DEEP LEARNING FOR CANDLESTICK CHART CLASSIFICATION: PREDICTING STOCK PRICE MOVEMENTS USING CONVOLUTIONAL NEURAL NETWORKS

Thomas Bordino
Columbia University

***Abstract*—This paper explores the effectiveness of deep learning models, particularly Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, for predicting stock price movements based on candlestick chart data. More precisely, we have tried to predict the trend of the last hour of a trading day using the OHLCV data from the current day and the previous days. Various model architectures were tested, including simple CNN models and LSTM networks, trained on different time intervals (1-minute, 5-minute, and 15-minute) and augmented chart types (traditional candlestick, Heiken Ashi, and pattern-specific colorings such as bullish engulfing and Doji candles). The objective was also to evaluate how different intervals and augmentations impact the classification performance of these models.**

Initial results revealed that for single input, simple CNN models performed well for basic pattern recognition in the charts, particularly with 5-minute intervals and certain pattern augmentations. However, the LSTM models consistently outperformed simple CNN models, demonstrating the LSTM's ability to capture temporal dependencies in financial time-series data. The best performance was achieved with an LSTM model using sequences of 8 days, which achieved a validation accuracy of 63.9%. These findings suggest that LSTMs are more suited for predicting price trends even in computer vision, especially when leveraging augmented features such as Heiken Ashi and pattern-specific coloring.

I. INTRODUCTION

The prediction of stock price movements is a central challenge in financial forecasting. As markets are inherently dynamic and influenced by a wide range of factors, accurate predictions require sophisticated modeling techniques. Traditional methods such as technical analysis rely heavily on chart patterns, such as candlestick charts, to provide insights into market trends. Recent advances in deep learning have provided new ways to leverage these visual patterns through computer vision techniques, enabling more automated and scalable prediction models.

In this paper, we explore the application of deep learning models, particularly Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, for predicting stock price trends from candlestick chart data. The main objective is to predict the trend of the last hour of a trading day based on the OHLCV (Open, High, Low, Close, Volume) data from the current and previous trading days. We aim to determine the effectiveness of different time intervals (1-minute, 5-minute, and 15-minute) and augmentations of candlestick chart patterns, such as Heiken Ashi and specific price action patterns like bullish engulfing and Doji candles, in improving model performance.

Our experiments involved testing a variety of deep learning architectures. Simple CNN models were first trained on individual candlestick charts to assess their ability to recognize fundamental price patterns. Additionally, we experimented with LSTM models, which are designed to capture sequential dependencies, allowing them to potentially model the temporal relationships between consecutive days' price movements. This is critical for stock price trend prediction, where price movements are not only dependent on individual chart patterns but also on the temporal evolution of market behavior.

The results showed that simple CNN models performed well on basic pattern recognition tasks, particularly when trained with 5-minute interval data and specific augmentations. However, LSTM models consistently outperformed the CNN models, demonstrating the superior ability of LSTMs to capture the temporal dynamics inherent in stock price movements. The highest validation accuracy, 63.9%, was achieved with an LSTM model trained on sequences of 8 days, highlighting the effectiveness of this architecture in leveraging both historical and current market data.

This paper thus contributes to the growing body of research on the use of deep learning for stock price prediction. By evaluating multiple model architectures and input configurations, we demonstrate that LSTMs, combined with carefully selected input features, provide a powerful tool for predicting market trends based on candlestick chart data.

II. RELATED WORK

A. LSTM Models for Stock Price Prediction

Long Short-Term Memory (LSTM) networks have been widely adopted for stock price prediction due to their capability to capture long-term dependencies in time-series data. Karan et al. (2020) demonstrated the power of LSTM networks for financial forecasting, using an LSTM model to predict stock prices with considerable success. Their work emphasized that LSTMs could effectively handle the sequential nature of stock data and outperformed traditional machine learning models that did not take temporal relationships into account [1]. Similarly, Zhang et al. (2020) applied LSTM networks to stock market prediction, showing how these models could leverage historical stock prices to predict future price movements with high accuracy. Their findings supported the idea that LSTMs are particularly effective for time-series forecasting, where the focus is on sequential dependencies in the data [2].

B. CNN-LSTM Hybrid Models for Stock Price Prediction

In recent years, combining Convolutional Neural Networks (CNNs) with LSTM networks has become a popular approach to stock price prediction, as it allows for the integration of both spatial and temporal information. Zhang et al. (2021) proposed a hybrid CNN-LSTM model for stock price forecasting, where CNNs were employed to extract features from stock chart images, and LSTMs were used to capture the temporal dependencies between stock price movements. Their study highlighted that this combined approach enhanced prediction accuracy, especially in capturing complex patterns present in stock charts [3]. The result from this study demonstrated that CNN-LSTM models could outperform traditional approaches by effectively leveraging both visual and temporal information to predict stock price movements .

C. Lack of Research on Candlestick Chart Plotting Styles

Although the use of CNN-LSTM hybrid models for stock price prediction has been extensively explored, there is a noticeable gap in the literature regarding the impact of different candlestick chart plotting styles on model performance. While traditional candlestick charts have been used in various studies, chart types such as Heiken Ashi or Renko charts have not been thoroughly investigated in the context of deep learning for stock price forecasting. Research into the effects of different chart styles on prediction accuracy could offer valuable insights, potentially improving the robustness of financial forecasting models.

III. METHODOLOGY

The methodology of this study encompasses three primary stages: data fetching, data plotting, and data splitting. Each stage is meticulously designed to prepare the dataset for effective training and evaluation of Convolutional Neural Network (CNN) models for stock price movement prediction.

A. Data Fetching

The dataset utilized in this research comprises minute-level stock price data for 24 distinct tickers, spanning from January 1, 2000, to September 30, 2024. This extensive dataset includes over 6,225 trading days per ticker, resulting in a comprehensive collection of financial data.

Data was sourced using the Alpha Vantage API, which provides reliable and up-to-date financial information. The fetched data includes essential Open, High, Low, Close, and Volume (OHLCV) metrics for each minute interval during trading hours. To ensure data integrity, preprocessing steps were undertaken to handle missing values, outliers, and any discrepancies across different tickers. Each trading day's data was standardized to align with market hours (9:30 AM to 4:00 PM Eastern Time), facilitating uniformity across all time series.

The cleaned data was organized into individual CSV files, categorized by ticker and date. This structured approach enabled efficient access and streamlined the subsequent data plotting process.

B. Data Plotting

Transforming raw numerical data into visual representations is crucial for leveraging CNNs' spatial feature extraction capabilities. Three distinct types of financial charts were generated to capture various aspects of stock price movements: Candlestick Charts, Heiken-Ashi Charts, and Pattern-Enhanced Candlestick Charts.

1) Candlestick Charts

Candlestick charts are fundamental tools in technical analysis, offering a visual summary of price movements within specific time intervals. Each candlestick encapsulates four key data points:

- **Open**: The price at the beginning of the interval.
- **High**: The highest price achieved during the interval.
- **Low**: The lowest price recorded during the interval.
- **Close**: The price at the end of the interval.

The body of the candlestick represents the range between the opening and closing prices, while the wicks (or shadows) indicate the extreme high and low prices. The color of the candlestick conveys the price trend within the interval: green denotes an upward movement (close>open), and red signifies a downward movement (close <open).

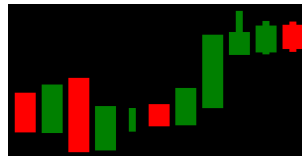
Fig. 1: Interval: 30min


Fig. 2: Interval: 15min

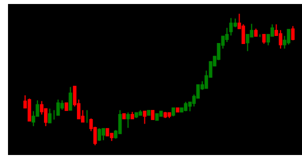
Fig. 3: Interval: 5min

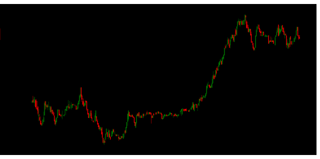
Fig. 4: Interval: 1min

Candlestick charts were generated for each trading day using data aggregated at different intervals, where each candlestick represents 1, 5, 15, or 30 minutes of trading activity, covering from 9:30 AM to 3:00 PM Eastern Time. These charts serve as the baseline visual input for the CNN models, encapsulating comprehensive price movement information leading up to the final trading hour.

2) Heiken-Ashi Charts

Heiken-Ashi charts are a variant of traditional candlestick charts, designed to smooth out price data and emphasize longer-term trends by mitigating market noise. Unlike standard candlesticks, Heiken-Ashi candles are computed using a modified formula that incorporates the average of previous periods.

The formulas for calculating Heiken-Ashi values are as follows:

$$\begin{aligned}
\text{Close}_{HA} &= \frac{\text{Open} + \text{High} + \text{Low} + \text{Close}}{4} \\
\text{Open}_{HA} &= \frac{\text{Open}_{prev} + \text{Close}_{prev}}{2} \\
\text{High}_{HA} &= \max(\text{High}, \text{Open}_{HA}, \text{Close}_{HA}) \\
\text{Low}_{HA} &= \min(\text{Low}, \text{Open}_{HA}, \text{Close}_{HA})
\end{aligned}$$

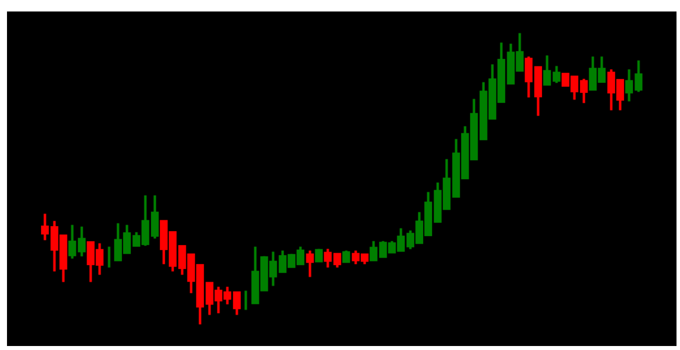
Fig. 5: Heiken-Ashi Chart

This smoothing technique reduces the impact of minor price fluctuations, making it easier to identify sustained trends and reversals.

Heiken-Ashi charts were created for each trading day, mirroring the time intervals used in traditional candlestick charts. This alternative visualization provides the CNN models with a different perspective on price movements, potentially enhancing the model's ability to detect underlying trends.

3) Pattern-Enhanced Candlestick Charts

To further refine the input data for the CNN models, pattern-enhanced candlestick charts were developed by highlighting specific candlestick patterns known to indicate potential market reversals or continuations. The following patterns were identified and color-coded:

- **Bullish Engulfing**: A green candlestick that completely engulfs the body of the preceding red candlestick, suggesting a potential upward reversal.
- **Bearish Engulfing**: A red candlestick that completely engulfs the body of the preceding green candlestick, indicating a potential downward reversal.
- **Doji**: A candlestick with a very small body, representing market indecision where the opening and closing prices are nearly equal.

By programmatically detecting these patterns and applying distinct colors, the pattern-enhanced charts emphasize critical formations that may influence price movements, thereby aiding the CNN models in recognizing and learning from these significant indicators.

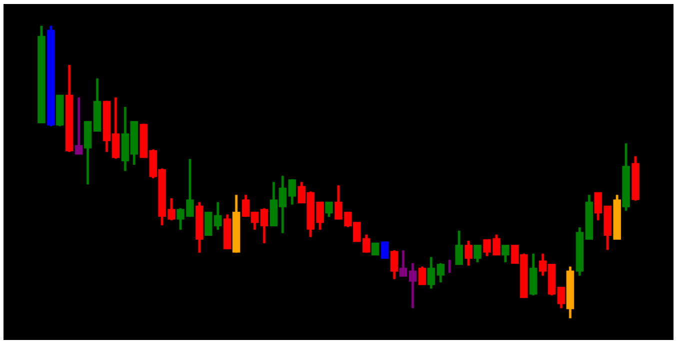

Fig. 6: Pattern-Enhanced Candlestick

Each pattern-enhanced chart was generated for the corresponding trading day, ensuring that the highlighted patterns are accurately represented within the visual data fed into the CNN models.

C. Data Splitting

To effectively evaluate the performance of CNN models, the data set was divided into training and testing subsets based on temporal segmentation. This approach ensures that the models are trained on historical data and tested on unseen future data, simulating real-world forecasting scenarios.

- **Training Set**: Comprises data from January 1, 2000, to June 30, 2023. This subset includes the majority of the dataset, providing the models with ample information to learn from various market conditions and trends.
- **Testing Set**: Encompasses data from July 1, 2023, to September 30, 2024. This subset serves as an unseen dataset to evaluate the model's predictive capabilities on new data.

Each chart within both subsets was assigned a binary label based on the price movement during the last trading hour:

- **Label 1**: Indicates that the closing price in the last hour was higher than the opening price (upward movement).
- **Label 0**: Indicates that the closing price in the last hour was not higher than the opening price (downward or no movement).

This labeling strategy provides a clear and actionable target variable for the CNN models, enabling them to learn the association between visual chart patterns and subsequent price movements. The temporal split ensures that the training process is free from data leakage, maintaining the integrity of the model evaluation.

The final dataset consists of more than 3 million charts across all tickers. This comprehensive preparation facilitates the training of robust CNN models capable of generalizing across diverse market conditions and ticker-specific behaviors.

D. Models

Each models were optimized through a trial-and-error approach, adjusting various hyperparameters and configurations to improve validation accuracy on the specific dataset. This iterative process allowed for fine-tuning the models to better capture patterns in the candlestick charts and enhance predictive performance.

1) Test on Full-Day Images

The initial phase of modeling focused on using full-day candlestick chart images to assess the effectiveness of Convolutional Neural Networks (CNNs) for trend prediction. A variety of CNN architectures were tested on these full-day charts to identify the most relevant model for the task. This served as a baseline to determine whether CNNs could effectively capture and learn patterns in the full-day candlestick data, setting the stage for further model refinement.

2) Augmentation and Model Refinement

Building on the results from the initial phase, the next step involved experimenting with augmented candlestick chart types, including different time intervals and partial chart representations. These augmentations were incorporated into a single CNN model to evaluate their impact on model performance. By augmenting the dataset, we aimed to explore how these variations influenced the model's ability to generalize and improve predictive accuracy, ultimately refining the model to handle diverse chart representations effectively.

3) Models with Multiple Images as Inputs

The final stage of modeling focused on using multiple images as inputs to provide a richer representation of the trading data. Two key approaches were tested:

- **Multiple Intervals of the Same Day:** Candlestick charts were plotted at different intervals (e.g., 1-minute, 5-minute, 15-minute, and 30-minute) for the same trading day. These interval-based charts were used together as inputs to a CNN to evaluate if capturing varying levels of granularity within the same day could improve prediction performance.
- **Sequence of Current and Previous Days:** To incorporate temporal context, candlestick charts from the current trading day (excluding the last hour) and several previous trading days were used in sequence. A hybrid model combining CNN-based feature extraction and sequential layers, such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRUs), was implemented to learn both spatial patterns in the charts and temporal relationships across multiple days.

IV. EXPERIMENTS AND RESULTS

A. Experiments

1) Plotting Method

A first function generates candlestick charts for a given ticker at specified time intervals. It starts by loading daily data from CSV files, converting timestamps and ensuring OHLC values are numeric. The data is sorted by timestamp, and the last hour is removed to focus on earlier trading periods.

The function resamples the data at the specified interval (e.g., 5 minutes) using aggregation methods for OHLC values. If there are insufficient data points after resampling, the file is skipped. Candlestick charts are then plotted, with green bars for upward movement and red for downward, and saved with a filename reflecting the ticker and date.

To handle the large number of charts efficiently, the `Agg` backend for `matplotlib` is used, which minimizes memory usage by saving images directly without rendering them interactively. After each plot is saved, `gc.collect()` is used to free memory, preventing crashes when generating thousands of charts.

2) Processing and Loading

A function preprocesses and splits candlestick chart images of various intervals and their labels into memory-mapped files for efficient storage and retrieval. It starts by loading the image paths and checking their validity. Valid images are resized to the specified dimensions, while invalid paths result in black images.

The data is then divided into training and testing sets based on a specified index. The function writes the images and their corresponding labels to memory-mapped files, ensuring efficient access without overloading the memory. After writing, the memory-mapped files are flushed to disk.

A custom data loader class is used to load the images and labels from the memory-mapped files in batches during training. This ensures efficient handling of large datasets. The loader is adaptable to the input you want to provide to the model, whether it's a single image, a sequence of images, or several independent images.

3) CNN Models

The model consists of several layers aimed at extracting high-level features from candlestick chart images. It begins with a series of convolutional layers designed to capture spatial patterns in the images. Each convolutional layer is followed by an activation function (typically ReLU) and a pooling layer to downsample the feature maps.

Model 1: Custom CNN for Candlestick Chart Classification The CNN architecture is structured as follows:

- **Input Layer:** Accepts input images, typically with dimensions of $300 \times 300 \times 3$ (height, width, and RGB channels).
- **Convolutional Layers:** Multiple convolutional layers apply filters to the input images, extracting feature maps that highlight various patterns such as trends, candlestick shapes, etc.
- **Activation Function:** ReLU is applied after each convolutional layer to introduce non-linearity.
- **Pooling Layers:** Max-pooling layers are used after each convolutional layer to reduce spatial dimensions and focus on the most important features.
- **Fully Connected Layers:** Flattened feature maps are passed to one or more fully connected layers, which perform classification tasks.
- **Output Layer:** A softmax or sigmoid activation function is used in the output layer to predict binary classification (up or down).

The model is trained using a cross-entropy loss function for binary classification, optimized via the Adam optimizer.

Model 2: Fine-Tuned ResNet In Model 2, a fine-tuning approach is applied to the ResNet architecture. The first layers of the pre-trained ResNet model are frozen to retain the learned features from its initial training, while the last 20 layers are unfrozen to adapt the model to the specific task of binary classification. A fully connected layer is added on top to map the features to the binary classification output. This model leverages the pre-trained weights and features of ResNet, with an emphasis on transfer learning for improved performance in candlestick chart classification.

Model 3: CNN with Multi-Head Attention Model 3 is similar to Model 1 but with an enhancement through the inclusion of a multi-head attention layer. The attention mechanism allows the model to focus on different parts of the input chart simultaneously, capturing various aspects of the candlestick patterns at different scales. This can help the model identify subtle trends and relationships that might not be captured by the standard convolutional layers alone. The output from the attention layer is then passed through a series of convolutional and fully connected layers to perform the binary classification task.

Model 4: Triple-Input CNN for Multi-Interval Analysis Model 4 is designed to leverage candlestick charts from multiple time intervals—1-minute, 5-minutes, and 15-minutes—providing a comprehensive view of price action at different granularities. The architecture includes:

- **Input Layers:** Three separate inputs for candlestick charts corresponding to 1-minute, 15-minute, and 5-minute intervals.
- **Convolutional Blocks:** Each input is processed independently through three convolutional layers (Conv2D) with increasing filters (16, 32, 64), each followed by max-pooling for feature extraction.
- **Feature Flattening:** Outputs from all three branches are flattened to vector form.
- **Feature Merging:** The flattened outputs are concatenated to combine features across intervals.
- **Fully Connected Layers:** The merged features are passed through a dense layer (64 units) and dropout for regularization.
- **Output Layer:** A sigmoid activation for binary classification (up or down).

This model captures distinct patterns and trends at multiple time scales, enhancing its ability to classify the future price movement based on past candlestick data. It is optimized using binary cross-entropy loss and the Adam optimizer.

Model 5: CNN-LSTM The CNN-LSTM model integrates both Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to capture spatial and temporal dependencies in candlestick chart images. It consists of two main inputs: one for the current chart image and one for a sequence of previous chart images plotted on complete day.

The architecture is structured as follows:

- **Input Layers:** A main input for the current chart image and a sequence input for a series of past images.
- **Convolutional Blocks:** Three convolutional layers (Conv2D) with increasing filters (16, 32, 64) followed by max-pooling, applied to the main input input.
- **Sequence Processing:** The sequence of images is processed through TimeDistributed CNN layers and then passed to a bidirectional LSTM layer (128 units) followed by a second LSTM layer (64 units).
- **Fully Connected Layers:** CNN and LSTM outputs are concatenated, followed by a dense layer (128 units) and dropout for regularization.
- **Output Layer:** A sigmoid activation for binary classification (up or down).

Model 6: CNN-GRU Model 5 is the same as model 4, except that GRU (Gated Recurrent Unit) layers are used instead of LSTM layers.

The precise architecture of the final models is chosen by trial and error. They are then trained using a cross-entropy loss function for binary classification, optimized via the Adam optimizer, in order to obtain the following results

B. Results

Each model was trained using a batch size of 32, with an initial learning rate of 0.0001 adjusted dynamically using a learning rate scheduler. Images originally plotted at a size of 1200x1200 pixels were resized to 300x300 pixels. Training was conducted for 50 epochs on a dataset evenly distributed across the binary classes (up and down). The optimizer used was Adam, and the loss function was binary cross-entropy. Early stopping with a patience of 10 epochs was applied to prevent overfitting, and the best model was selected based on validation accuracy.

1) Full Day Chart

Before delving into prediction tasks, it is crucial to evaluate whether the models can effectively recognize patterns from candlestick charts. This test is performed using charts representing the entire trading day without any future price movement prediction. The objective is to verify that the models can process the visual and temporal information embedded within the charts and retrieve meaningful patterns related to market behavior.

Testing on full-day charts also helped to refine the model architecture by enabling us to choose relevant parameters, such as the number of layers, filter sizes, and the appropriate image reshaping dimensions. By assessing the models' performance on these charts, we could identify and fine-tune the key aspects of the architecture, ensuring that the models were capable of distinguishing significant features, such as trends, volatility, and candlestick shapes, which are foundational to any subsequent predictive tasks. This step serves as a baseline assessment, highlighting the pattern recognition ability of the models without the added complexity of forecasting.

Interval	Model	Validation Accuracy
1 min	Model 1	0.94444
1 min	Model 2	0.53055
1 min	Model 3	0.54662
1/5/15 min	Model 4	0.948555
1 min	Model 5 (sequence length = 4)	0.93247
1 min	Model 6 (sequence length = 4)	0.92926

TABLE I: Validation accuracy for different input configurations.

Model 1, the simple CNN, performs best, likely due to its simplicity and ability to effectively capture spatial patterns in the geometric structure of candlestick charts. This suggests that for this specific task, a straightforward CNN is more suitable than more complex models.

Model 2, the fine-tuned ResNet, performs poorly despite its pre-trained weights. This may be because ResNet was trained on real-world images, making it less adaptable to the abstract and structured nature of candlestick charts. Model 3, which adds multi-head attention to the CNN, shows a slight improvement but still doesn't surpass the custom CNN, indicating that the attention mechanism doesn't offer substantial benefits for this dataset.

Models 5 and 6 are not used to their full potential in this preliminary problem, since the model will focus the features for classification on the last image and its final trend.

2) Single Input Augmentation

In this approach to single input augmentation, we explore the impact of different time intervals on model performance by training Model 1 separately on candlestick charts from various intervals, including 1-minute, 5-minute, and 15-minute time frames. Each interval captures distinct market behaviors and provides the model with unique perspectives on price movements. By experimenting with these intervals individually, we aim to identify which interval offers the most informative and reliable features for the classification task.

Additionally, we augment the dataset by incorporating different types of candlestick chart representations. This includes traditional candlestick charts as well as Heiken Ashi charts, which are known for their smoother appearance and ability to highlight trends more clearly. Furthermore, specific chart patterns, such as bullish and bearish engulfing patterns and Doji candles, are colored to emphasize their significance. These augmentations aim to improve the model's ability to identify key price action signals and patterns, thereby enhancing classification accuracy for candlestick chart analysis. (In the following table, CDL is for candlestick, HA for Heiken Ashi, E for engulfing and D for Doji).

Interval	Pattern Recognition	Chart Type	Validation Accuracy
30 min	None	CDL	0.50347
15 min	None	CDL	0.55903
5 min	None	CDL	0.56255
1 min	None	CDL	0.55903
5 min	E	CDL	0.56597
5 min	ED	CDL	0.57986
5 min	None	HA	0.57986
5 min	ED	HA	0.59722

TABLE II: Validation accuracy for different input configurations.

The results demonstrate that augmentations with different chart types and patterns improve performance, especially when incorporating Heiken Ashi (HA) charts. Notably, the best results are observed with the 5-minute interval when both the "ED" (engulfing and doji) pattern coloring and Heiken Ashi charts are applied, leading to a notable increase in accuracy. This suggests that these augmentations help the model better identify significant patterns, such as engulfing and Doji candles, which are crucial for predicting price movements.

On the other hand, the 30-minute interval with no augmentations shows the lowest performance, indicating that longer time intervals may not provide as much actionable information for classification. The 5-minute interval, especially with "ED" and "HA" augmentations, consistently outperforms the other intervals, highlighting the effectiveness of combining shorter timeframes with specific pattern-focused augmentations. This cross-analysis confirms the importance of interval selection and chart type in enhancing model performance.

C. Hybrid Deep Learning Models

To enhance the performance of the multi-interval, LSTM, and GRU models, we made several adjustments to the input data and training parameters. A key change was reducing the image size to 150x150 pixels to decrease computational load while still maintaining sufficient detail for the models to extract meaningful features. Additionally, we increased the number of epochs to 100 to allow the models to fully train on larger input data. Based on previous results, we selected the augmented 5-minute interval data as it provided a good balance between model performance and computational efficiency. These adjustments were necessary to ensure the models could effectively learn from the multi-input data while managing the increased computational demands.

Interval	Model Logic	Sequence Length	Validation Accuracy
1/5/15 min	Multi-Input	3	0.54264
5 min	LSTM	3	0.62379
5 min	LSTM	4	0.62700
5 min	LSTM	8	0.63987
5 min	LSTM	16	0.59163
5 min	GRU	8	0.60128

TABLE III: Validation accuracy for different input configurations.

The multi-interval model shows no particular interest compared with models using a single interval. The results indicate that LSTM models generally outperformed the GRU model, which suggests that LSTM networks are better at capturing the temporal dependencies in this dataset. Increasing the sequence length for the LSTM models initially led to improvements in performance, with the best results achieved using a sequence length of 8. However, the performance slightly declined with a sequence length of 16, possibly due to overfitting or excessive complexity.

V. Conclusion and Future Work

A. Conclusion

In this paper, we explored various deep learning architectures for predicting the end-of-day trend based on candlestick chart images. We demonstrated that simple convolutional neural networks (CNN) were particularly effective for this task, especially when trained with 5-minute interval data, which enabled the models to efficiently capture patterns in geometrical chart shapes. By leveraging more complex models, such as LSTM and GRU networks, we observed improvements in performance, as these models captured temporal dependencies within chart sequences. Despite this, the increase in accuracy was marginal, and the complexity of training these models was substantially higher. These findings provide valuable insights for the field of financial chart pattern recognition, suggesting that simpler models might be more practical for real-time trading systems, especially when trained on short time intervals.

B. Future Work

During the course of this research, we attempted to implement a graph-based network that could link each candlestick chart for a given day to all charts from previous dates for the same ticker. However, this approach was computationally expensive and did not yield satisfactory results due to the vast amount of data that needed to be processed. Furthermore, we tried to modify the inputs of the LSTM to consider several images with fewer candles for the same day, which could potentially improve performance by trying to identify temporal intra-day patterns.

Additionally, while this study focused on the end-of-day trend prediction, the same methodology could be extended to explore various timeframes and price movements throughout the trading day. By expanding the scope of the problem, more complex trends could be detected, improving model robustness. Further enhancement could be achieved by incorporating additional features such as numerical indicators, sentiment analysis derived from news, or macroeconomic data, all of which could provide complementary information to the models. Although attempts were made to include these data sources, limited time and resources prevented these efforts from significantly improving model accuracy. Future work could explore these avenues in more depth, potentially increasing the relevance and performance of the models for real-world trading systems.

References

[1] S. Karan, J. Wang, *Stock Price Prediction Using LSTM, RNN, and CNN-Sliding Window Model*, IEEE Xplore, 2020.

[2] G. Zhang, L. Wang, *Stock Market Prediction Using Deep Learning LSTM Model*, MDPI, 2020.

[3] Y. Zhang, X. Liu, *Stock Price Prediction Using Combined LSTM-CNN Model*, MDPI, 2021.