Journal of Mathematics and Modeling in Finance (JMMF)
Vol. 5, No. 2, Summer & Fall 2025
Research paper

Applications of Some Deep Learning Algorithms to Predict Trend in the Forex Exchange Market

Mohammad Ali Jafari[1], **Sina Ghasemilo**[2]

[1] Financial Mathematics Department, Finance Faculty, Kharazmi University
m.a.jafari.math@gmail.com

[2] Financial Mathematics Department, Finance Faculty, Kharazmi University
ghasemilosina1376@gmail.com

Abstract:
Predicting time series has always been one of the challenges in the financial markets. With the increase in the amount of data, the need to use modern tools instead of classical statistical and time series methods has become clear. In this paper, some deep learning algorithms such as Multilayer Perceptrons (MLPs), Keras Classification, Temporal Fusion Transformer (TFT, developed by Google), Extreme Learning Machine Classification (ELMC) and Propagation Hierarical Learning Network (PHILNet) are used for trading on the foreign exchange market. The efficiency and accuracy of these algorithms are presented. In this order, the EUR/USD data is used as input for the above algorithms.

Keywords: Deep learning, Forex market, Trend of the EUR/USD
Classification: 68T07

1 Introduction

Recently, deep learning algorithms have been widely used for climate and weather prediction, web traffic prediction, time series prediction in finance such as demand and sales prediction, and stock price prediction. In this paper, we have focused on applications of deep learning algorithms in the Forex market. Some applications of deep learning algorithms in the forex exchange market are mentioned below. For more information, see [1-5] and the references therein. Das et al. propose a hybrid forecasting model that combines Empirical Mode Decomposition (EMD) with fast reduced Kernel Extreme Learning Machine (KELM) [1]. The authors focused on forecasting foreign exchange rates and used several currency pairs, including CAD/HKD, CAD/CNY, CAD/USD, CAD/BRL, CAD/JPY, EUR/USD, and GBP/USD. Their work showed that the proposed AEMD-KELM method provides better accuracy. Kishore Kumar Sahu et al. focused on the prediction of exchange rates, which is important for financial managers and economic traders [2]. They used Extreme Learning Machines (ELM) augmented by metaheuristic algorithms such as Fireworks Algorithm (FWA), Chemical Reaction Optimization

[1]Corresponding author

Received: 19/05/2025 Accepted: 11/08/2025
https://doi.org/10.22054/JMMF.2025.85949.1183

(CRO), and Teaching Learning-Based Optimization (TLBO) for pre-training the ELM. This approach aimed to improve the learnability and accuracy of the forecasting model by taking into complexity and non-linearity of the exchange rate data. Navarro et al. presented an innovative architecture specifically designed for the prediction of time series data [3]. The PHILNet was shown to effectively capture complex patterns in historical data and improve the performance. The authors emphasized the model's ability to handle various time series such as the financial market [3]. Lu Zhao and Wei Qi Yan investigated transformer models such as Temporal Fusion Transformer (TFT) [4]. This study focused on the exchange rates of the currency pairs NZD/USD, NZD/CNY, NZD/GBP, and NZD/AUD. Imran Ali Khan investigated the prediction of exchange rates using machine learning and deep learning models, including LSTM, MLP, and Random Forest [5]. The study focused on exchange rates such as USD/PKR, PKR/USD, and MYR/PKR, and highlights the strengths of LSTM and Random Forest in predicting future trends. These findings are valuable for investors and policy makers and include recommendations for future improvement in real-time analysis and model interpretability to address the typical volatility in emerging market currencies.

In addition, deep learning algorithms have already been successfully applied in other markets [6-10]. Q. A. Nguyen et al. compared the TFT model with Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) to predict stock prices [6]. The study showed that the TFT, performs better than traditional models by utilizing attention mechanisms and combining temporal and static features. The results showed that the TFT achieved the lowest error rates compared to the SVR and the LSTM, which established it as an efficient tool for financial market analysis. Weng et al. investigated the importance of COVID-19-related news in predicting the volatility of crude oil futures [7]. In this study, an innovative model called GA-RFOS-ELM was introduced, which includes a genetic algorithm for parameter tuning and a forgetting factor. This factor was developed for online updating and learning to respond to real-time COVID-19 news in the crude oil market. Weng et al. presented an intelligent model called GA-ROSELM for predicting the gold price [8]. The model uses daily gold price data collected from public websites and employs a genetic algorithm for parameter optimization, combined with the Akaike Information Criterion (AIC). AIC helped select the best variables, including the price of silver, the S&P 500 Index, the price of crude oil, and gold prices over the last three days. The results showed that GA-ROSELM achieved higher accuracy compared to traditional models such as ARIMA, SVM, BP, ELM, and OS-ELM. Juan Laborda et al. applied the TFT for the joint forecasting of GDP in 25 OECD countries [9]. This study found that the TFT outperformed traditional regression models, especially in times of economic instability, such as during the COVID-19 pandemic. The authors emphasize the interpretability of the model, which enables an in-depth analysis of the influence of various explanatory variables on GDP forcasts. Zhang et al. who focus on predicting the performance of the Chinese macroeconomic system [10]. The study addresses the challenges faced in making accurate economic forecasts, and emphasizes that conventional economic forecasts often have low accuracy. The aim of this study is to introduce and apply the TFT model as an advanced method for forecasting economic performance. The authors developed a scientific forecasting framework based on Gross Final Product (GFP) to analyze the dynamic relationships between demand-side indicators and production indicators, and to highlight the non-linear characteristics of socio-economic systems. The TFT is used to predict output, incorporating attention mechanisms and variable selection.

The organization of this paper is as follows. In Section 2, a brief overview of the deep learning algorithms that have been applied in this paper, are presented. In Section 3 numerical results are presented. Finally, a brief conclusion is presented in Section 4.

2 Models

This Section provides a brief overview of the models used. The learning is divided into the following four parts.

1. Supervised learning

In this type of learning the training data contains the correct output for given input. In other words, in supervised learning, the goal is to find a mapping from inputs to outputs called labeled set.

2. Un-Supervised learning

This type of learning does not include the training data.

3. Semi-Supervised learning

In this type of learning, both types of data are included, i.e, the training data with the correct output for the given input, and the training data without the correct output for the given data.

4. Reinforcement learning

In this type of learning the correct output data does not contain. This approach focuses on behaviors that the machine performs to maximize its reward.

In this work, the supervised learning is used to predict time series. In other words, time series prediction is transformed into a supervised learning problem. First, the supervised learning model is trained and then the model is used for prediction.

First, the daily data of the forex exchange market related to EUR/USD is stored in rows of the DataFrame. If we look at the next day, whose data is in the next row of the DataFrame, its label is 2 if the increase on the next day, is more than 0.05%. If the decrease is less than -0.05%, its label is 0, and otherwise the label is 1. Using forex data in the period April 1, 2023 to April 1, 2024 and label column, the model is trained. After that, the model is used for prediction.

Remark

All the programs written by Python are available in the following Github page: link

MLP

The multilayer perceptron is a feedforward algorithm for artificial neural networks that is used for various types of regression, and classification, problems. The network consists of several layers of interconnected neurons, that allow it to learn complex relationships between inputs and outputs. This network contains nodes known as neurons, which are the computational units of the algorithm. The layer on the left, called the input layer, consists of a set of neurons that represent the input features. Each neuron in the hidden layer processes the values of the previous layer using a weighted linear summation, followed by the application of a non-linear activation function, such as the hyperbolic tangent function. Finally, the output layer receives the values from the last hidden layer and converts them into output values. In this work, the algorithm is optimized with 5 hidden layers, and the activation function Relu is used. This Multi-Layer Perceptron (MLP) model uses a grid search approach to optimize its hyperparameters for better performance in classification tasks. The model consists of fully connected layers, where the number of neurons in the hidden layers is a key consideration in the optimization. The hidden_layer_sizes parameter includes configurations such as (10,), (50,), (100,), (50, 50), and (100, 100), so that the model can experiment with different architectures. The best configuration found was (100, 100), which indicates two hidden layers, of 100 neurons each, allowing the model to learn complex patterns in the data. The activation function used throughout the network is the ReLU (Rectified Linear Unit), which introduces nonlinearity and improves the model's ability to capture complicated relationships in the data. To prevent overfitting, the alpha parameter is obtained 0.0001 by GridSearch. For the learning rate, the learning_rate_init parameter is set to 0.001, which is an optimal value for the initial step size during the training process, and allows for stable convergence. By using GridSearchCV with cross-validation cv=3, the model iteratively evaluates different combinations of these hyperparameters to obtain the optimal settings based on the accuracy values.

Keras Classifier

The KerasClassifier model was developed for multi-class classification tasks and uses a neural network architecture optimized for feature extraction and dimensionality reduction. This model improves prediction accuracy and generalization by systematically refining the feature space across multiple layers. First, the input data is processed by a first hidden layer of 64 neurons with the ReLU activation function, allowing the model to capture complex patterns. The second hidden layer

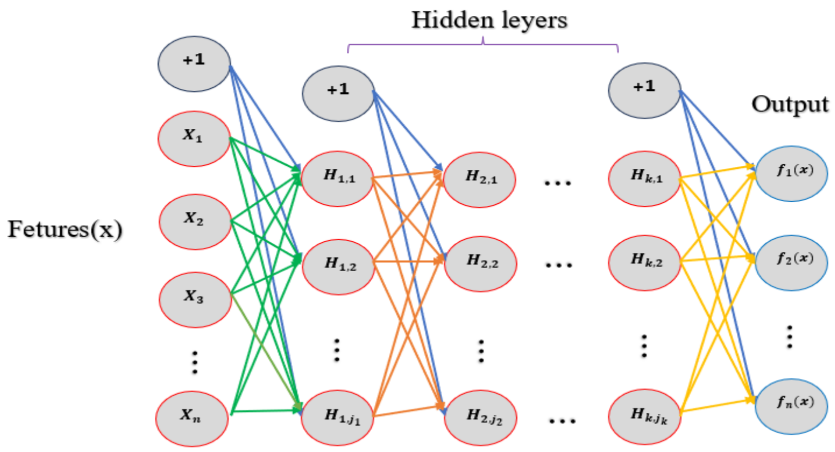

Figure 1: The structure of MLP algorithm

refines these features with 32 neurons, and continues the ReLU activation to improve the depth of the model during feature extraction. The final output layer consists of 3 neurons and using the softmax activation function, which is ideal for generating class probabilities in multi-class classification scenarios. The model is compiled using the Adam optimizer, which is known for its adaptive learning capabilities and efficient convergence. The loss function used is the sparse categorical cross entropy, which works directly with integer coded labels, to improve classification performance. This model is integrated into the KerasClassifier interface of Scikeras, allowing seamless use in scikit-learn pipelines. The training process includes 20 epochs and a batch size of 32, balancing frequent updates to maintain stability and prevent overfitting. After training, the model is tested on unseen data, tracking prediction time to evaluate computational efficiency. Test accuracy serves as the primary evaluation metric, and measures the generalization ability of the model. Improvements such as batch normalization can be introduced after each layer to stabilize activations and accelerate convergence. In addition, dropout layers can be applied at a rate of 0.2 to reduce overfitting.

Temporal Fusion Transformer

The Temporal Fusion Transformer (TFT) algorithm developed by Google. It is an attention-based deep learning model designed for multi-horizon time series forecasting. By integrating multi-head attention mechanisms with a fusion decoder, TFT can capture both short-term and long-term dependencies while providing interpretable explanations of temporal dynamics. One of its advantage is its ability to handle diverse input types (such as time-based, contextual, and time-series data) and deliver accurate forecasts. Compared to traditional algorithms alike LSTM and GRU, TFT demonstrates superior performance, particularly in long-term forecasting [11]. The model is implemented with a hidden size of 8, two LSTM layers, a dropout rate of 0.1, a loss function of is CrossEntropy, and max_epoch is 10 in this tuning. It is trained using PyTorch Lightning and employs time series data for label prediction.

Extreme Learning Machine

The ELMClassifier class is utilized to create a prediction model that employs the Extreme Learning Machine (ELM) method. This model is suitable for multi-class or multi-label classification and can be updated with new data. Key points in model optimization include the initial settings, where the number of neurons in the hidden layer is set to 100, the regularization parameter (alpha) is set to 1e-7, the batch size for model updates is 40, and the activation function is set to Relu. Additionally, the Euciliden metric is selected. After adjusting the model parameters, the training data, X_train_scaled and y_train, is applied for training.

PHILNet

This neural network algorithm, based on hierarchical learning (developed by Max Planck Institute) is inspired by the functioning of the human cerebral cortex and is applied to improve predictions in multi-step time series problems. The algorithm's architecture is a modified version of HLNET, where complex tasks are broken down into simpler ones and processed in a step-by-step manner. A smoothing layer is used to reduce noise and fluctuations in input data by employing methods such as moving average, Gaussian filters, and exponential smoothing. This layer enables the model to better focus on primary trends and patterns. The smoothing parameter σ is an important hyperparameter that determines the size of the moving average window and significantly impacts the model's accuracy. Forward Step and Error Calculation in Each level generates its specific prediction, and its error is smoothed using future real data. Errors from the initial levels contribute to reducing convergence time. In the implemented PHILNet model, we use a combination of LSTM layers and fully connected layers (fc1, fc2) for feature extraction and classification. The model starts with an LSTM layer that processes sequential data, capturing temporal dependencies. The LSTM output is then passed through two fully connected layers: fc1 maps the output to 128 neurons, and fc2 reduces it to 64 neurons. The smoothing layer maps thesen 64 neurons to the number of classes (num_classes), providing the final predictions. To optimize the model's hyperparameters, we employed Grid Search, exploring various combinations of hidden dimensions, learning rates, batch sizes, and epochs. The best configuration found was with a hidden dimension of 32, a learning rate of 0.001, a batch size of 64, and 30 epochs. This tuning significantly improved the model's accuracy and performance. Future enhancements could include integrating Batch Normalization and Dropout layers to further accelerate convergence, reduce overfitting, and improve generalization. Currently, the model applies ReLU activation functions between layers to introduce non-linearity, enhancing its learning capacity. This combination of temporal feature learning by LSTM and dense feature extraction, coupled with hyperparameter optimization, makes the model highly robust and effective for multi-class classification tasks in financial time series data. The PHILNet algorithm structure is shown in figure 2.

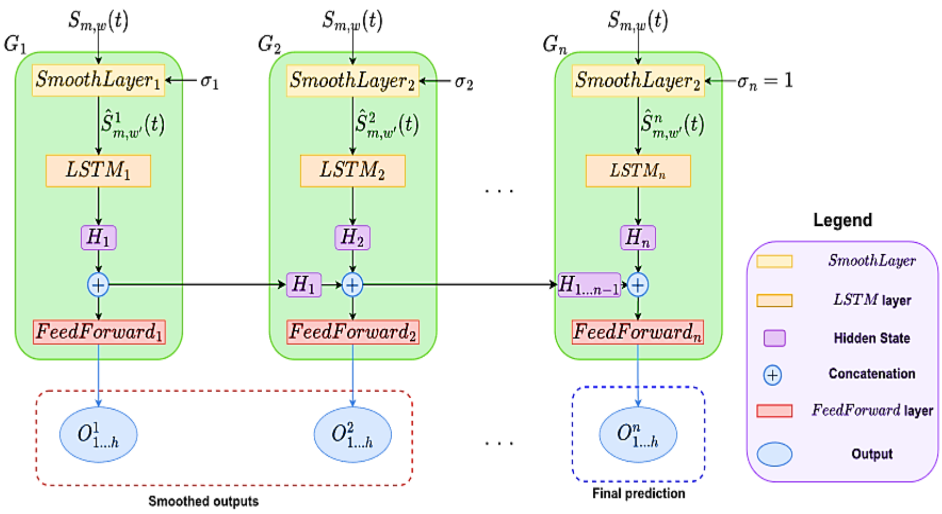

Figure 2: The structure of PHILNet algorithm [3]

3 Numerical Results

In this section, the confusion matrix of the aforesaid algorithms (i.e. MLP, Keras, TFT, ELM and PHILNet) are presented in figures 3-7.

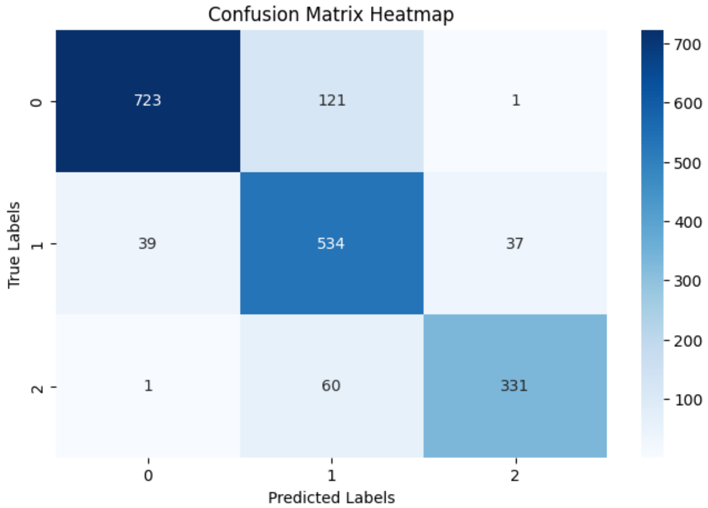

Figure 3: Heatmap of the MLP algorithm

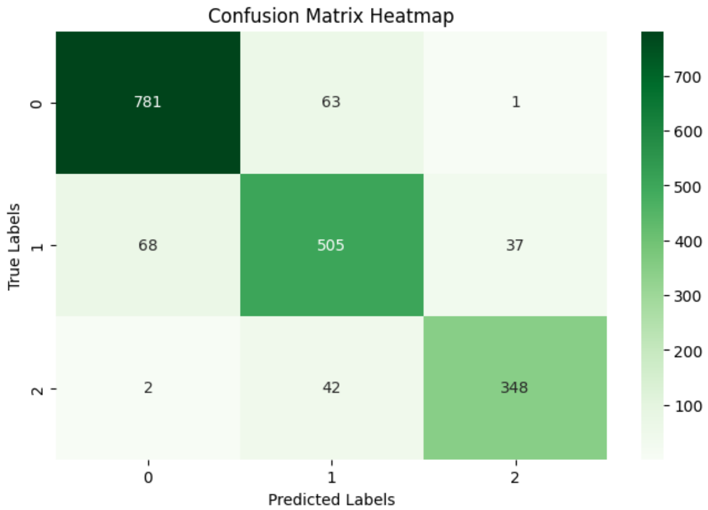

Figure 4: Heatmap of the KerasClassifier algorithm

Furthermore, the accuracy, precision, recall, F_1-Score, and run time of each algorithm is mentioned in the Table1.

Based on the results, since the PHILNet is the best algorithm, the performance of the algorithm with different values of the Hidden_Dim, Learning rate, Bach size, and Epoch is presented in the Table 2.

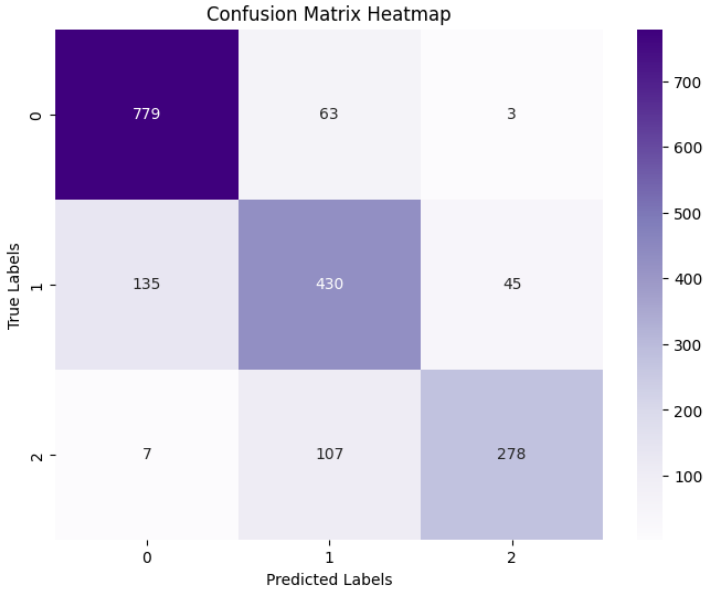

Figure 5: Heatmap of the TFT algorithm

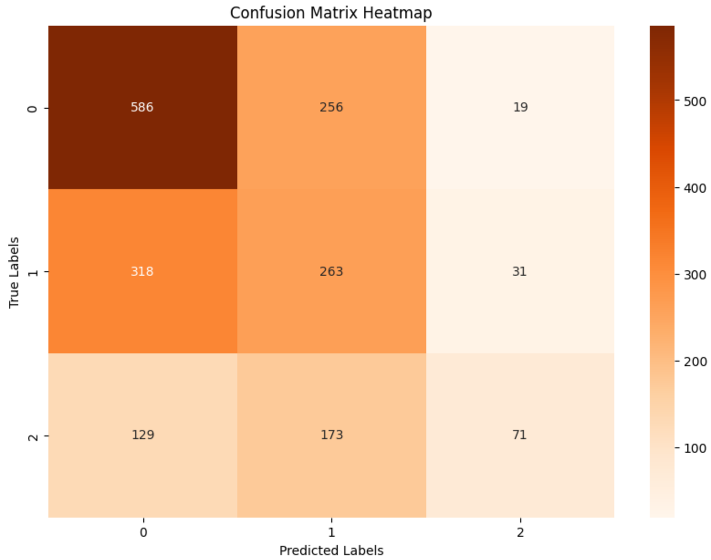

Figure 6: Heatmap of the ELM algorithm

Metric	Hidden_Dim			Learning Rate		Batch Size			Epoch	
	32	64	128	**0.01**	0.001	**16**	32	64	20	**30**
Accuracy	**88.46**	87.16	88.36	**88.46**	88.35	**88.46**	87.70	87.16	88.35	**88.46**
Precision	**88.45**	87.14	88.44	**88.45**	88.34	**88.45**	87.80	87.14	88.34	**88.45**
Recall	**86.46**	87.16	88.36	**88.46**	88.35	**88.45**	87.70	87.16	88.35	**88.46**
F$_1$-Score	**88.45**	87.13	88.35	**88.45**	88.31	**88.45**	87.74	87.13	88.31	**88.45**

Table 2: Grid search of PHILNET report

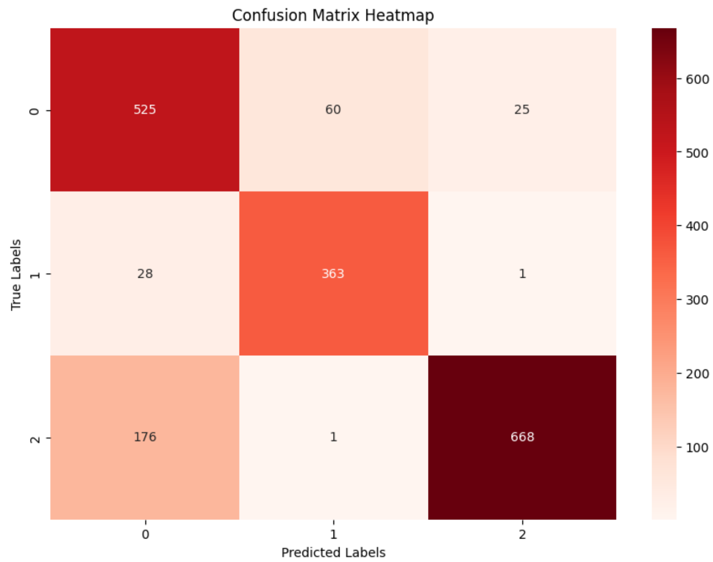

Figure 7: Heatmap of the PHILNet algorithm

Algorithm	**Accuracy**	**Precision**	**Recall**	**F_1-Score**	**Time**
MLP	84.35	84.68	84.35	84.19	215.83
Keras	86.30	86.95	86.30	86.48	8.26
TFT	54.65	51.26	54.65	41.98	414.50
ELM	83.37	83.44	83.37	83.34	1.47
PHILNet	88.46	88.45	88.46	88.45	735.25

Table 1: Algorithms report.

4 Conclusion

In this paper, five different types of deep learning algorithms are used for trading in the foreign exchange market. The results show that deep learning algorithms can successfully predict the trend of the EUR/USD currency pair. The performances of these algorithms show that each algorithm has strengths. Although the running time of the PHILNet algorithm is longer than the other algorithms, its accuracy, precision, and F_1-Score are the best.

Bibliography

[1] Das, A., Kumar, K., and Sahu, K. K., *Data decomposition based fast reduced kernel extreme learning machine for currency exchange rate forecasting and trend analysis*, Expert Syst.

Appl. **92** (2018), 103–116.

[2] Sahu, K. K., Nayak, S. C., and Behera, H. S., *Forecasting exchange rates using Extreme Learning Machines enhanced with metaheuristic algorithms*, Internat. J. Comput. Intell. Syst. **16** (2023), no. 1, 67–85.

[3] Jiménez Navarro, M. J., Martínez Ballesteros, M. d. M., Martínez Álvarez, F., and Asencio Cortés, G., *PHILNet: A novel efficient approach for time series forecasting using deep learning*, Inform. Sci. **632** (2023), 815–832.

[4] Zhao, L. and Yan, W. Q., *Prediction of currency exchange rate based on Transformers*, J. Risk Financ. Manag. **17** (2024), no. 8, 332.

[5] Khan, I. A., *Evaluating foreign exchange rate forecasting: A deep dive into LSTM, MLP, and Random Forest approaches*, Preprint (2023).

[6] Nguyen, Q. A., Son, H., and Nguyen, P., *A lightweight multi-head attention transformer for stock price forecasting*, Preprint (2023).

[7] Weng, F., Zhang, H., and Yang, C., *Volatility forecasting of crude oil futures based on a genetic algorithm regularization online extreme learning machine with a forgetting factor: The role of news during the COVID-19 pandemic*, Resour. Policy (2021), 102148.

[8] Weng, F., Chen, Y., Wang, Z., Hou, M., Luo, J., and Tian, Z., *Gold price forecasting research based on an improved online extreme learning machine algorithm*, J. Ambient Intell. Humaniz. Comput. **11** (2020), 4101–4111.

[9] Laborda, J., Ruano, S., and Zamanillo, I., *Multi-country and multi-horizon GDP forecasting using Temporal Fusion Transformers*, Math. **11** (2023), no. 12, 2625.

[10] Ouyang, R., Pei, T., Fang, Y., and Zhao, Y., *Commodity systemic risk and macroeconomic predictions*, Energy Econ. (2024), 107807.

[11] Lim, B., Arik, S., Loeff, N., and Pfister, T., *Temporal Fusion Transformers for interpretable multi-horizon time series forecasting*, Internat. J. Forecast. **37** (2021), no. 4, 1748–1764.

How to Cite: Mohammad Ali Jafari[1], Sina Ghasemilo[2], *Applications of Some Deep Learning Algorithms to Predict Trend in the Forex Exchange Market*, Journal of Mathematics and Modeling in Finance (JMMF), Vol. 5, No. 2, Pages:65–75, (2025).