Stock Price Prediction Based on Deep Learning

Xuejiao Chen
School of Mathematics, The University of Sydney, Sydney 2006, Australia

Keywords: Deep Learning, Time Series Forecasting, Stock.

Abstract: Financial time series forecasting stands as a cornerstone in investment decision-making and risk management. Nonetheless, traditional statistical models often grapple with capturing intricate nonlinear patterns and enduring dependencies within data. To enhance prediction accuracy, this study delves into the feasibility of employing deep learning technology in financial time series forecasting. 5 deep learning models have been constructed, containing deep multi-layer perceptron (DMLP), convolutional neural networks (CNN), long short-term memory networks (LSTM), recurrent neural networks (RNN), and auto-encoders (AE), leveraging real transaction market data to forecast log returns. Through empirical comparison, we ascertain that the CNN model excels in harnessing data features, outperforming other models in prediction accuracy. Nevertheless, AE models exhibit the poorest performance in this task, attributed to their deficiency in modeling time dependencies. Overall, this study validates the possible usefulness for predicting financial time series data and furnishes valuable insights for future research endeavors.

1 INTRODUCTION

With the rapid development of economic globalization, the dynamics and complexity of financial markets are increasing, making the prediction of financial product prices and their volatility a key and challenging issue. This study focuses on constructing effective financial time series forecasting models. Financial time series differ from typical time series in that they often exhibit complex characteristics such as non-linearity, non-nationality, and high auto-correlation. These characteristics limit the effectiveness of traditional forecasting models like ARMA and GARCH in practice, as these models often rely on assumptions of linearity and stationarity, which do not capture the true dynamics of financial markets.

With the development of artificial intelligence technology, machine learning methods such as Support Vector Machines (SVM), Random Forests, and Gradient Boosting Trees have been used to predict financial time series data and have been compared for accuracy with traditional methods, and efficiently handle bias and variance in time series data (Jiang, 2021). Nevertheless, these preliminary machine learning technologies also show certain limitations, especially in handling high-dimensional data and over-fitting issues. Moreover, these methods often overlook the auto-correlation characteristic of time series.

To overcome these limitations, this study introduces deep learning methods, which are extensions of machine learning algorithms inspired by the human brain and utilize multi-layer neural networks to simulate decision-making processes. In this experiment, we leverage their strong non-linear modeling capabilities to address the complexities of financial time series. Deep learning models, such as DMLP, CNN, LSTM, RNN and AE have proven effective in various domains. These models are better at capturing the auto-correlation of time series and addressing non-nationality issues (Neagoe et al., 2018).

The main contributions of this paper are as follows: First, we systematically compare various deep learning models in the prediction of financial time series; second, we discuss these models' effectiveness in handling high-dimensional data and preventing over-fitting. The structure of the paper is organized as follows: we begin with an introduction to the research background and problem statement, then detail the experimental methods and the deep learning models used, followed by reporting and analyzing the experimental results, and conclude with a summary of findings and future research directions.

Chen, X.
Stock Price Prediction Based on Deep Learning.
DOI: 10.5220/0013004900004601

In *Proceedings of the 1st International Conference on Innovations in Applied Mathematics, Physics and Astronomy (IAMPA 2024)*, pages 151-159
ISBN: 978-989-758-722-1

2 RELATED WORK

Initially, stock price predictions relied primarily on mathematical models and statistical methods. Ding et al. (2010) researched the volatility of financial products using SV, GARCH, and EWMA models. The findings indicated that the EWMA model excelled in volatility prediction, whereas the SV model outperformed GARCH when volatility was highly variable and random (Ding and Meade, 2010). Wahyudi et al. (2017) found that the ARIMA model has strong short-term predictive capabilities, effectively competing with existing stock price prediction technologies (Wahyudi, 2017). Rouf et al. (2021) compared the application of SVM and neural networks in forecasting financial time series. The study discovered that although SVM is a commonly used prediction method, it is slow in forecasting, whereas simple deep learning methods are unable to make accurate predictions due to randomness issues (Rouf et al, 2021).

Lu (2024) addressed the limitations of RNN and LSTM by developing an efficient Time-Series Recurrent Neural Network (TRNN), which compresses time series data to enhance the accuracy of stock price predictions (Lu and Xu, 2024). Zaheer et al. (2023) experimented with the Shanghai Composite Index (000001) and mixed existing deep learning methods, comparing models such as CNN-RNN and CNN-LSTM. The results showed that CNN-RNN performed best among these hybrid models (Zaheer et al., 2023). Fang et al. (2023) proposed an improved LSTM-based model, incorporating a cross-entropy loss function and an adaptive network mechanism, capable of making precise predictions in time series with significant long-term volatility (Fang et al., 2023). By using genetic algorithm to optimize parameters of RNN, Al et al. (2023) successfully improved the prediction performance of the model and found the most suitable parameter configuration (Haromainy et al., 2023). Masini et al. (2023) investigates the application of supervised machine learning techniques to stock price prediction, conducts in-depth analysis of linear models, especially regularization models such as ridge regression, and also explores nonlinear models and integration methods including random forests (Masini et al., 2023).

3 METHOD

3.1 DMLP

MLP is the most traditional form of DNN (Wang, Yan and Oates, 2017), and DMLP is one type of advanced MLP with more hidden layer, which can handle both linearly separable and nonlinear separable data.

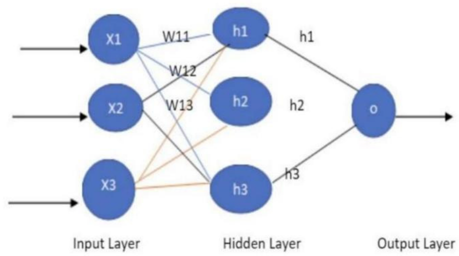

Figure 1: DMLP structure (Sutskever et al., 2013).

Hidden layer, as the most important part of the model, comprises four hidden layers, each with 64 neurons and utilizing the ReLU activation function. The output layer ultimately produces the final prediction, with a single node for regression problems, generating continuous value predictions such as Log_return (Figure 1). The DMLP model employs back-propagation for learning, propagating errors from the output layer neurons back to the hidden layer to iteratively optimize the algorithm of the model, specifically the weights of the connections between layers (Sutskever et al., 2013).

3.2 RNN

RNN is meticulously crafted to handle sequential data effectively. Unlike DMLP, RNN is capable of processing sequences of arbitrary length, with hidden layers passing information between different points in time (Freeborough and Van, 2022). The input layer adjusts log_return feature values to sequence data in (sample, time_steps, function) format for processing by the RNN. The hidden layer, which consists of two loop layers, each layer uses 64 SimpleRNN units (Figure 2). When receiving new data at each time step, the current log_return is weighted with the previous hidden state and the hidden layer state is updated through the ReLU activation function. The output layer integrates information from all points in time to produce the final predicted value.

At every time step h_t:

① Obtain the input at present point of time x_t and the latent representation at the previous point of time h_{t-1} .

② Compute the updated latent representation at present point of time h_t:

$$h_t = f(W \cdot x_t + U \cdot h_{t-1} + b) \qquad (1)$$

- W and U represent the parametric transformation matrix that map the input and the previous latent state, respectively, onto the current latent state.
- b represents a bias.
- f represents the activation function, such as ReLU, tanh, etc.

③ Generate the output of at present y_t.

$$y_t = V \cdot h_t + c \qquad (2)$$

- V represents the parametric transformation matrix from the previous latent state to the output.
- c represents a bias (Yan & Ouyang 2018).

3.3 LSTM

Common RNN is prone to gradient vanishing or gradient explosions when dealing with long sequences, which limits their ability to learn long-distance dependencies. To solve this problem, we introduce the LSTM (Sherstinsky, 2020), which features an intricate internal structure composed of four primary interactive elements: a unit cell state (responsible for long-term memory storage) and three gates that regulate the flow of information (input gate, forget gate, and output gate). This design enables the LSTM to selectively introduce or discard information from the cell state as necessary, thereby effectively preserving and updating long-term memory representations (Figure 3).

Hidden layer:
The following experiment uses 4 hidden layers, and the structure of each layer is as follows:
Cell state: The most important part of LSTM, allowing the network to pass and maintain long-term information.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot c_t \qquad (3)$$

Input gate: regulates which aspects of the newly introduced log_return data should be incorporated into the cell state. Its functionality is governed by sigmoid activation functions.

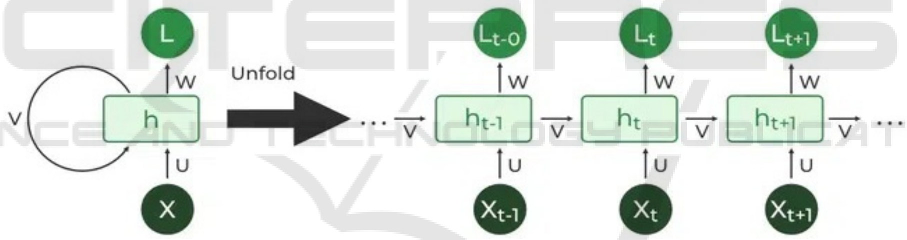

Figure 2: RNN structure (Introduction to recurrent neural network).

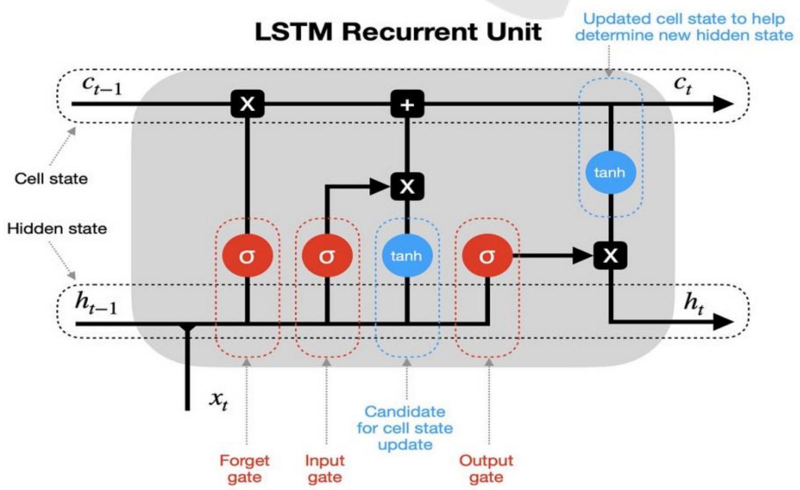

Figure 3. LSTM structure (LSTM Recurrent Neural Networks)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$c_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (5)$$

W_i and W_c represent the parametric transformation matrix that map the input and the previous cell state onto the input gate and candidate cell state representations, respectively. b_i and b_c are the corresponding biases.

Forget gate: The activation function is used to determine the information in the log_return unit state that needs to be forgotten or discarded.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

σ represents the sigmoid activation function. W_f represents the parametric transformation matrix of the forgetting gate. b_f represents a bias (Yan and Ouyang, 2020).

3.4 CNN

CNN is unique in their layer type and structure and are designed to process data with clearly defined grid patterns, such as images. The synergistic combination of convolutional layers and pooling layers within the CNN enables the efficient extraction of localized patterns and global information from the raw time series data, ultimately facilitating the final prediction task (Mehtab and Sen, 2021).

The convolution layer learns multiple convolution cores from input data log_return through convolution operations, and uses 64 filters to extract local feature patterns (Figure 4).

The activation layer employs the ReLU activation function, thereby introducing non-linear transformations to the data.

The pooling layer down-samples the features output by the convolutional layer, reducing the spatial dimension and the number of parameters while preserving the most significant local features. Then, before the fully connected layers, the high-dimensional data obtained from the convolution layer and the pooling layer are flattened into a one-dimensional vector, all features are integrated, and the regression predicted value log_return is finally output.

$$output = ReLU(W \cdot input + b) \quad (8)$$

3.5 Auto-Encoder(AE)

The goal of the AE is to reconstruct itself from the input data, that is, it attempts to map the input data to itself, predicting the data through this process (Figure 5). AE usually consists of encoder and decoder two parts.

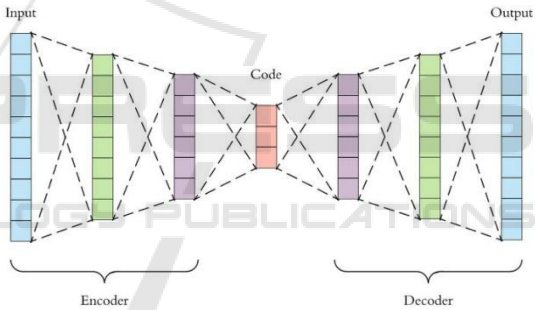

Figure 5. AE structure (Applied Deep Learning - Part 3).

$$output[i] = \sum_k kernel[k] \times input[i + k]) + bias \quad (7)$$

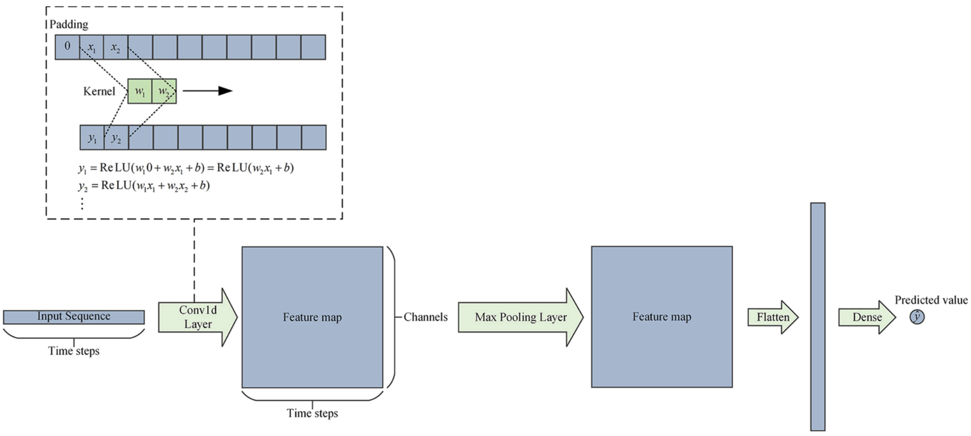

Figure 4. CNN structure (Zhang et al., 2024)

The primary function of an encoder is to compress high-dimensional input features into a lower-dimensional encoded representation through multiple hidden layers. In this particular model, the encoder employs a fully connected layer with a ReLU activation function to compress and encode input features into a 32-dimensional representation.

Conversely, the decoder undertakes the task to reconstruct the primordial high-dimensional input features from the low-dimensional encoded representation. It shares a similar structure to the encoder but with the arrangement of hidden layers and neurons reversed. In this model, the decoder utilizes a linearly activated fully connected layer that generates vectors with the same feature dimension as the original input, with the objective of reconstructing the input data (Sezer et al., 2020).

4 EXPERIMENT

Real life financial market data tends to be highly uncertain and complex. Traditional statistical models, while widely used in finance, may have limitations when dealing with these properties. The aim of the experiment is to evaluate these models (DMLP, RNN, LSTM, CNN, AE) on the logarithmic return of financial derivatives in the actual trading market.

Data set

The data set used in this experiment comes from Kaggle and contains relevant stock market data on the actual execution of financial markets. The data set is recorded in seconds, carefully reflecting the rapid changes in real financial markets. This data set is used to predict future log_returns to better reflect stock price movements.

Firstly, calculate the logarithmic return rate of each successive time point for each different time_id in the data set, analyze the market dynamics of each time period, and do further time series analysis.

Logarithmic return for each interval is calculated using:

$$log\_return = log(\frac{Price\ at\ t+1}{Price\ at\ t}) \tag{9}$$

4.1 Evaluation Indicators

To assess the model's predictive accuracy, this study employed three widely adopted regression performance metrics: MSE, RMSE, and MAE. These measures facilitated a quantitative comparison between the model's predicted values and the actual observed values obtained from the experimental data (Figure 6, figure 7 and figure 8).

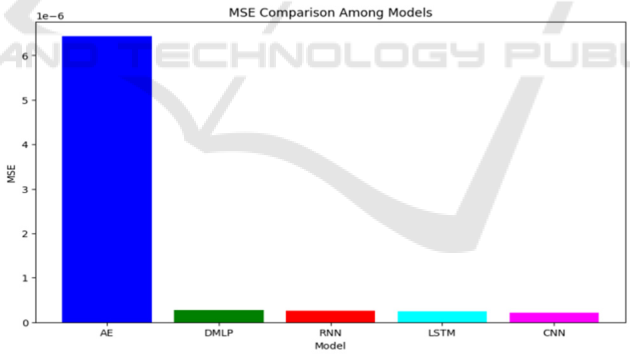

Figure 6. MSE Comparison among models (Picture credit: Original).

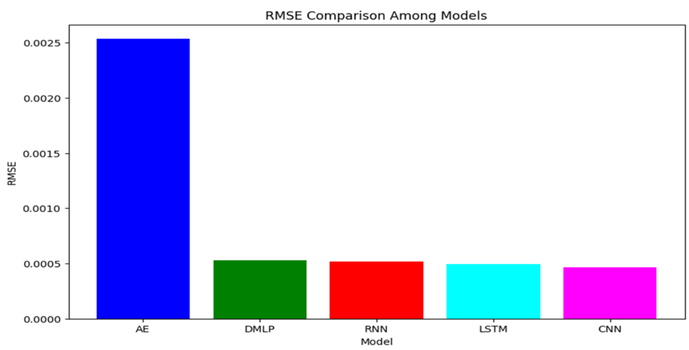

Figure 7. RMSE Comparison among models (Picture credit: Original).

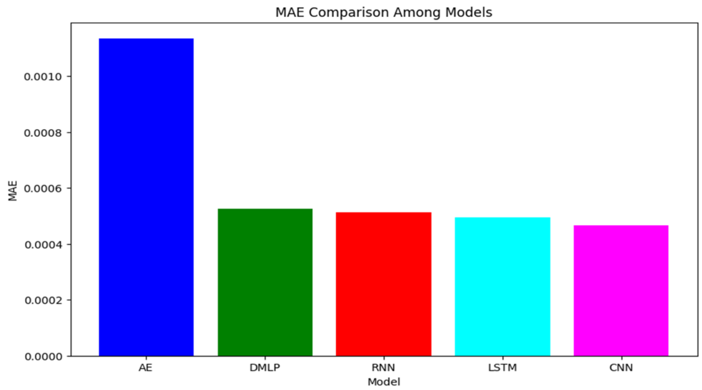

Figure 8. MAE Comparison among models (Picture credit: Original).

Table 1. Indicators comparison.

Method	MSE	RMSE	MAE
DMLP	2.995e-07	0.000547	0.000544
RNN	2.897e-07	0.000538	0.000537
LSTM	2.534e-07	0.000503	0.000503
CNN	2.416e-07	0.000492	0.000489
AE	0.941	0.969998	0.413337

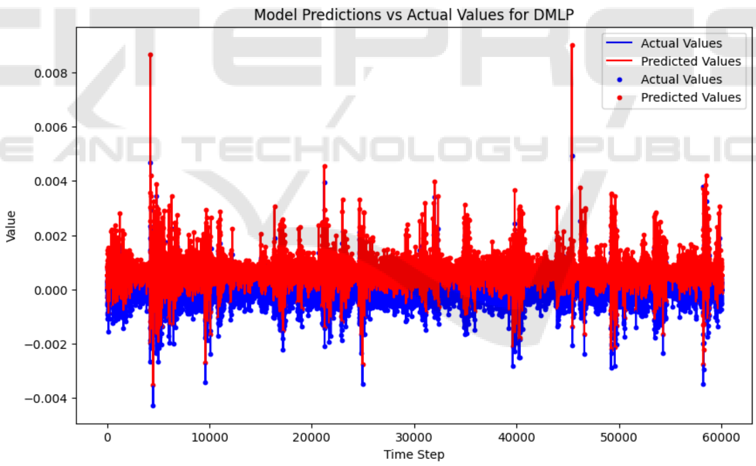

Figure 9. Prediction vs Actual values plots for DMLP (Picture credit: Original).

According to the above experimental results (Table 1), AE has the worst performance and CNN has the best three indicators, although there is little difference in the indicators of DMLP, RNN, LSTM and CNN. Therefore, the comprehensive performance of CNN model is the best, and the comprehensive performance of AE model is the worst. This shows that the CNN model can fit and predict the log_return more accurately, and can capture the characteristics and rules of the data, while the performance of AE model is relatively poor.

4.2 Prediction Vs Actual Values Plots

In addition to the above indicators that can evaluate the performance of the model, Prediction vs Actual values Plots can also help observe whether Log_return fluctuations have been captured.

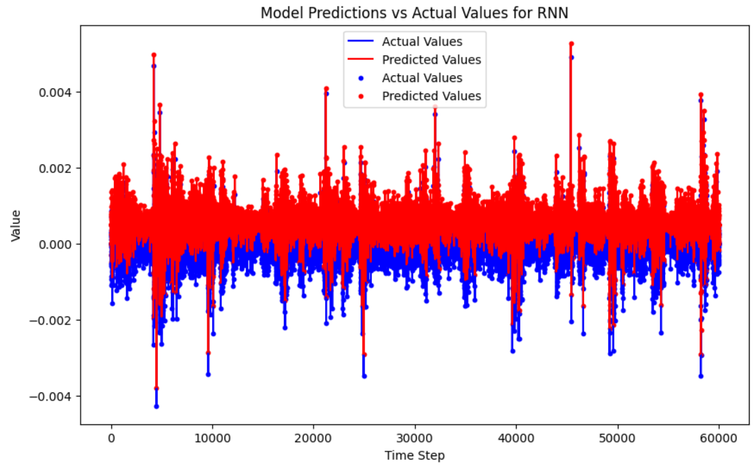

Figure 10. Prediction vs Actual values plots for RNN(Picture credit: Original).

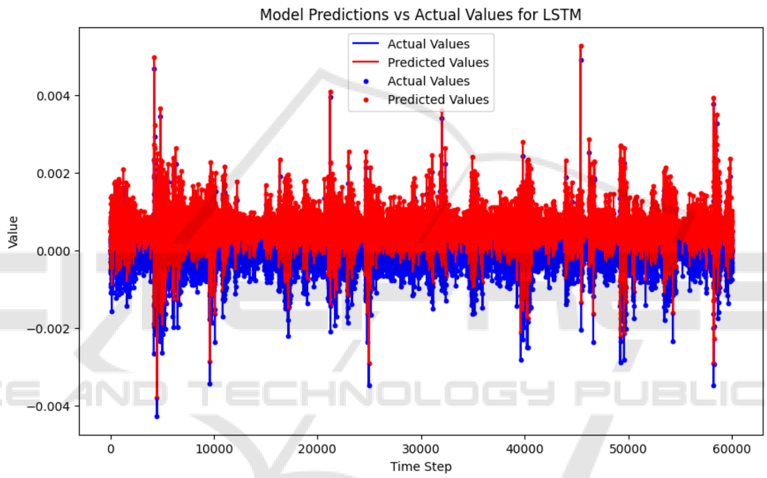

Figure 11. Prediction vs Actual values plots for LSTM (Picture credit: Original).

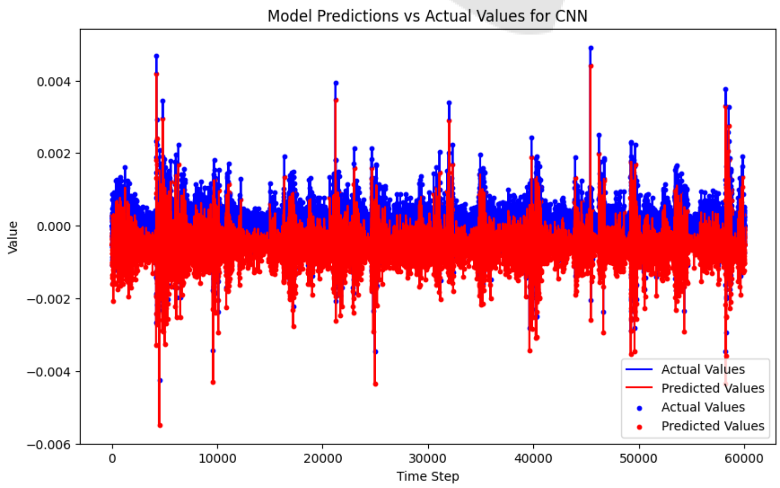

Figure 12. Prediction vs Actual values plots for CNN (Picture credit: Original).

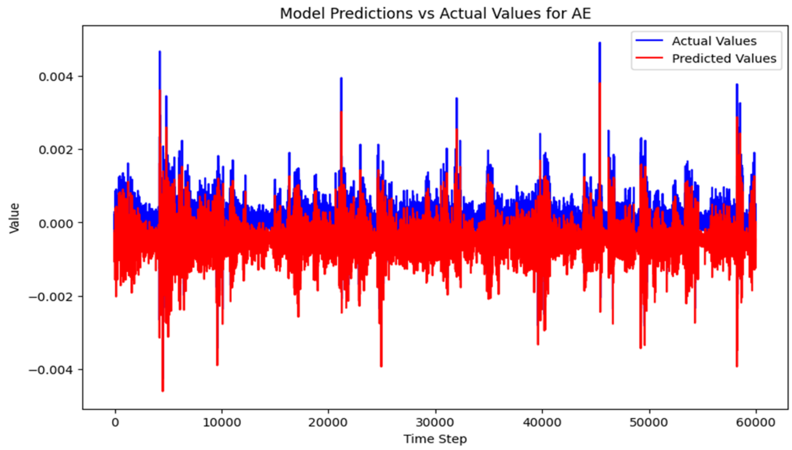

Figure 13. Prediction vs Actual values plots for AE (Picture credit: Original).

These 5 deep learning models exhibit distinct characteristics in forecasting financial time series. The DMLP model shows significant prediction discrepancies due to its inability to effectively model long-term dependencies (Figure 9). The RNN model improves upon the DMLP but still exhibits large deviations, constrained by gradient vanishing or exploding issues (Figure 10). In contrast, the LSTM model (Figure 11), by incorporating gating mechanisms and memory units, captures the long-range interdependence inherent in time-series data, culminating in more precise prognostications. The CNN model can efficiently capture localized features and mitigate noise, outperforming LSTM. Conversely, the Acoustic Emission (AE) model performs poorly in time series prediction tasks as it is designed to learn a compressed representation of data only (Figure 12). In conclusion, for financial time series prediction tasks, CNN and LSTM models can more effectively capture features and dependencies, yet further exploration and improvements are needed to enhance accuracy and stability in deep learning models (Figure 13).

Therefore, different deep learning models show obvious differences in dealing with financial time series prediction due to their differences in structure and principle. Compared with other models, CNN model and LSTM model can capture the features and dependencies of time series data more effectively, so they achieve better performance in this task. However, deep learning models still need to be further explored and improved to improve the accuracy and stability of financial time series predictions.

4.3 Discussion

Mainly analyse the reasons for the above results and the differences between CNN and AE models, and review other models.

CNN is good at capturing local features and can effectively extract important features from input data. This is very helpful for problems that require analysis and prediction of time series data. By sliding over the input data through the convolution kernel, the CNN model can identify and extract important features to maximize the use of the structural information of the input data. The CNN model has strong generalization ability and can better fit the complex data distribution, resulting in more accurate predicting outcomes. AE model aims to unveil the latent feature representations inherent within the input data to enable compression and reconstruction of the data. However, in this task of time series prediction, simply learning the feature representation of the data may not adequately capture important information such as time dependence in the data.

The DMLP model demonstrates inadequacy in capturing persistent dependencies among time series data. Conversely, the RNN model introduces loop joins to model sequence data, improving predictive performance. However, issues like disappearing gradients or explosions constrain its full potential. The LSTM model effectively addresses the gradient problem of RNN for long series, yielding more accurate predictions with consistent trends. Consequently, AE and other models may underperform compared to CNN in this task. CNN's structure and characteristics make them well-suited for solving time series forecasting challenges, leveraging spatio-temporal features for optimal performance in this domain.

5 CONCLUSION

This study investigates the performance comparison of various deep learning models by analyzing real trading market data. The log_return was employed as the target feature, and multiple deep neural network algorithms, including DMLP, CNN, LSTM, RNN, and AE, were constructed to predict the log_return. The obtained results were comprehensively discussed, and experimental evaluations were conducted. Through rigorous experimental comparisons, it was discovered that the CNN model structure and features are better suited for processing time series prediction tasks. This is because the CNN model can effectively capture and utilize the inherent characteristics of the data, enabling it to outperform other models in this specific application. However, AE and other models exhibited relatively poor performance due to its lack of capability in modeling time dependencies present in the data. Moving forward, future research endeavors could focus on further exploring the potential application of improved AE models in the realm of time series prediction. Alternatively, efforts could be directed towards developing more sophisticated and efficient models to enhance the preciseness and efficiency of predicting financial time series. Such advancements would contribute to unlocking the full potential of deep learning techniques in this crucial domain.

REFERENCES

Jiang, W. Applications of deep learning in stock market prediction: recent progress. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery , 2021, 184, 115537.

Neagoe, V. E., Ciotec, A. D., & Cucu, G. S. Deep convolutional Neural Networks versus multilayer perceptron for financial prediction. In 2018 International Conf. on Communications.2018, pp. 201-206.

Ding, J. and Meade, N. ‘Forecasting accuracy of stochastic volatility, GARCH and EWMA models under different volatility scenarios’, Applied Financial Economics, 2010, 20(10), pp. 771–783.

Wahyudi, S. T. The ARIMA Model for the Indonesia Stock Price. International Journal of Economics & Management, 2017, 11.

Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., & Kim, H. C. Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. Electronics, 2021, 10(21), 2717.

Lu, M., & Xu, X. TRNN: An efficient time-series recurrent neural network for stock price prediction. Information Sciences, 2024, 657, 119951.

Zaheer, S., Anjum, N., Hussain, S., Algarni, A. D., Iqbal, J., Bourouis, S., & Ullah, S. S. A multi parameter forecasting for stock time series data using LSTM and deep learning model. Mathematics, 2023, 11(3), 590.

Fang, Z., Ma, X., Pan, H., Yang, G., & Arce, G. R. Movement forecasting of financial time series based on adaptive LSTM-BN network. Expert Systems with Applications, 2023, 213, 119207.

Al Haromainy, M. M., Prasetya, D. A., & Sari, A. P. Improving Performance of RNN-Based Models With Genetic Algorithm Optimization For Time Series Data. TIERS Information Technology Journal, 2023, 4(1), 16-24.

Masini, R. P., Medeiros, M. C., & Mendes, E. F. Machine learning advances for time series forecasting. Journal of economic surveys, 2023, 37(1), 76-111.

Wang, Z., Yan, W., & Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International joint conference on neural networks, 2017, pp. 1578-1585.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. On the importance of initialization and momentum in deep learning. In International conference on machine learning. 2013, pp.1139-1147.

Freeborough, W., & van Zyl, T. Investigating explainability methods in recurrent neural network architectures for financial time series data. Applied Sciences, 2022, 12(3), 1427.

Introduction to recurrent neural network https://www.Geeksforgeeks.org/introduction-to-recurrent-neural-network/

Yan, H., & Ouyang, H. Financial time series prediction based on deep learning. Wireless Personal Communications, 2018, 102, 683-700.

Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D, 2020, 404, 132306.

LSTM Recurrent Neural Networks — How to Teach a Network to Remember the Past, https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e

Yan, H., & Ouyang, H.. Financial time series prediction based on deep learning. Wireless Personal Communications, 2020, 102, 683-700.

Mehtab, S., & Sen, J. Analysis and forecasting of financial time series using CNN and LSTM-based deep learning models. In Advances in Distributed Computing and Machine Learning 2021. pp. 405-423.

Zhang, C., Sjarif, N. N. A., & Ibrahim, R. Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2024, 14(1), e1519.

Applied Deep Learning - Part 3: Autoencoders, https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. ASC, 2020, 90, 106181.