Bare Demo of IEEEtran.cls for IEEE Journals

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

Ahmed Medhat 230065 Shaza Ashraf 238765 Omar Ayman 234809 Andrew Talaat 237011 Mohamed Sherif 234323

***Abstract*—Stock market prediction is a challenging task due to the inherently noisy and non-linear nature of financial time series data. In this project, we explore the effectiveness of various machine learning models—including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, Logistic Regression, Linear Regression, and Convolutional Neural Networks (CNN)—in forecasting stock price trends. Each model is evaluated on a dataset containing daily stock price features such as 'Open', 'High', 'Low', 'Close', and 'Volume'. Evaluation metrics like Root Mean Square Error (RMSE) and R^2 score were used to assess predictive performance. Our results demonstrate that deep learning models, particularly CNNs, outperform traditional machine learning techniques, achieving the lowest RMSE and highest R^2 values.**

I. Introduction

Predicting stock market trends has long been a critical yet complex goal for investors and financial analysts. Traditional methods often rely on statistical models and domain knowledge, which may fall short in capturing the high volatility and non-linear behavior of market data. Machine learning (ML) techniques offer a promising alternative, capable of learning intricate patterns from large historical datasets [1].

The growing availability of financial data and advancements in computational power have enabled researchers to apply ML algorithms such as KNN, SVM, and CNN to financial forecasting tasks [2][3]. In this project, we aim to evaluate the performance of a range of models—from basic classifiers like Naive Bayes to more complex architectures like Convolutional Neural Networks—on the problem of stock price prediction.

Our main contributions are as follows:

- We compare the performance of six different ML algorithms on the same stock price dataset.
- We implement a CNN to capture temporal dependencies in the data.
- We develop a GUI that allows end-users to visualize predicted stock trends interactively.

The remainder of this paper is organized as follows: Section III reviews related work in the field, Section IV describes the dataset and preprocessing steps, Section V details the methodology, Section VI presents experimental results, and Section VII concludes the paper.

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

II. Related Work

A number of studies have examined the application of machine learning to stock price forecasting. Patel et al. [4] used a combination of decision trees, random forests, SVM, and neural networks to forecast the Nifty and SENSEX indices, concluding that ensemble models yielded better accuracy than individual classifiers. Zhang et al. [5] explored the use of Long Short-Term Memory (LSTM) networks for time-series prediction and showed improved performance over traditional models due to LSTM's ability to retain long-term dependencies.

In another study, Fischer and Krauss [6] compared deep learning models to logistic regression and random forests for predicting S&P 500 stock movements. Their results demonstrated the superiority of deep learning, especially in capturing non-linear trends.

Our approach differs in that we provide a direct side-by-side comparison of both classical and deep learning models, including a CNN model adapted for 1D financial data. Additionally, we contribute a usable interface for model output visualization.

III. Data Description and Preprocessing

The dataset used in this study was sourced from Yahoo Finance and includes daily historical stock data. The features extracted from the data include 'Open', 'High', 'Low', 'Close', and 'Volume'. Our target variable is the 'Close' price of the stock.

We performed the following preprocessing steps:

- **Missing Values**: Rows with missing values were dropped to maintain data integrity.
- **Normalization**: All numerical features were scaled using Min-Max normalization to ensure consistency across models.
- **Windowing**: For the CNN model, we created a sliding window of 30 consecutive days to predict the next day's 'Close' price.
- **Splitting**: The dataset was split into training (80%) and testing (20%) sets using chronological order to simulate real-world scenarios.

IV. Methodology/Approach

We implemented six different models for comparison:

K-Nearest Neighbors (KNN): A non-parametric method used for both classification and regression. We used k=5k = 5k=5 and Euclidean distance as the metric.

Model	RMSE	R^2 Score
Linear Regression	4.75	0.68
Logistic Regression	N/A	N/A
KNN	3.90	0.72
SVM	3.45	0.76
CNN	2.10	0.89

Support Vector Machine (SVM): A supervised model effective in high-dimensional spaces. For this task, we used Support Vector Regression (SVR) with a radial basis function (RBF) kernel.

Naive Bayes: Although generally suited for classification, we included it to assess how well it could predict categorical stock trends (up/down).

Logistic Regression: Used to classify stock movement direction, predicting whether the stock would go up or down the next day.

Linear Regression: Used to predict the next day's 'Close' price using a linear combination of input features.

Convolutional Neural Network (CNN): Our deep learning model consisted of:

- A Conv1D layer with 64 filters and kernel size 3
- A MaxPooling1D layer with pool size 2
- A Dropout layer (0.2 rate) to reduce overfitting
- A Dense layer with 50 units
- An output Dense layer with 1 unit
  The model was compiled with the Adam optimizer and 'mean_squared_error' loss, trained for 20 epochs.

Additionally, we developed a simple GUI using Python's Tkinter library to allow users to input a stock ticker and view both actual and predicted prices as a plot.

V. Results

The models were evaluated using two metrics:

- **Root Mean Square Error (RMSE)**: Measures average prediction error
- **R^2 Score**: Measures how well predictions approximate actual values

Performance Table:

Discussion: The CNN model outperformed all other models, demonstrating its strength in handling time-series data due to its ability to capture local patterns via convolutional filters. KNN and SVM also provided reasonable performance. Traditional classifiers like Naive Bayes and Logistic Regression were less effective, particularly for regression tasks. The GUI provided a helpful tool for visualizing predictions and real values, making our solution more interpretable and user-friendly.

Conclusion

This study explored and compared various machine learning techniques for stock market prediction, using historical price data. Traditional models like KNN and SVM showed reasonable predictive power, but deep learning models, particularly CNNs, significantly outperformed them in terms of RMSE and R^2. The success of CNN underscores the potential of deep learning in capturing complex patterns in sequential financial data.

While the results are promising, the study is limited by the scope of the dataset and lack of external features such as news sentiment or macroeconomic indicators. Future work may include the use of LSTM models, ensemble learning, or the integration of multi-source data to further improve predictive accuracy.