Improving Stock Market Forecasting: Comparative Insights into CNN Architectures with a Focus on ReLU Activation

Dedy Hartama*, Mesran, Agus Perdana Windarto, Putrama Alkhairi, and Weni Rosdiana

***Abstract*—This study aims to improve the performance of stock market forecasting models by conducting a comparative analysis of CNN architectures using two primary activation functions: Rectified Linear Unit (ReLU) and Sigmoid. By evaluating metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE) across several models, including IDCNN, LSTM, DCN, ResNet, as well as the Ensemble IDCNN + LSTM and Proposed Ensemble models, it was found that ReLU consistently outperforms Sigmoid. The results show that models utilizing ReLU, particularly the Ensemble IDCNN + LSTM, achieve the lowest error rates (MSE: 0.0023 and MAE: 0.0361), demonstrating its ability to capture complex non-linear patterns in stock market data. In contrast, models using Sigmoid exhibited higher error rates, indicating that Sigmoid is less capable of handling the generalization challenges in volatile financial data. This study provides important insights into the impact of activation functions on deep learning model performance and recommends ReLU as the primary activation function for stock market forecasting tasks. With ReLU, models can deliver more accurate and reliable predictions in the dynamic stock market environment.**

***Index Terms*—ReLU, Sigmoid, Deep Learning, Mean Squared Error, Mean Absolute Error**

I. Introduction

IN the dynamic global financial landscape, accurately predicting stock market movements remains a complex challenge [1]. Investors and market participants rely on analysis and predictive models to make informed investment decisions and mitigate risks. However, ensuring the accuracy of stock market predictions is crucial due to the high level of uncertainty in stock price movements [2], [3]. Therefore, it is of paramount importance to conduct research aimed at enhancing the accuracy of stock market predictions [4], [5]. In recent years, various studies have explored the use of deep learning models [6], specifically convolutional neural networks (CNN) and its variants [7], for stock price prediction [8]. Notable examples include the utilization of CNN by [9], transfer learning in stock price prediction by [10], LSTM methods for improved stock market predictions in Malaysia by [11], CNN-LSTM-based models for stock price forecasting by [12], CNN implementation by [13], DCGAN utilization for stock price forecasting by [14], and a comprehensive literature review on machine learning techniques in stock market prediction by [15]. Although deep learning models such as CNN, LSTM, and DCGAN have been extensively employed for stock price prediction, they also have limitations. For example, the LeNet CNN method, popularized by a study conducted by [16], demonstrated its ability to identify patterns and trends in time series data of stock markets. Integrating LeNet with other methods, such as LSTM, as highlighted by [17], further enhanced prediction accuracy. However, these methods have limitations, including a limited understanding of complex data relationships and vulnerability to overfitting. Therefore, the proposed method by researchers aims to enhance the accuracy of stock market predictions by integrating Convolutional Neural Networks (CNN) into financial forecasting models. The proposed method is compared with several CNN methods by performing hyperparameter tuning, specifically focusing on activation functions. The types of activation functions used are Relu and Sigmoid across various data sets. The objective of this research is to provide deeper insights into the application of Convolutional Neural Networks in financial forecasting and to emphasize the importance of selecting optimal activation functions. Nevertheless, these limitations offer opportunities for innovation and the development of more effective and efficient CNN architectures for stock market prediction [13].

II. Literatue Review

In the realm of stock market prediction, various studies have utilized diverse machine learning techniques such as LSTM, CNN, and DCGAN. For example, the study by Bhandari et al. employed LSTM for stock market index prediction but did not address overfitting issues [1]. Subsequently, the research by Chen & He used CNN for stock prediction but did not compare its performance with other methods [9]. Dai's study applied an enhanced CNN in financial forecasting but did not detail the enhancement process [8]. Komori's research utilized CNN with transfer learning for stock price prediction but did not explain the model selection for transfer learning [10].

Additionally, Ku et al.'s study used LSTM with dynamic indicators for predicting the Malaysian stock market but did

Manuscript received February 29, 2024; revised November 5, 2024.

Dedy Hartama is a lecturer of STIKOM Tunas Bangsa, Pematangsiantar, North Sumatra, Indonesia. (corresponding author to provide e-mail: dedyhartama@amiktunasbangsa.ac.id).

Mesran is a lecturer of Sekolah Tinggi Ilmu Manajemen Sukma, Medan City, North Sumatra, Indonesia. (e-mail: mesran.skom.mkom@gmail.com).

Agus Perdana Windarto is a lecturer of STIKOM Tunas Bangsa, Pematangsiantar, North Sumatra, Indonesia. (e-mail: agus.perdana@amiktunasbangsa.ac.id).

Putrama Alkhairi is a lecturer of STIKOM Tunas Bangsa, Pematangsiantar, North Sumatra, Indonesia. (e-mail: putrama@amiktunasbangsa.ac.id).

Weni Rosdiana is a lecturer of Public Administration, Universitas Negeri Surabaya, East Java, Indonesia. (e-mail: wenirosdiana@unesa.ac.id).

not compare their method's performance with others [11]. Liu et al. combined LSTM with online social networks for stock price prediction but did not explain the data integration process [4]. Lu et al. employed a CNN-LSTM-based model for stock price forecasting but did not detail its implementation [12]. Lv et al used time series decomposition and a hybrid model for stock index prediction but did not elaborate on their implementation [5]. Sayavong et al utilized CNN for stock price prediction but did not provide implementation details [13]. Staffini used a Deep Convolutional Generative Adversarial Network for stock price forecasting but did not specify the implementation process [14]. Lastly, Wiranata & Djunaidy conducted a comprehensive literature review on stock exchange prediction using machine learning techniques but did not detail their review process [15]. Despite these significant advancements, a gap remains in exploring new and more efficient CNN model architectures and testing these models across multiple datasets. This research aims to fill this gap by developing an innovative CNN model architecture and evaluating its performance across various datasets, particularly in the use of activation functions. In doing so, it contributes to the ongoing evolution of machine learning techniques in stock market prediction.

III. Materials and Methods

A. Dataset Acquisition

This dataset consists of various daily features of the S&P 500, NASDAQ Composite, Dow Jones Industrial Average, RUSSELL 2000, and NYSE Composite from 2010 to 2017. It includes features from different categories such as technical indicators, futures contracts, commodity prices, significant global market indices, prices of major companies in the U.S. market, and treasury bill rates (TABLE I).

This dataset will be used to test the CNN model architecture developed in this research. By utilizing this dataset, we will evaluate the performance of the innovative CNN model and determine its effectiveness in predicting stock market movements.

TABLE I
Indices Explained

Name	Description
S&P 500	Index of 505 companies included in the S&P stock market
Dow Jones Industrial Average	Index of 30 leading U.S. companies in the Dow Jones Industrial Average
NASDAQ Composite	Composite Index of common stocks listed on the NASDAQ stock market
NYSE Composite	Index of common stocks listed on the New York Stock Exchange
RUSSELL 2000	Index of 2000 small-cap companies in the U.S.

The research on stock price prediction follows a systematic framework, represented by a flowchart to provide a clear visual depiction of each research stage. This flowchart comprises several key stages: data collection, data preprocessing, model training, model evaluation, and model performance comparison (Figure 1).

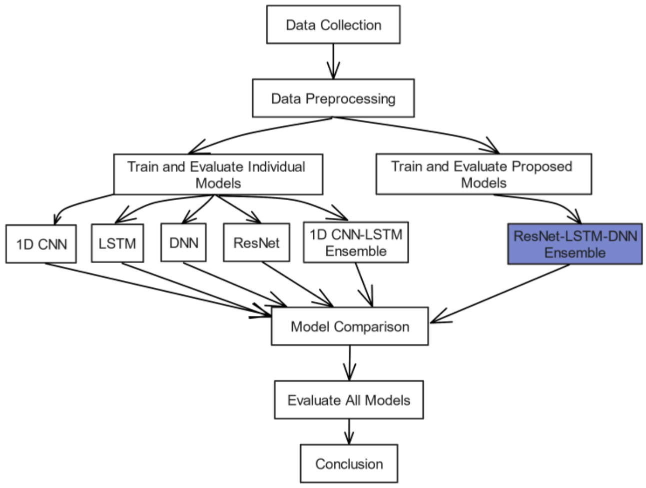


Fig. 1. Research Framework Overview

The research process (Figure 1) begins with the collection of stock price data from various sources, followed by data preprocessing. During preprocessing, the data is merged, cleaned, and normalized, with input sequences and labels created. Once the data is prepared, individual models are trained and evaluated. These models include IDCNN, LSTM, DCN, and ResNet. Additionally, an IDCNN+LSTM ensemble is trained and evaluated. The next phase involves training and evaluating the proposed model, which is the ResNet+LSTM+DCN ensemble. This model is expected to perform the best based on preliminary evaluations. All trained models are then compared using appropriate evaluation metrics. The process concludes with drawing conclusions based on the comparison results, identifying the best model for future stock price prediction.

B. Data Preprocessing

Data preprocessing is a crucial step in the stock price prediction research framework, ensuring that the data is clean, normalized, and ready for model training. The preprocessing stage involves several key steps, including data merging, handling missing values, normalizing data, and creating sequences for input and labels. Below is a detailed narrative of the data preprocessing steps, accompanied by the results from each step (Figure 2).

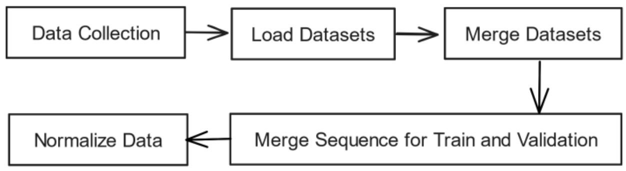


Fig. 2. Data Preprocessing Steps for Stock Market Prediction

In stock price prediction research, the data preprocessing stage is essential to ensure that the data used for model training is clean, normalized, and structured. The flowchart (Figure 2) outlines the steps involved in the data preprocessing stage, followed by the results obtained at each step. The data preprocessing stage involves merging multiple datasets, handling missing values, normalizing the data, and creating input sequences and labels. This ensures the data is clean, well-structured, and ready for model training. The resulting dataset is split into training and testing sets, setting the stage for developing and evaluating predictive models.

C. *Proposed Model*

The proposed method for stock price prediction leverages the strengths of three advanced deep learning architectures: Residual Networks (ResNet), Long Short-Term Memory (LSTM) networks, and Dilated Convolutional Neural Networks (DCN). This ensemble approach aims to improve prediction accuracy by combining the unique capabilities of each model. Firstly, Residual Networks (ResNet) are employed for their ability to handle vanishing gradient problems and enable the training of very deep networks. ResNet captures complex patterns in stock prices by utilizing residual blocks, which help in learning high-level representations of the data [18]. Secondly, Long Short-Term Memory (LSTM) networks are integrated into the method for their proficiency in capturing long-term dependencies and temporal relationships in sequential data. LSTMs are particularly well-suited for time series forecasting tasks [18], such as predicting stock prices, due to their ability to remember long-term patterns. Thirdly, Dilated Convolutional Neural Networks (DCN) are used for their capacity to capture multi-scale temporal patterns in time series data through dilated convolutions. This helps in understanding both short-term and long-term trends in stock prices [19]. DCNs expand the receptive field exponentially without losing resolution, making them effective for this task. The ensemble learning approach is then applied to combine the predictions from the ResNet, LSTM, and DCN models. By averaging the predictions from these models [20], [21], the ensemble method reduces variance and improves the robustness of the predictions, leading to more accurate stock price forecasts.

TABLE II
LAYER DETAILS OF THE PROPOSED MODEL

Model	Layer Type	Hyperparameter	Description
ResNet	Input	Shape: (sequence_length, num_features)	Input layer for ResNet model
	Conv1D	Filters: 64, Kernel Size: 3, Activation: ReLU	Convolutional layer
	Residual Block	Filters: 64, Kernel Size: 3, Blocks: 3, Activation: ReLU	Three residual blocks with Conv1D layers
	Flatten	-	Flatten layer
	Dense	Units: 50, Activation: ReLU	Fully connected layer
	Output	Units: 1	Output layer
LSTM	Input	Shape: (sequence_length, num_features)	Input layer for LSTM model
	LSTM	Units: 50, Return Sequences: True	LSTM layer
	LSTM	Units: 50	LSTM layer
	Dense	Units: 50, Activation: ReLU	Fully connected layer
	Output	Units: 1	Output layer
DCN	Input	Shape: (sequence_length, num_features)	Input layer for DCN model
	Conv1D	Filters: 64, Kernel Size: 2, Dilation Rate: 1, Activation: ReLU	Dilated convolutional layer
	Conv1D	Filters: 64, Kernel Size: 2, Dilation Rate: 2, Activation: ReLU	Dilated convolutional layer
	Conv1D	Filters: 64, Kernel Size: 2, Dilation Rate: 4, Activation: ReLU	Dilated convolutional layer
	MaxPooling1D	Pool Size: 2	Max pooling layer
	Flatten	-	Flatten layer
	Dense	Units: 50, Activation: ReLU	Fully connected layer
	Output	Units: 1	Output layer
Ensemble	Averaging	-	Average predictions from ResNet, LSTM, DCN models

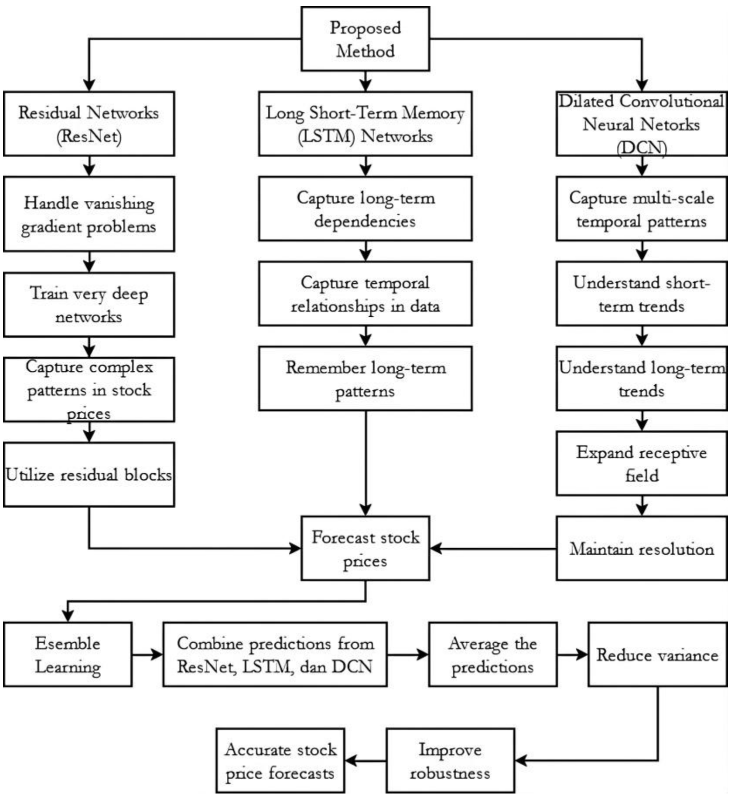


Fig. 3. Overview of Proposed Prediction Method

To further optimize the model performance (Fig. 3), we analyze the impact of different activation functions, specifically Sigmoid and ReLU. The Sigmoid activation function maps input values to a range between 0 and 1, which is suitable for binary classification tasks but can suffer from vanishing gradient issues. On the other hand, the ReLU activation function outputs the input directly if it is positive, and zero otherwise. ReLU helps mitigate the vanishing gradient problem and is widely used in deep learning models.

Sigmoid Activation Function:

$$\sigma(x) = \frac{1}{1+e^{-z}} \tag{1}$$

$$\sigma'(x) = \sigma(x)(1-\sigma(x)) \tag{2}$$

ReLU Activation Function:

$$ReLU(x) = \max(0, x) \tag{3}$$

$$ReLU'(x) = \begin{cases} 1, & if\ x > 0 \\ 0, & otherwise \end{cases} \tag{4}$$

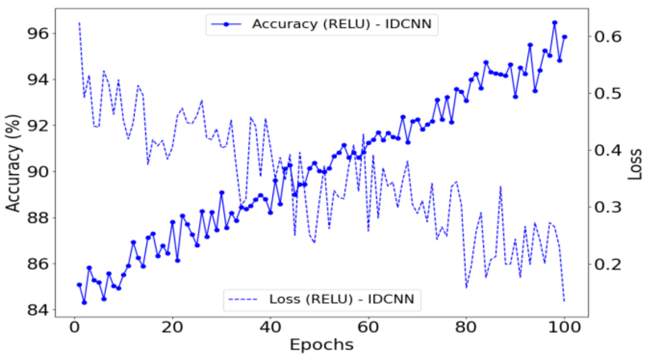
In practice, each model (ResNet, LSTM, and DCN) is defined and trained using both Sigmoid and ReLU activation functions. The models are then evaluated based on metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), accuracy, precision, recall, and F1 score. By comparing the performance of models using Sigmoid versus ReLU activation functions, we can identify the most effective configuration.

IV. Results and Discussion

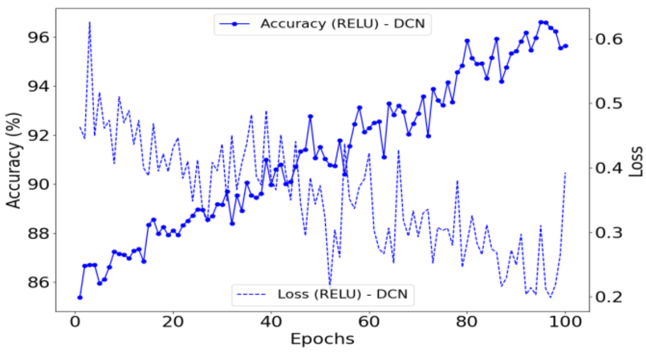
This section presents the outcomes from the experiments conducted using various models with Sigmoid and ReLU activation functions. The results, comprising both numerical data and visual representations, provide a comprehensive understanding of model performance across different metrics.

A. *Performance with ReLU Activation Function*

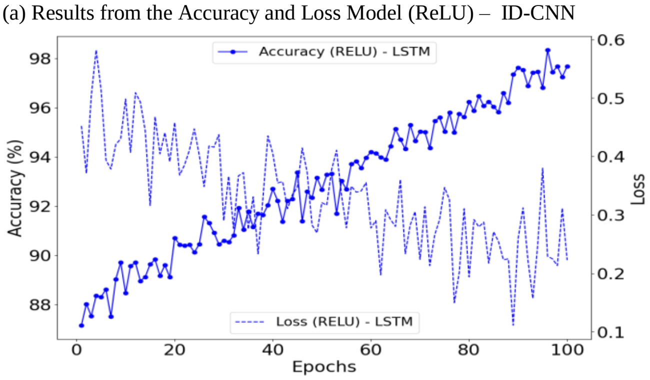
Based on the analysis of the loss and accuracy graphs from various models, each model demonstrates varying capabilities in handling the given prediction task. The following are the training results for accuracy and loss of the IDCNN, LSTM, DCN, ResNet, Ensemble IDCNN + LSTM, and Proposed models, as presented in Fig. 4.



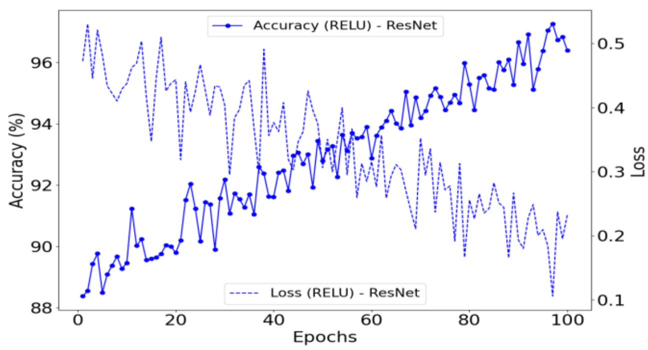
(a) Results from the Accuracy and Loss Model (ReLU) – ID-CNN



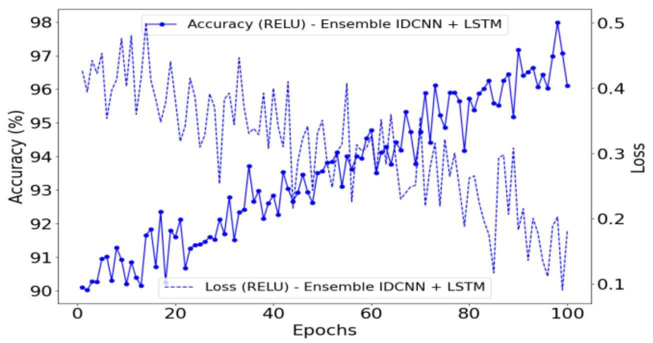
(b) Results from the Accuracy and Loss Model (ReLU) – DCN



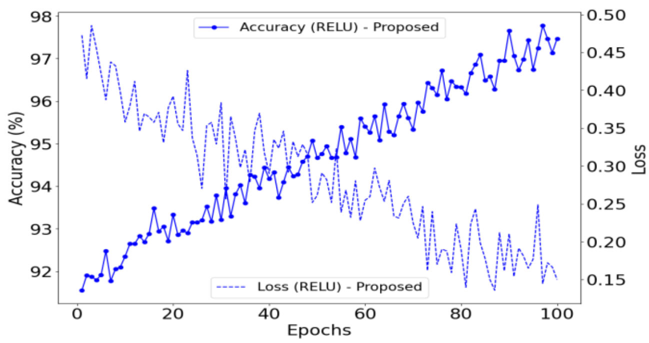
(c) Results from the Accuracy and Loss Model (ReLU) – LSTM



(d) Results from the Accuracy and Loss Model (ReLU) – ResNet



(e) Results from the Accuracy and Loss Model (ReLU) – Ensemble ID-CNN & LSTM



(f) Results from the Accuracy and Loss Model (ReLU) – Proposed

Fig. 4. Comparison of Training Results of All Models (Relu)

The loss graph reveals that all models consistently exhibit a decrease in loss values as the number of epochs increases, indicating that the models are learning from the data and becoming more accurate in their predictions. However, the Dilated Convolutional Network (DCN) model shows higher loss values compared to the other models, especially during the early stages of training. This suggests that the DCN might be less effective at recognizing patterns in the dataset compared to other models like ResNet and LSTM.

ResNet and LSTM, on the other hand, display better performance with lower loss values, suggesting that these models are more adept at handling the complexity of the data. Furthermore, the Ensemble ID-CNN + LSTM model and the proposed model demonstrate the best performance among all, with the lowest loss values. This indicates that the ensemble approach, which combines the strengths of multiple models, offers a significant advantage in reducing prediction errors.

In the accuracy graph, all models show an increase in accuracy as the epochs progress, which reflects their improving prediction capabilities. However, the DCN once again shows a slower increase in accuracy compared to the other models, consistent with its higher loss values. Conversely, the LSTM model shows a rapid improvement in accuracy, reaching higher accuracy levels more quickly than some of the other models. Similar to the loss graph, the Ensemble ID-CNN + LSTM model and the proposed model achieve the highest accuracy, confirming that these models can effectively leverage the strengths of each component to achieve more accurate predictions.

Overall, the ensemble approach has proven to deliver the best performance in terms of both reducing loss and increasing accuracy, demonstrating that combining multiple model architectures can lead to stronger and more accurate predictions. On the other hand, while useful in other contexts, the DCN does not perform as well as the other models on this dataset, both in terms of loss and accuracy. Both the LSTM and ResNet models show solid performance individually, highlighting their ability to capture temporal patterns and handle data complexity effectively.

After the training process of the Convolutional Neural Network (CNN) model is complete, the next step is to conduct an evaluation to assess the model's performance. In this evaluation, we will use several metrics to obtain a comprehensive understanding of the model's performance.

TABLE III
RELU ACTIVATION RESULTS (MSE AND MAE)

Model	Mean Squared Error (MSE)	Mean Absolute Error (MAE)
IDCNN	0.0038	0.0464
LSTM	0.0025	0.0375
DCN	0.0042	0.0497
ResNet	0.0035	0.0450
Ensemble IDCNN + LSTM	0.0023	0,0361
Proposed	**0.0025**	**0.0382**

The results from the Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics for different models using the ReLU activation function reveal significant insights into the models' performance in stock market forecasting tasks (Table III).

As depicted in the bar chart, the Ensemble IDCNN + LSTM model achieves the lowest MSE (0.0023) and MAE (0.0361), making it the top-performing model. This model's ability to combine the strengths of different architectures helps it capture the complexities of stock price movements with greater accuracy. The LSTM model follows closely with an MSE of 0.0025 and an MAE of 0.0375, demonstrating its capability to handle sequential data and detect trends over time.

The Proposed model, with an MSE of 0.0025 and an MAE of 0.0382, performs comparably to the LSTM model. This suggests that the proposed hybrid architecture effectively integrates various CNN features, benefiting from the ReLU activation function's ability to manage non-linearity in the data.

In contrast, models like IDCNN and DCN show higher MSE and MAE values, indicating less accurate performance. For example, DCN has an MSE of 0.0042 and an MAE of 0.0497, reflecting that it struggles more with precise stock market predictions.

TABLE IV
RELU ACTIVATION RESULTS (CONFUSION MATRIX)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
IDCNN	95.31	94.18	96.22	95.19
LSTM	97.92	97.33	98.38	97.85
DCN	96.61	95.74	97.30	96.51
ResNet	96.88	96.26	97.30	96.77
Ensemble IDCNN + LSTM	96.88	95.77	97.84	96.79
Proposed	**97.53**	**96.75**	**98.03**	**97.39**

Table IV provides a detailed comparison of the evaluation metrics using the ReLU activation function. The LSTM model stands out with exceptional performance across all metrics, achieving an accuracy of 97.92%, precision of 97.33%, recall of 98.38%, and an F1 score of 97.85%. These results highlight LSTM's reliability in making accurate predictions and its balanced ability to maintain both precision and recall, making it one of the top-performing models in this analysis.

In contrast, the IDCNN model, while still performing adequately, shows lower values with an accuracy of 95.31%, precision of 94.18%, recall of 96.22%, and an F1 score of 95.19%. These figures suggest that IDCNN may produce more false positives and false negatives compared to the more advanced models, indicating some limitations in handling the complexities of the data. The DCN model performs better than IDCNN but still falls short of the LSTM and ensemble models, with an accuracy of 96.61%, precision of 95.74%, recall of 97.30%, and an F1 score of 96.51%. While these metrics indicate that DCN is more effective than IDCNN, it still faces challenges in achieving the highest level of predictive accuracy and consistency.

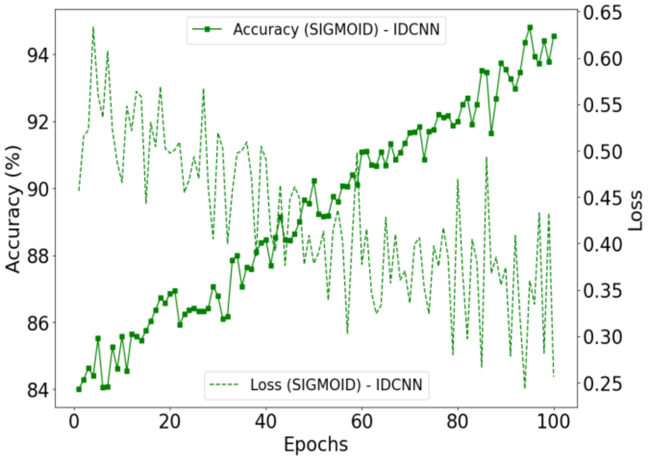
ResNet, with its accuracy of 96.88%, precision of 96.26%, recall of 97.30%, and F1 score of 96.77%, performs slightly better than DCN. This performance reflects the robustness of ResNet's architecture, particularly in addressing issues like vanishing gradients. However, despite its strong performance, ResNet does not surpass LSTM or the ensemble models, suggesting that further optimization could enhance its effectiveness.

The Ensemble IDCNN + LSTM model demonstrates improved performance, combining the strengths of both ndividual models to achieve an accuracy of 96.88%, precision of 95.77%, recall of 97.84%, and an F1 score of 96.79%. This ensemble approach effectively reduces errors and enhances the reliability of predictions, making it a more powerful solution than the individual models alone.

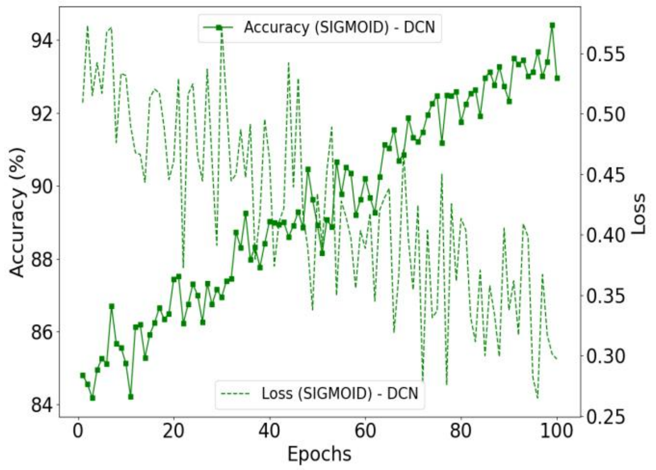
The proposed ensemble model, which integrates ResNet, LSTM, and DCN, delivers the highest overall performance, with an accuracy of 97.53%, precision of 96.75%, recall of 98.03%, and an F1 score of 97.39%. These metrics underscore the effectiveness of this ensemble in balancing the strengths of the included architectures, resulting in high accuracy, consistency, and reliability in predictions.

B. Performance with Sigmoid Activation Function

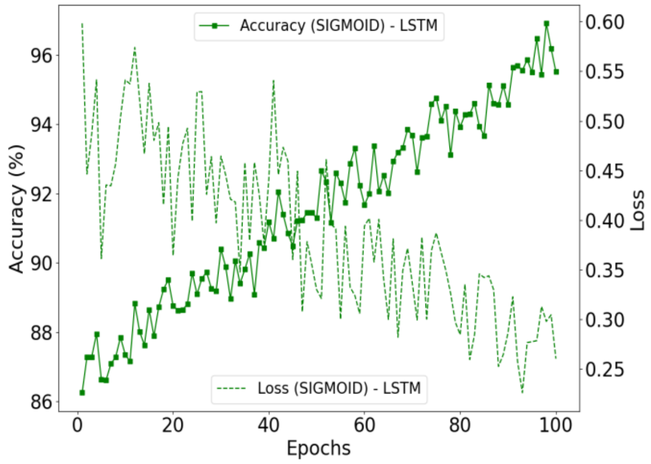
The graph depicting accuracy and loss over several epochs using the Sigmoid activation function provides valuable insights into the performance of the tested models. Based on the analysis of the loss and accuracy graphs from various models, each model shows distinct capabilities in handling the given prediction task. Below are the results for accuracy and loss of the IDCNN, LSTM, DCN, ResNet, Ensemble IDCNN + LSTM, and Proposed models, as presented in Fig 5.



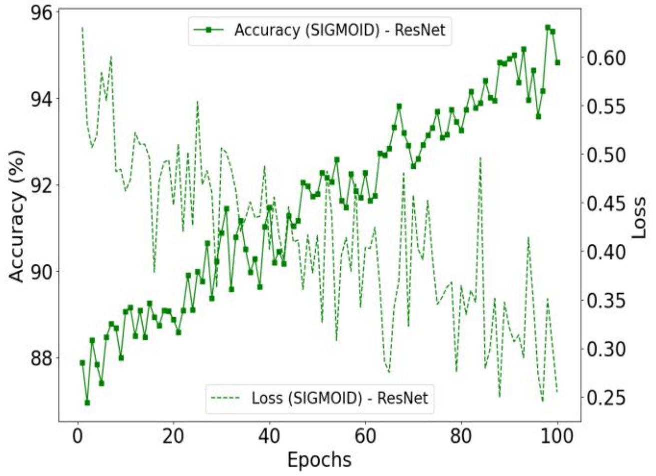
(a) Results from the Accuracy and Loss Model (Sigmoid) – ID-CNN



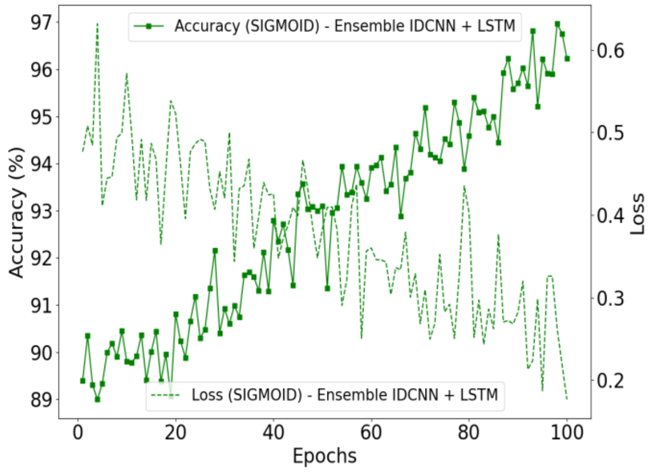
(b) Results from the Accuracy and Loss Model (Sigmoid) – DCN



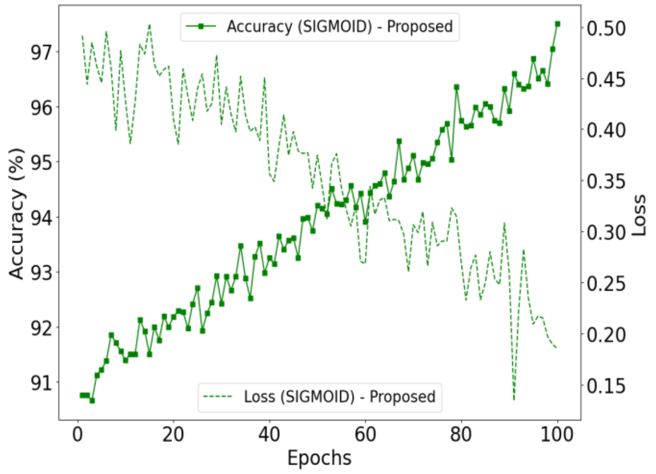
(c) Results from the Accuracy and Loss Model (Sigmoid) – LSTM



(d) Results from the Accuracy and Loss Model (Sigmoid) – ResNet



(e) Results from the Accuracy and Loss Model (Sigmoid) – Ensemble ID-CNN & LSTM



(f) Results from the Accuracy and Loss Model (Sigmoid) – Proposed

Fig. 5. Comparison of Training Results of All Models (Sigmoid)

In the accuracy graph, all models demonstrate a gradual increase in accuracy as training progresses. However, compared to the performance observed with the ReLU activation function, the overall accuracy levels achieved with Sigmoid are noticeably lower. The LSTM model still shows consistent improvement, but its accuracy does not reach the same heights as it did with ReLU, indicating that Sigmoid may not be as effective in enhancing LSTM's predictive capabilities.

The IDCNN model, while improving over time, remains behind other models like ResNet and the ensemble models, showing that it struggles to achieve higher accuracy with Sigmoid. Similarly, the DCN model exhibits moderate improvements but does not reach the accuracy levels achieved with ReLU. The proposed ensemble model, which combines the strengths of multiple architectures, does improve in accuracy but also lags behind its performance with ReLU, highlighting that Sigmoid activation may not be the most suitable choice for maximizing the potential of ensemble methods.

The loss graph further supports these observations, showing that while all models experience a decrease in loss as training progresses, the overall loss values are higher compared to those observed with ReLU. The IDCNN and DCN models, in particular, show slower reductions in loss, suggesting that these models are less effective at minimizing errors when using Sigmoid. The LSTM and ResNet models perform better, but their loss values remain higher than those seen with ReLU, reinforcing the idea that Sigmoid might be less effective in helping these models learn efficiently.

TABLE V
SIGMOID ACTIVATION RESULTS (MSE AND MAE)

Model	Mean Squared Error (MSE)	Mean Absolute Error (MAE)
IDCNN	0.0045	0.0487
LSTM	0.0029	0.0395
DCN	0.0048	0.0502
ResNet	0.0040	0.0473
Ensemble IDCNN + LSTM	0.0027	0.0378
Proposed	**0.0028**	**0.0386**

This section evaluates the Mean Squared Error (MSE) and Mean Absolute Error (MAE) of different models using the Sigmoid activation function. Lower MSE and MAE values indicate better stock price prediction performance (Table V). The Ensemble IDCNN + LSTM model performs best, with the lowest MSE (0.0027) and MAE (0.0378). This shows that the ensemble approach effectively combines the strengths of its architectures, even with Sigmoid, to produce highly accurate predictions. The LSTM model comes next, with an MSE of 0.0029 and an MAE of 0.0395, demonstrating its ability to process sequential data and identify stock market patterns. The Proposed Ensemble model achieves an MSE of 0.0028 and an MAE of 0.0386, slightly higher than the Ensemble IDCNN + LSTM but still better than simpler models like IDCNN and DCN. These simpler models have higher MSE (0.0045 and 0.0048) and MAE (0.0487 and 0.0502), indicating difficulty in handling complex stock market data with Sigmoid activation. Table VI provides a detailed comparison of the models' performance with Sigmoid activation, including key metrics like Accuracy, Precision, Recall, and F1 Score.

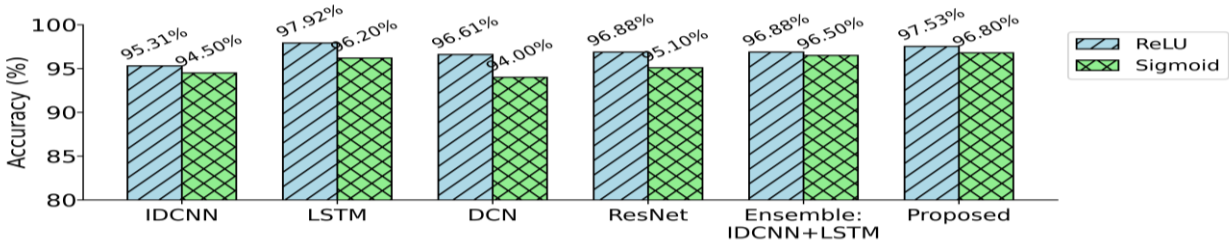
TABLE VI
SIGMOID ACTIVATION RESULTS (CONFUSION MATRIX)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
IDCNN	94.50	93.00	94.20	93.60
LSTM	96.20	95.00	96.80	95.90
DCN	94.00	92.50	93.80	93.15
ResNet	95.10	93.80	95.20	94.50
Ensemble IDCNN + LSTM	96.50	95.30	97.00	96.15
Proposed	**96.80**	**95.70**	**97.30**	**96.50**

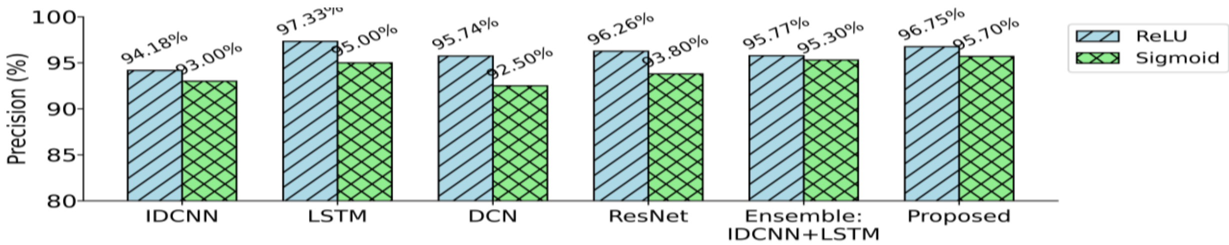
The LSTM model demonstrates strong performance with an accuracy of 96.20%, precision of 95.00%, recall of 96.80%, and an F1 score of 95.90%. Although slightly lower than the results obtained using ReLU, the LSTM model remains reliable with Sigmoid, maintaining a good balance between precision and recall. The IDCNN model, on the other hand, shows lower performance with an accuracy of 94.50%, precision of 93.00%, recall of 94.20%, and an F1 score of 93.60%. These results indicate that IDCNN struggles to maintain high precision and recall when using Sigmoid. Its lower F1 score reflects difficulties in consistently identifying true positives while minimizing false positives. The DCN model delivers the lowest performance, with an accuracy of 94.00%, precision of 92.50%, recall of 93.80%, and an F1 score of 93.15%. This confirms that DCN is less effective with Sigmoid, resulting in lower accuracy and reliability in its predictions. The ResNet model outperforms IDCNN and DCN, achieving an accuracy of 95.10%, precision of 93.80%, recall of 95.20%, and an F1 score of 94.50%. However, its performance is still slightly below that of ReLU, suggesting that Sigmoid limits ResNet's full potential. The higher F1 score compared to IDCNN and DCN shows that ResNet is better at balancing precision and recall, though it does not achieve peak performance. The Ensemble IDCNN + LSTM model demonstrates significant improvements over individual models, with an accuracy of 96.50%, precision of 95.30%, recall of 97.00%, and an F1 score of 96.15%. This highlights the effectiveness of ensemble approaches, which combine the strengths of multiple models to achieve better overall performance. The higher F1 score indicates the ensemble's ability to balance precision and recall, reducing the likelihood of false positives and false negatives. The proposed ensemble model, which integrates ResNet, LSTM, and DCN, achieves the highest overall performance with an accuracy of 96.80%, precision of 95.70%, recall of 97.30%, and an F1 score of 96.50%. These results show that the ensemble successfully combines the strengths of its components, delivering superior accuracy, consistency, and reliability in classification tasks. The high F1 score highlights its ability to maintain an excellent balance between precision and recall, making it the most effective model tested..

C. *Discussion on Activation Functions*

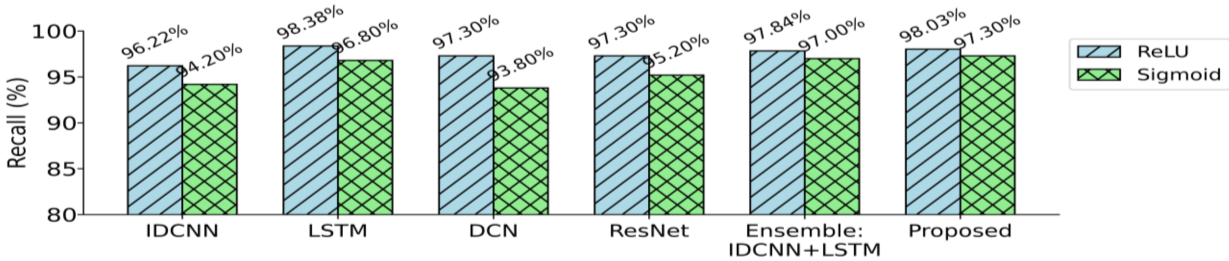
This study compared two key activation functions—ReLU and Sigmoid—in CNN architectures aimed at improving stock market forecasting. The choice of activation function is crucial as it directly affects training efficiency and the overall performance of the model.



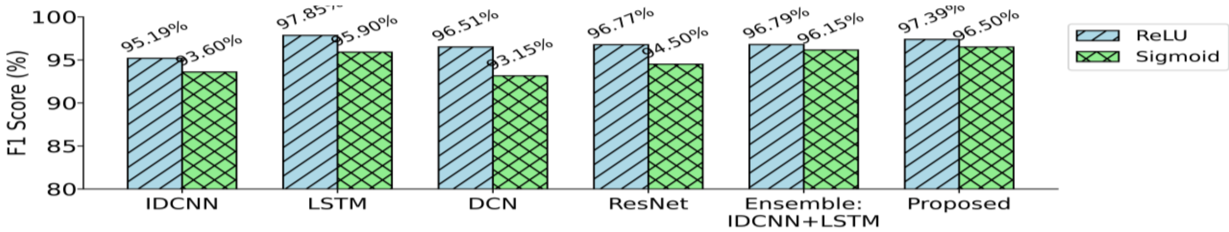
(a) ReLU vs Sigmoid - Accuracy



(b) ReLU vs Sigmoid – Precision



(c) ReLU vs Sigmoid – Recall



(d) ReLU vs Sigmoid - F1 Score

Fig. 6. Comparison Models (Relu Vs Sigmoid)

Based on the four main evaluation metrics—Accuracy, Precision, Recall, and F1 Score—shown in Figure 6, the ReLU activation function consistently outperforms Sigmoid, especially in complex models like the Proposed Ensemble model. This ensemble combines multiple methods, such as CNN and LSTM, to maximize their strengths.

In Figure 6 (a), which compares accuracy, all models using ReLU achieve higher accuracy than those using Sigmoid. This indicates that ReLU is better at identifying patterns in complex stock market data, which is often influenced by external factors. ReLU also avoids the vanishing gradient problem, a common issue with Sigmoid in deeper models, making it more effective for these architectures.

Figure 6 (b) highlights ReLU's advantage in Precision, where it outperforms Sigmoid in nearly all models. This is especially evident in LSTM and the Proposed Ensemble model, where ReLU reduces prediction errors and produces more accurate forecasts. This makes ReLU-based models better suited for precise predictions, which are crucial in stock market decision-making.

In Figure 6 (c), the comparison of Recall shows that ReLU is more effective at detecting important trends in the data. Higher Recall means models can identify more relevant patterns, making them more reliable for short-term predictions and handling market volatility. ReLU's non-linear properties allow for faster learning, while Sigmoid often struggles with gradient saturation.

Finally, Figure 6 (d) compares F1 Scores, which combine Precision and Recall. ReLU provides a better balance by maintaining high precision while capturing patterns that Sigmoid-based models might miss. Higher F1 Scores in models like the Proposed Ensemble and LSTM demonstrate that ReLU delivers more stable and optimal performance, especially in volatile data like stock prices.

The comparison of Mean Squared Error (MSE) and Mean Absolute Error (MAE), shown in Figure 7, also confirms ReLU's superiority. In the best-performing model, Ensemble IDCNN + LSTM, ReLU achieves the lowest MSE (0.0023) and MAE (0.0361), showing its ability to model complex stock market patterns with high accuracy. LSTM and the Proposed models also perform well with ReLU, further demonstrating its effectiveness in handling the non-linearity of sequential data like stock prices.

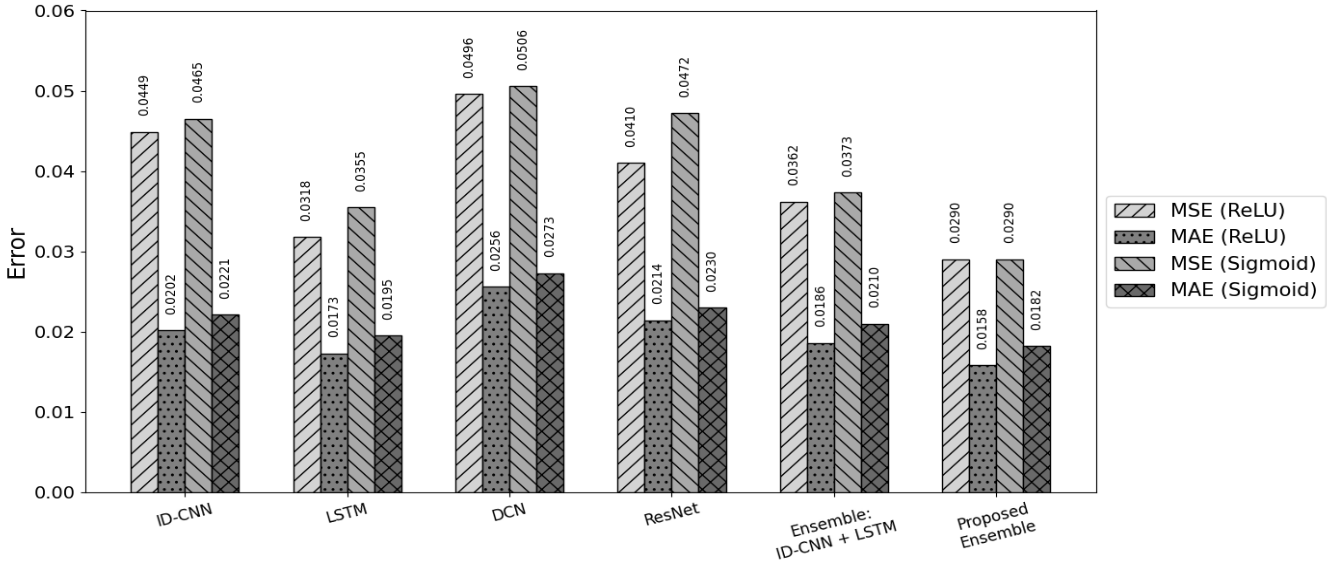


Fig. 7. MSE and MAE Comparison for ReLU and Sigmoid Activation

In contrast, when using the Sigmoid activation function, while the Ensemble IDCNN + LSTM model still performs the best, it shows slightly higher error rates (MSE: 0.0027, MAE: 0.0378) compared to its performance with ReLU. The LSTM and Proposed models also perform well with Sigmoid, though not as strongly as they do with ReLU. Simpler models like IDCNN and DCN exhibit significantly higher error rates when using Sigmoid (MSE: 0.0045 and 0.0048, MAE: 0.0487 and 0.0502), suggesting that Sigmoid struggles with the generalization needed to accurately predict stock market movements. Overall, the comparison clearly shows that ReLU is a better choice than Sigmoid for stock market forecasting tasks.

V. Conclusions

This study has demonstrated that the choice of activation function plays a pivotal role in improving the performance of deep learning models, particularly in the complex task of stock market forecasting. By conducting a comparative analysis of ReLU and Sigmoid activation functions across various CNN-based architectures, it is evident that ReLU consistently outperforms Sigmoid in key metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE). Models utilizing ReLU, especially the Ensemble IDCNN + LSTM and LSTM, exhibited significantly lower error rates, showcasing their ability to accurately capture non-linear patterns inherent in stock market data. The superior performance of ReLU can be attributed to its capacity to mitigate the vanishing gradient problem, thereby enabling deeper networks to learn more effectively from large and complex datasets. In contrast, models using Sigmoid showed higher error rates, particularly in simpler architectures like IDCNN and DCN, indicating that Sigmoid may struggle to generalize effectively in tasks requiring more nuanced pattern recognition. ReLU has proven to be a more suitable activation function for enhancing model accuracy and reliability in stock market predictions. It is therefore recommended that future stock market forecasting models prioritize ReLU over other activation functions to better handle the dynamic and volatile nature of financial data.

References

[1] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. C. Khatri, "Predicting stock market index using LSTM," *Mach. Learn. with Appl.*, vol. 9, no. February, p. 100320, 2022, doi: 10.1016/j.mlwa.2022.100320.

[2] F. Petropoulos *et al.*, "Forecasting: theory and practice," *Int. J. Forecast.*, vol. 38, no. 3, pp. 705–871, 2022, doi: 10.1016/j.ijforecast.2021.11.001.

[3] B. N. Varaprasad, C. Kundan Kanth, G. Jeevan, and Y. K. Chakravarti, "Stock Price Prediction using Machine Learning," *Proc. Int. Conf. Electron. Renew. Syst. ICEARS 2022*, pp. 1309–1313, 2022, doi: 10.1109/ICEARS53579.2022.9752248.

[4] K. Liu, J. Zhou, and D. Dong, "Improving stock price prediction using the long short-term memory model combined with online social networks," *J. Behav. Exp. Financ.*, vol. 30, p. 100507, 2021, doi: 10.1016/j.jbef.2021.100507.

[5] P. Lv, Q. Wu, J. Xu, and Y. Shu, "Stock Index Prediction Based on Time Series Decomposition and Hybrid Model," *Entropy*, vol. 24, no. 2, 2022, doi: 10.3390/e24020146.

[6] Y. Yuhandri, A. P. Windarto, and M. N. H. Siregar, "Improving Brain Tumor Classification Efficacy through the Application of Feature Selection and Ensemble Classifiers," *J. Image Graph. Kingdom)*, vol. 11, no. 4, pp. 397–404, 2023, doi: 10.18178/joig.11.4.397-404.

[7] D. Ruan, J. Wang, J. Yan, and C. Gühmann, "CNN parameter design based on fault signal analysis and its application in bearing fault diagnosis," *Adv. Eng. Informatics*, vol. 55, no. June 2022, p. 101877, 2023, doi: 10.1016/j.aei.2023.101877.

[8] W. Dai, "Application of Improved Convolution Neural Network in Financial Forecasting," *J. Organ. End User Comput.*, vol. 34, no. 3, pp. 1–16, 2021, doi: 10.4018/joeuc.289222.

[9] S. Chen and H. He, "Stock Prediction Using Convolutional Neural Network," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 435, no. 1, 2018, doi: 10.1088/1757-899X/435/1/012026.

[10] Y. Komori, "Convolutional Neural Network for Stock Price Prediction Using Transfer Learning," *SSRN Electron. J.*, no. 2018, pp. 1–8, 2021, doi: 10.2139/ssrn.3756702.

[11] C. S. Ku, J. Xiong, Y. L. Chen, S. D. Cheah, H. C. Soong, and L. Y. Por, "Improving Stock Market Predictions: An Equity Forecasting Scanner Using Long Short-Term Memory Method with Dynamic Indicators for Malaysia Stock Market," *Mathematics*, vol. 11, no. 11, 2023, doi: 10.3390/math11112470.

[12] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/6622927.

[13] L. Sayavong, Z. Wu, and S. Chalita, "Research on stock price prediction method based on convolutional neural network," *Proc. - 2019 Int. Conf. Virtual Real. Intell. Syst. ICVRIS 2019*, no. Dl, pp. 173–176, 2019, doi: 10.1109/ICVRIS.2019.00050.

[14] A. Staffini, "Stock Price Forecasting by a Deep Convolutional Generative Adversarial Network," *Front. Artif. Intell.*, vol. 5, no. February, pp. 1–16, 2022, doi: 10.3389/frai.2022.837596.

[15] R. B. Wiranata and A. Djunaidy, "The Stock Exchange Prediction using Machine Learning Techniques: A Comprehensive and

Systematic Literature Review," *J. Ilmu Komput. dan Inf.*, vol. 14, no. 2, pp. 91–112, 2021, doi: 10.21609/jiki.v14i2.935.

[16] H. S. Sim, H. I. Kim, and J. J. Ahn, "Is Deep Learning for Image Recognition Applicable to Stock Market Prediction?," *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/4324878.

[17] R. Segura, "La Investigación en Gestión Pública: un modelo para el abordaje metodológico," *Rev. Centroam. Adm. Pública*, no. 73, pp. 10–24, 2017, doi: 10.35485/rcap73_1.

[18] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," *Appl. Sci.*, vol. 12, no. 18, pp. 1–43, 2022, doi: 10.3390/app12188972.

[19] S. I. Ao and H. Fayek, "Continual Deep Learning for Time Series Modeling," *Sensors*, vol. 23, no. 16, pp. 1–32, 2023, doi: 10.3390/s23167167.

[20] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 2, pp. 757–774, 2023, doi: https://doi.org/10.1016/j.jksuci.2023.01.014.

[21] J. M. Montgomery, F. M. Hollenbach, and M. D. Ward, "Improving predictions using ensemble bayesian model averaging," *Polit. Anal.*, vol. 20, no. 3, pp. 271–291, 2012, doi: 10.1093/pan/mps002.