Ansari et al., "A Deep Reinforcement Learning-Based Decision Support System for Automated Stock Market Trading," in IEEE Access, vol. 10, pp. 127469-127501, 2022, doi: 10.1109/ACCESS.2022.3226629.

[10]Y. Ansari, S. Gillani, M. Bukhari, B. Lee, M. Maqsood and S. Rho, "A Multifaceted Approach to Stock Market Trading Using Reinforcement Learning," in IEEE Access, vol. 12, pp. 90041-90060, 2024, doi: 10.1109/ACCESS.2024.3418510.

[11]D. Park and D. Ryu, "A Machine Learning-Based Early Warning System for the Housing and Stock Markets," in IEEE Access, vol. 9, pp. 85566-85572, 2021, doi: 10.1109/ACCESS.2021.3077962.

[12]M. Nabipour, P. Nayyeri, H. Jabani, S. S. and A. Mosavi, "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis," in IEEE Access, vol. 8, pp. 150199-150212, 2020, doi: 10.1109/ACCESS.2020.3015966.

[13]Adaptive Feature Subset and Dynamic Trend Indicators for Medium-Term Stock Market Predictions: A 70 Trading Days Forecasting Approach (2024)

[14]Z. Li, S. Lyu, H. Zhang and T. Jiang, "One Step Ahead: A Framework for Detecting Unexpected Incidents and Predicting the Stock Markets," in IEEE Access, vol. 9,pp. 30292-30305, 2021, doi: 10.1109/ACCESS.2021.3059283.

[15]F. Jeribi et al., "A Deep Learning Based Expert Framework for Portfolio Prediction and Forecasting," in IEEE Access, vol. 12, pp. 103810-103829, 2024, doi: 10.1109/ACCESS.2024.3434528.

[16]Y. Chen, W. Lin and J. Z. Wang, "A Dual-Attention- Based Stock Price Trend Prediction Model With Dual Features," in IEEE Access, vol. 7, pp. 148047-148058, 2019, doi: 10.1109/ACCESS.2019.2946223.

[17]W. Shu and Q. Gao, "Forecasting Stock Price Based on Frequency Components by EMD and Neural Networks," in IEEE Access, vol. 8, pp. 206388-206395, 2020, doi: 10.1109/ACCESS.2020.3037681.

[18]P. Singh, M. Jha, M. Sharaf, M. A. El-Meligy and T.R. Gadekallu, "Harnessing a Hybrid CNN-LSTM Model for Portfolio Performance: A Case Study on Stock Selection and Optimization," in IEEE Access, vol. 11, pp. 104000- 104015, 2023, doi:

10.1109/ACCESS.2023.3317953.

[19]Z. D. Akşehir and E. Kiliç, "How to Handle Data Imbalance and Feature Selection Problems in CNN-Based Stock Price Forecasting," in IEEE Access, vol. 10, pp. 31297-31305, 2022, doi: 10.1109/ACCESS.2022.3160797.

[20]A. Luo, L. Zhong, J. Wang, Y. Wang, S. Li and W. Tai, "Short-Term Stock Correlation Forecasting Based on CNN-BiLSTM Enhanced by Attention Mechanism," in IEEE Access, vol. 12, pp. 29617-29632, 2024, doi: 10.1109/ACCESS.2024.3369419.

[21]A Novel Convolutional Neural Networks for Stock Trading Based on DDQN Algorithm (2023)

[22]Enhancing Stock Market Prediction: A Robust LSTM-DNN Model Analysis on 26 Real-Life Datasets (2024)

[23]Fusion of Technical Indicators and Sentiment Analysis in a Hybrid Framework of Deep Learning Models for Stock Price Movement Prediction (2024)

[24]G. S. Atsalakis and K. P. Valavanis, "Surveying stock market forecasting techniques–Part II: Soft computing methods," Expert Systems with Applications, vol. 36, no. 3,pp. 5932–5941, 2009.

[25]G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: Forecasting and control, 5th ed., Wiley, 2015.

[26]T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[27]O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," Applied Soft Computing, vol. 90, p. 106181, 2020.

[28]Y. Zhang, C. Aggarwal, and G. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2017, pp. 2141– 2149.