© DEC 2024 | IRE Journals | Volume 8 Issue 6 | ISSN: 2456-8880

A Comparative Analysis of Machine Learning Algorithms for USD/EUR Foreign Exchange Rate Forecasting

OBI-OBUOHA ABIAMAMELA[1], MUBARAQ ONIPEDE[2], OWA OLORUNSOLA JASON[3]
[1,2,3] *National Center for Artificial Intelligence and Robotics, National Information Technology Development Agency, Abuja, Nigeria.*

Abstract- The Forex market is a beast—complicated, ever-changing, and influenced by an overwhelming number of factors. Yet, in the chaos lies opportunity. With the advancement of machine learning, are we finally capable of taming this beast? This study dives deep into the comparison of four machine learning models—Artificial Neural Network (ANN), Autoregressive Integrated Moving Average (ARIMA), Random Forest, and Support Vector Machine (SVM)—to predict the hourly price movements of the EUR/USD currency pair. The dataset, sourced from MetaTrader 5, spans over three years and was engineered to include domain-specific indicators like RSI, Bollinger Bands, and moving averages. ARIMA performed well at spotting linear trends but couldn't handle the market's unpredictable, non-linear flips. ANN stepped up, proving its worth in capturing these chaotic patterns, especially during sudden trend changes. Random Forest, though brilliant in training, suffered a severe case of overfitting when tested. As for the SVM models, the Linear and RBF kernels showed balance, while the Polynomial and Sigmoid kernels stumbled in tackling the intricate dance of Forex data. Each model had its highs and lows: ARIMA's precision with linear trends, ANN's knack for non-linear shifts, Random Forest's training prowess but testing pitfalls, and the SVM's moderate but stable performance. The takeaway? ANN shines brightest in volatile markets, ARIMA is the go-to for linear trend detection, and a hybrid approach combining their strengths might just be the silver bullet. In conclusion, this study underscores the importance of picking the right tool for the job in the Forex market. Careful parameter tuning, feature engineering, and even hybrid models could hold the key to consistently outsmarting the market.

Indexed Terms – Autoregressive Integrated Moving Average, Support Vector Machine, Random Forest, Artificial Neural Network.

I. INTRODUCTION

"In this business, if you're good, you're right six times out of ten. You're never going to be right nine times out of ten." Peter Lynch. —But does this statement remain true? Especially considering the waves AI is making in today's world of prediction and analysis, delivering results with prophetic precision. By the end of this paper, we see whether or not Peter Lynch stands corrected.

The forex market is extremely complicated, with literally countless variables continuously interacting, causing constant instantaneous changes every millisecond. Over the years, Forex traders and analysts have studied the relationships between some of these variables, and how they affect Foreign Exchange. It has proven impossible to capture all the variables, but the key ones have been encapsulated into concepts called indicators that help simplify global trends and human sentiment. Finally, it comes down to simple supply and demand: if more people buy a currency, its value rises; if more people sell, its value falls. However, various factors influence market sentiment, making it difficult to forecast future price fluctuations with full certainty.

For traders, forex indicators are useful instruments for examining market patterns and spotting possible trading opportunities. The Relative Strength Index (RSI), which gauges momentum and overbought/oversold conditions, Bollinger Bands, which display price volatility and possible reversals, and Moving Averages, which smooth out price data to identify patterns, are a few examples of popular indicators. Traders can make better decisions on when to enter and exit trades by being aware of these indicators.

Hence rather than an individualistic approach to analyzing complicated events that affect market behavior, structures have been created that visually represent several factors that help simplify the complexity of the Forex Market (*Still on indicators*). This statement also captures the fact that prediction models are limited by that simplicity.

This in my opinion justifies the results of Machine Learning models, considering that the quality of predictions largely depends on the quality of the data used in training, and for your information—the features are the indicators.

The purpose of this study is to assess and contrast how well different machine learning models predict the hourly price fluctuations of the EUR/USD currency pair.

II. LITERATURE REVIEW

On November 5th, 2024, at the National Centre for Artificial Intelligence and Robotics, Obi-Obuoha Abiamamela addressed a room of eager young minds. With energy in his voice and passion in his words, he declared, "We are going to publish. Yes! We are going to publish." It was not just a statement—it was a challenge, a call to action to push the boundaries of what we know and share it with the world.

This chapter takes up that challenge, diving into the fascinating, unpredictable world of forex trading. Predicting currency movements, like the EUR/USD pair, is not easy—it is like trying to read the wind. But with supervised machine learning, we have tools to make sense of the chaos. From Support Vector Machines to Neural Networks, we will explore how these methods help us make better predictions and where they still fall short in the race to understand and master the forex market.

A. Forex

The Forex market is the world's largest financial market, where currencies are traded internationally [1, 2, 12]. It is a market that is open around the clock. Banks, corporations, governments, and individual traders all participate in the purchase, sale, and speculation of currency pairs. The Forex market supports worldwide trade and investment by allowing currencies to be exchanged for one another.

What are forex indicators? Forex indicators are instruments for analyzing historical price and volume data in the currency market. They assist traders in identifying market patterns and prospective trading opportunities. Leading indicators forecast future price changes, while trailing indicators confirm patterns that have already begun [2].

Euro/USD pair (EUR/USD)—is the most popular currency pair in the world. This is because the Eurozone and the US are the two largest economies in the West. Interest rate changes by the European Central Bank and the Federal Reserve can significantly impact the value of the Euro compared to the US Dollar. Additionally, economic problems in European countries like Italy and Greece can weaken the Euro and strengthen the Dollar [34].

B. Supervised Machine Learning

Supervised learning, sometimes referred to as supervised machine learning, is a subset of artificial intelligence and machine learning [3]. It is distinguished by its application of labelled data sets to train algorithms that reliably identify information or forecast results [3, 26].

Supervised learning trains models using labeled data, teaching them to produce correct outputs. This training involves feeding the model input data paired with their corresponding correct outputs. As determined by a loss function, the model learns by modifying its parameters to reduce the discrepancy between its predictions and the actual outputs [3].

Supervised learning can be categorized into two primary tasks as can be seen in Figure 1.

- Classification: This involves assigning data points to specific categories. Algorithms like linear classifiers, SVMs, decision trees, k-nearest neighbors, and random forests are commonly used for this task.
- Regression: Involves predicting numerical values. Its goal is to represent the link between input

variables and a continuous output variable. Popular regression algorithms include linear regression, logistic regression, and polynomial regression.

C. Support Vector Machine

Support Vector Machines (SVMs) are powerful tools in machine learning, designed to handle a variety of tasks like classification, regression, and even detecting anomalies in data. They are especially useful in areas such as text analysis, recognizing images, filtering spam, analyzing handwriting, studying gene patterns, identifying faces, and spotting unusual data points [4].

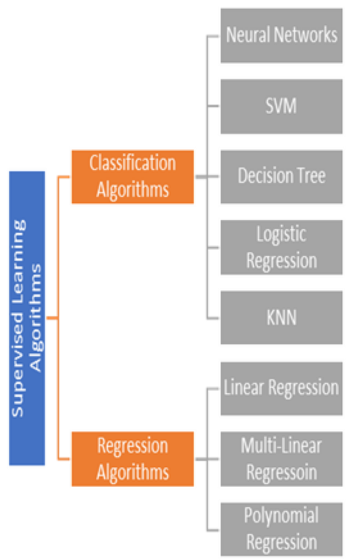

Figure 1. The Supervised Learning Family Tree

What makes SVMs stand out is their focus on finding the best possible boundary (called a hyperplane) that separates different groups in a dataset. This approach ensures they work well for both simple binary classifications and more complex multiclass problems. SVMs are incredibly flexible and reliable, making them a go-to choice for tackling a wide range of real-world machine-learning challenges [4, 5]. Its ability to classify different types of data points can be seen in Figures 2,3,4 and 5.

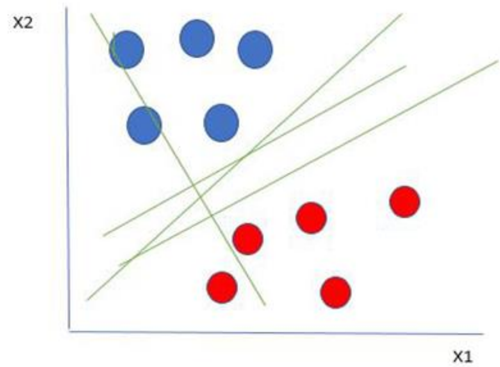

Figure 2. Linearly Separable Data Points [6].

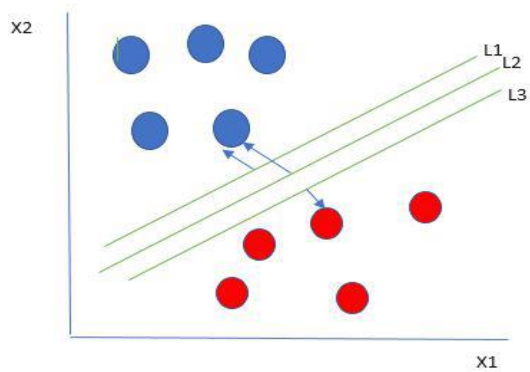

Figure 3. Multiple hyperplanes can separate the two classes [7].

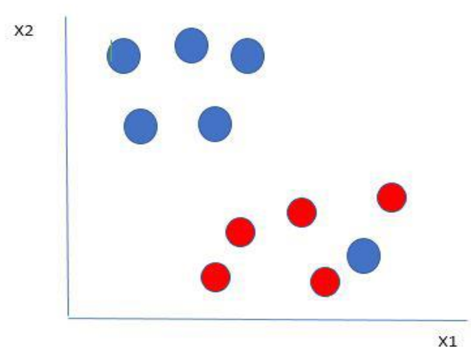

Figure 4. Selecting hyperplane for data with outliers [8].

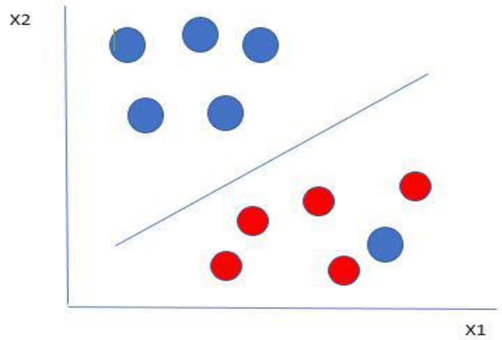

Figure 5. Hyperplane which is the most optimized one [9].

D. Random Forest

Random Forest is an ensemble method that creates a collection of decision trees for either classification or regression tasks [11, 12, 10]. Each tree is built by randomly selecting samples from the training data, with replacements. Additionally, at each decision point (node) in the tree, a subset of the total input features is randomly chosen, and the best split is selected based on those features. This process is repeated for each tree in the forest [11]. An illustration of its working principle can be seen in Figure 6.

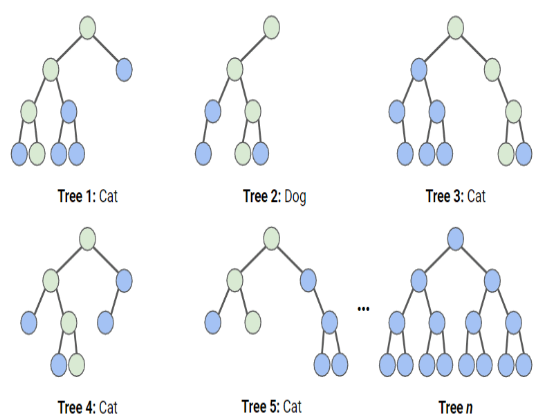

Figure 6. Illustration of how random forest classification works [13].

The trees are allowed to grow fully without any pruning, meaning they are not limited in size. The final prediction is made by aggregating the outputs from all the trees: in classification, this involves a majority vote, and in regression, it involves averaging the results [10, 12].

Random Forest typically performs better than a single decision tree like C4.5 (a popular decision tree algorithm that creates binary trees by splitting data based on the feature that best separates the classes) because it reduces overfitting and increases generalization accuracy. It tends to be more reliable and less noise-sensitive than other methods like Adaboost (a boosting technique that combines multiple weak classifiers to improve overall performance) [12].

E. Recurrent Neural Network

A Recurrent Neural Network (RNN) is a type of neural network that goes beyond traditional feedforward networks by incorporating a form of memory. Unlike standard models where each input is treated separately, RNNs use their internal state to retain information from previous steps, allowing them to process sequences of data in a more context-aware way [14]. This means that the network not only evaluates the current input but also considers the outputs it has generated from prior inputs to inform its next decision.

The recurrent nature of RNNs means that after each input is processed, the output is passed back into the network, creating a loop that reinforces the model's memory. This makes RNNs particularly well-suited for tasks that involve sequential data, such as speech recognition or handwriting analysis, where the order of the data is crucial. While other neural networks treat each input independently, RNNs recognize the relationship between successive inputs, giving them a distinct advantage in tasks where context or sequence matters [14, 15].

F. Autoregressive Integrated Moving Average

The Autoregressive Integrated Moving Average (ARIMA) model is a sort of regression analysis that assesses the strength of a single dependent variable in the context of many changing variables. The approach seeks to anticipate future securities or financial market movements by analyzing the differences between values in the series rather than actual values [17, 19]. [18] explains that in time-series analysis, autoregression is a statistical method that makes the assumption that a time series' present value is a function of its previous values.

Autoregressive models use linear regression with lagged variables derived from prior steps [18]. Unlike linear regression, the autoregressive model does not include any additional independent variables other than the previously projected outcomes. Consider the following formula in Equation 1.

$$p(x) = \prod_{i=1}^{n} p(x_i|x_1, x_2, \ldots x_{i-1}) = \prod_{i=1}^{n} p(x_i|x_{<i})$$

(1)

An autoregressive model, as stated in the probabilistic term, assumes that the results of the previous variables conditionally affect the subsequent one and distributes independent variables over n potential stages.

Integrated (I): reflects the differencing of raw observations so that the time series becomes stationary.

Moving average (MA): A moving average model applied to lagged observations takes into account the reliance between an observation and a residual error [17].

G. Artificial Neural Network

Artificial Neural Networks are composed of a vast number of interconnected "artificial neurons" that receive input signals from other neurons, analyze them using nonlinear functions, and then transmit them. Weights (W) influence the strength of the connections between neurons [21, 22, 23].

Feed-forward propagation is like passing information straight through a neural network. It starts with the input, moves through hidden layers, and finally reaches the output. At each step, the neurons calculate their values based on the inputs they get, using weights and an activation function. This process is how the network makes predictions or decisions [24, 25].

Backpropagation, on the other hand, is all about learning from mistakes. After the network makes a prediction, it checks how far off it was from the correct answer. Then, it works backward, layer by layer, adjusting the weights to make the prediction better next time. It's a bit like practicing and improving after getting feedback [25]. These can be seen in Figure 7.

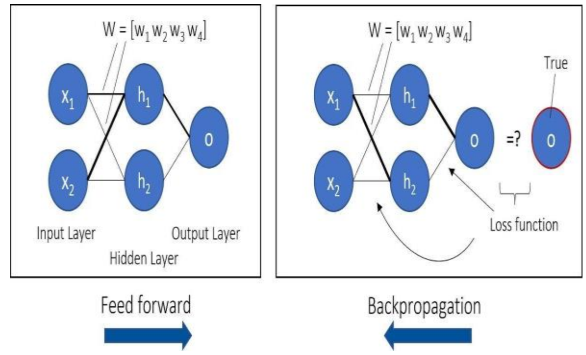

Figure 7. A neural network is shown schematically, with feed-forward and backpropagation [20].

H. Related Works

In Forex trading, an indicator of currency pairs performance creates the problem of predicting an increase or decrease in an exchange rate, thus it could be a binomial classification problem where the goal is to guess, depending on the input parameters, how the exchange rate at the output will change [33, 26]. Despite the fact that this problem can be solved using both traditional statistical approaches and neural networks, these methods have not proven very effective. The following compiles a review of some of the literature published in this domain.

Another first-generation model that has been applied in the analysis and forecasting of financial time series including Forex rates is the ARIMA. In dealing with non-stationary data, and especially when confronted with nonlinear effects – a feature typical for financial returns – ARIMA models are rather restrictive [26]. Furthermore, it is obvious that most of the ARIMA models impose a constant variance and normal distribution of the data points which are very unlikely for the Forex market because of high fluctuations [27]. There has been the emergence of other models in the recent past, particularly the ANNs machine learning, which seem to complement some weaknesses of the ARIMA. For instance, the use of MLP is common for predictions of Forex because they are capable of describing a nonlinear relationship. However, a major issue that has been pointed out in previous research in respect of these models is that the datasets employed to train them was relatively large. This may lead to overfitting as well as poor real-life predictions, due to the small amount of information on which to draw [26].

Furthermore, as many experts have noted, noisy characteristics of Forex data decrease the effectiveness of such MLPs and other supervised learning models, for instance, Support Vector Machines (SVM). There is a key drawback of these models; they fail quickly when faced with the high frequency and variability that is characteristic of financial time series, and which results in lower accuracy and unreliable predictions [27].

Another field in which Forex prediction was searched is an attempt to use Recurrent Neural Networks (RNNs): Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). They are excellent in dealing with sequential data and modeling temporal relations, which qualifies these models for predicting Forex rates coming after patterns in their historical rates. Nonetheless, several researchers have observed that application of these models in Forex trading trade trapped in massive hurdles due to high tuning and optimization.

Trading strategies have used Genetic Algorithms (GA) to enhance the process. A study carried out by [28], shows that GA can be used to improve trading strategies that employ technical tools called the trend indicators. Nevertheless, such approaches can turn out to be unprofitable if expenses for trading exceed a certain level.

ANNs have been employed for several years to forecast the forex rates. A study [29] further showed that ANNs are proficient in capturing nonlinear relationships in forex data. Yet, the discussion on how to create the highest quality ANN continues to rage on the topic of choosing the proper structure and input.

The review on the selected forex forecasting model's LSTM and ANN was carried out systematically. This evaluation endowed the research with basic measures such as RMSE, MAE, and MSE in measuring the performance of the chosen models. However, the preoccupation with the EURUSD currency pair means that the findings cannot be easily applied to other currency pairs [30].

Another study looked at four currencies to compare the LSTM with Support Vector Regression (SVR), and Random Forest in forex rate prediction. Preliminary findings indicated that compared to SVR and LSTM models achieved higher prediction accuracy than the Random Forest. The research also postulates that the best models should be assessed using more than one key figure, including MAE or RMSE [31].

The authors [32] benchmarked SVM with other models such as; ANN and ARIMA in forex rate prediction. Classification was slightly better with SVMs with radial basis function kernels but tend to be more difficult to put in the correct form and would take longer time to use in different situations.

III. METHODOLOGY

This methodology outlines the steps taken in developing and evaluating the predictive models: An Artificial Neural Network (ANN), Autoregressive Integrated Moving Average (ARIMA) model, Rando, Forest and Support Vector Machine. The steps include data collection, preprocessing, model training, evaluation, and comparison.

A. Data Extraction and Preprocessing

Data Source and Collection

Historical EUR/USD exchange rate data was obtained from MetaTrader 5, covering the period from January 1, 2013, to September 30, 2016. The dataset included the following fields: Open (price at the start of the hour), High (highest price during the hour), Low (lowest price during the hour), and Close (price at the end of the hour). To balance detail and computational efficiency, an hourly (H1) timeframe was selected, as it captures medium-term trends while minimizing noise. The exported data, saved in TSV format, was processed using Python scripts to convert timestamps to datetime format and filter the dataset into training (January 1, 2013, to December 31, 2015) and testing (January 1, 2016, to September 30, 2016) subsets. The cleaned data was then saved in CSV format for further analysis. The choice of hourly data was based on its ability to balance granularity and interpretability, effectively capturing intra-day price trends while reducing the excessive noise associated with minute-level data.

Feature Engineering

Feature engineering was essential in preparing the dataset for ARIMA and ANN models. The initial dataset included fields such as date, time, open, high, low, close, tickvol, vol, spread, and trend. The trend was created by comparing consecutive Close prices. An uptrend (1) was assigned if the current price was higher than the previous, and a downtrend (0) was assigned if it was lower. As can be seen in Figure 8.

To enhance predictive performance, a correlation analysis was conducted to identify relevant features and create new, domain-specific features. As a result, the engineered features included open, high, low, close, rsi (Relative Strength Index), ma9 (9-period Moving Average), ma21 (21-period Moving Average), bb_upper (Bollinger Band Upper), bb_middle (Bollinger Band Middle), bb_lower (Bollinger Band Lower), and atr (Average True Range).

$$\text{TREND}_t = \begin{cases} 1 & \text{if close}_{t+1} > \text{close}_t \\ 0 & \text{otherwise} \end{cases}$$

Figure 8. Logic of TREND Feature.

All features were normalized using StandardScaler to ensure faster convergence during model training. The newly created features, derived from domain knowledge, were critical in capturing market dynamics and improving the models' ability to predict trends. Technical details of feature selection and creation emphasize the integration of statistical indicators to enrich the dataset and enhance model interpretability.

Class Imbalance Handling

To address potential class imbalance in the training dataset, class weights were computed using the “compute_class_weight” function from Scikit-learn. The 'balanced' mode was employed to calculate weights inversely proportional to the class frequencies, ensuring that underrepresented classes were given higher importance during training. This step mitigates bias toward the dominant class and improves the model's performance on minority classes.

The y_train labels were converted into a NumPy array format to align with the computational requirements of the function. For future scalability to multi-class classification tasks, the computed weights were organized into a dictionary for direct compatibility with model training frameworks. The implementation steps are summarized below:

This approach ensures that class imbalance does not adversely affect the training process, enhancing the model's fairness and predictive capabilities across all classes.

B. Model Development

ARIMA (Autoregressive Integrated Moving Average)

ARIMA is a statistical model for time series forecasting that makes predictions about future values by using error terms and historical observations. The model works best with univariate time series data, where moving average (MA), autoregressive (AR), and differencing (I) components are used to make predictions.

- ARIMA Model Selection Process

Making sure the time series is stationary—that is, that its statistical characteristics do not alter over time—is the first stage in choosing an ARIMA model. The Augmented Dickey-Fuller (ADF) test was used to check for stationarity. The test's null hypothesis cannot be disproved if the series is determined to be non-stationary, suggesting that there are trends or seasonality in the data. To get rid of trends and stabilise the series mean in these situations, differencing is used, which involves subtracting the current value from the previous value. By eliminating seasonal oscillations and long-term trends, this transformation aids in making the series stationary.

After achieving stationarity, choose the appropriate ARIMA parameters: p (autoregressive order), d (degree of differencing), and q (moving average order). The best values for these parameters are estimated using Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) graphs. The ACF and PACF are diagnostic techniques used to discover the underlying structure of a time series and select the best-fit model, this suggests an ARIMA (0, d, q) model. These plots help guide the selection of the appropriate AR and MA terms for the ARIMA model. In the case of this study, after applying the ADF test, the p-value was found to be less than 0.05, which allowed us to reject the null hypothesis and conclude that the series is indeed stationary. However, if the p-value had been greater than 0.05, we would have needed to apply differencing again until the series became stationary. Once stationarity was confirmed, the dataset was deemed suitable for the application of the ARIMA model.

Thus, ensuring stationarity through differencing is a critical step in preparing the data for ARIMA modeling, as non-stationary data would lead to biased or misleading results.

- ARIMA Model Training

The ARIMA model was trained using the training dataset's historical Close prices. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots provided valuable insights into the model parameters, which included the autoregressive order (p), differencing degree (d), and moving average order (q). For example, an ARIMA (1,1,1) configuration was found to be optimal for the dataset. Once the parameters were determined, the ARIMA model was manually fitted to the data. It is important to note that ARIMA models are applied directly to non-stationary time series data. The differencing step, which is essential for achieving stationarity, is automatically handled by the ARIMA model internally during the fitting process.

The trained model was then used to anticipate future values based on previous trends and the error structure discovered during the training phase. The model's performance was evaluated using the summary result from model fitting shown below. This gave important statistics and insights into the model's accuracy and fit, which were necessary for evaluating its predicting skills.

ANN (Artificial Neural Network)

The Artificial Neural Network (ANN) model was developed to capture non-linear patterns in the data, offering a flexible approach to prediction tasks, particularly in understanding complex relationships within financial time series.

- ANN Network Architecture

The architecture of the ANN consists of several key layers. The input layer receives standardized features to ensure uniform scaling during model training, enabling the network to learn efficiently. The model includes nine hidden layers, all of which are fully connected dense layers. These layers utilize the ReLU (Rectified Linear Unit) activation function, which is commonly used to introduce non-linearity into the network. This formula can be seen in Equation (2)

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2}$$

The number of neurons in each hidden layer was fine-tuned through testing to improve the model's predicting capabilities. The output layer is composed of a single neuron with a sigmoid activation function, which is appropriate for binary classification applications, such as predicting the direction of market trends (i.e., up or down). The mathematical expression can be seen in Equation 3.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

- ANN Training Process

The training data was divided into two sets: training and validation, with a validation split of 0.13 (13%). To minimize the loss function, the model was trained utilizing backpropagation and the Adam optimizer. Furthermore, class weights were introduced to handle any potential class imbalance in the dataset, guaranteeing that the model can effectively learn from both classes. In order to prevent overfitting and enhance the training process, early stopping was implemented, monitoring the validation precision. This mechanism ensures that training halts if there is no improvement in precision over a set number of epochs, in this case, 8 epochs. The model was configured to restore the best weights to preserve the optimal state after training. This approach allowed the ANN to be trained effectively, with a focus on maximizing precision as a key evaluation metric. The ANN leveraged engineered features (e.g., PIP change, rolling averages) and multi-layer architecture to model complex, non-linear patterns. The addition of trend-based features enhanced the ANN’s precision in capturing directional changes.

Random Forest

A Random Forest Classifier was employed, configured with the following hyperparameters:

- n_estimators = 350: The number of decision trees in the forest is specified by this parameter. By lowering variance and boosting robustness, a larger tree count typically increases the model's accuracy. However, it also increases computational cost. Setting this value to 350 aims to strike a balance between accuracy and efficiency.
- max_depth =150: This setting regulates each decision tree's maximum depth. Although it is more likely to overfit, a deeper tree might catch more intricate patterns. By keeping the depth at 150, overfitting can be avoided while yet permitting adequate complexity.
- min_samples_split =3: This parameter determines the minimum number of data points required to split an internal node. A higher value can lead to simpler trees, reducing overfitting, but may also result in underfitting if set too high. A value of 3 is a reasonable choice to balance these trade-offs.
- Random_state = 1: This parameter ensures that the model's output is reproducible by setting the seed for the random number generator. The behavior of the model can be reliably reproduced across runs by fixing the random state.

Support Vector Machine (SVM)

For the SVM model, different kernels were used to train the model in an attempt to capture more complexity and check which would perform best.

- SVM with Linear Kernel: For linearly separable data, this model's linear kernel is appropriate. It determines the best hyperplane to divide the classes.
- SVM with Sigmoid Kernel: The sigmoid kernel is appropriate for challenging classification problems because it adds non-linearity to the decision boundary. It can be thought of as a single hidden layer neural network. The C and gamma parameters, in particular, which regulate the trade-off between margin maximization and misclassification penalty, have a significant impact on the model's performance.
- SVM with Polynomial Kernel: More complex decision boundaries are made possible by the polynomial kernel, which maps data into a higher-dimensional space. The complexity of the model is determined by the degree of the polynomial kernel. While a lower degree might miss complex patterns, a higher degree could result in overfitting. Finding the ideal degree and other factors requires hyperparameter adjustment.
- SVM with Radial Basis Function (RBF) Kernel: Because it can efficiently handle non-linear interactions, the RBF kernel is a common choice for SVM. It allows for versatility in creating intricate patterns by mapping data into an infinite-dimensional space. The impact of each individual data point on the decision boundary is managed by the gamma parameter. A more complex decision boundary with a higher gamma value is more likely to overfitting.

IV. RESULTS

Various evaluation metrics were used to determine to assess the performance of the models on the dataset as well as its ability to guide predictions in a real-world scenario. Some of which include precision, accuracy and recall.

Precision (P)

The ratio of true positive detections to the total detections. It gauges how well the model recognizes the right objects. It is mathematically defined as seen in Equation 4.

$$\frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Positives (FP)}} \tag{4}$$

Recall (R)

The proportion of true positive detections to the total number of real objects. It shows that the model can locate all pertinent objects. It is mathematically defined as seen in Equation 5.

$$\frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Negative (FN)}} \tag{5}$$

F-1 Score

This is the harmonic mean of precision and recall, which provides a balance between the two.

The precision, recall and evaluation metrics can be seen in Figures 9, 10 and 11.

Precision Metrics for the Training and Test Set

Features (8)	ANN	Random Forest	SVM Linear	SVM RBF	SVM Poly	SVM Sigmoid
Class 1	0.56	1.00	0.53	0.53	0.54	0.50
Class 0	0.51	1.00	0.51	0.51	0.51	0.50
Weighted Avg	0.54	1.00	0.52	0.53	0.52	0.51
Macro Avg	0.54	1.00	0.52	0.53	0.52	0.51

Figure 9. Precision Metrics for the Training and Test Sets

Recall Metrics for the Training and Test Set

Features (8)	ANN	Random Forest	SVM Linear	SVM RBF	SVM Poly	SVM Sigmoid
Class 1	0.04	1.00	0.54	0.47	0.23	0.50
Class 0	0.97	1.00	0.56	0.66	0.81	0.51
Weighted Avg	0.51	1.00	0.52	0.51	0.52	0.51
Macro Avg	0.51	1.00	0.52	0.51	0.52	0.51

Figure 10. Recall Metrics for the Training and Test Sets

Accuracy Metrics for the Training and Test Set

Features (8)	ANN	Random Forest	SVM Linear	SVM RBF	SVM Poly	SVM Sigmoid
Accuracy	0.51	1.00	0.52	0.53	0.52	0.51

Figure 11. Accuracy Metrics for the Training and Test Sets

A. ARIMA Model Performance

The ARIMA (1,1,1) model demonstrated a strong ability to capture linear dependencies and short-term trends within the EUR/USD dataset. Residual analysis showed minimal autocorrelation, indicating a good fit for the linear components of the time series. However, the model struggled to account for non-linear price movements, leading to structured residuals and highlighting its limitations. Performance metrics, including a Mean Absolute Error (MAE) of 0.9766 and a Root Mean Squared Error (RMSE) of 0.9767, reflected these challenges. ARIMA was particularly prone to lagging behind actual data during turning points, a critical shortcoming in financial trend forecasting.

B. ANN Model Performance

The Artificial Neural Network (ANN) proved effective in capturing non-linear patterns inherent in the dataset. Its predictions closely aligned with actual price movements, particularly at turning points where ARIMA underperformed. ANN's precision for Class 1 (0.56) and Class 0 (0.51) indicated balanced performance, with a weighted average precision of 0.54. Recall metrics also highlighted ANN's reliability, achieving values of 0.04 for Class 1 and 0.97 for Class 0, emphasizing its ability to predict the dominant class effectively. Overall, ANN maintained consistent accuracy of 0.51 across both training and test datasets, confirming its generalization capabilities.

C. SVM and Random Forest Models

Random Forest exhibited near-perfect performance on the training set, with precision and recall of 1.00 for both classes. However, its performance on the test set significantly declined, with precision and recall dropping to 0.50, suggesting severe overfitting.

SVM Linear Kernel achieved balanced precision values of 0.53 and 0.51 for Classes 1 and 0, respectively, with consistent accuracy of 0.52 across training and test datasets.

SVM RBF Kernel demonstrated slightly better recall for Class 0 (0.66), but its performance for Class 1 (precision of 0.53, recall of 0.47) indicated challenges in detecting minority class patterns.

SVM Poly Kernel had a high recall for Class 0 (0.81) but significantly lower recall for Class 1 (0.23), reflecting a strong bias toward the dominant class.

SVM Sigmoid Kernel struggled to perform well, with precision and recall values hovering near 0.50, highlighting its inability to model complex relationships effectively.

D. Comparative Observations

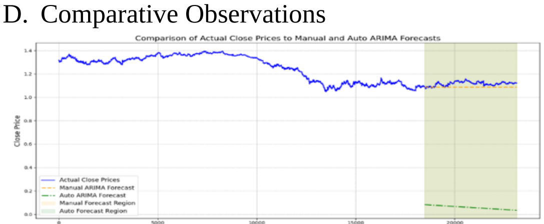

Figure 12. Comparison of Actual Close Prices to Manual and Auto ARIMA Forecasts

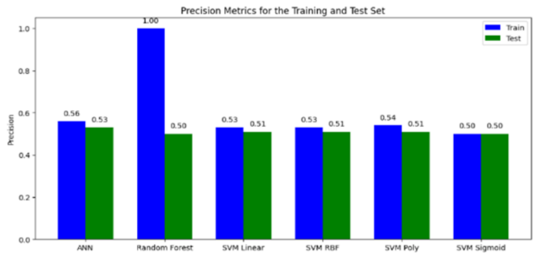

Figure 13. Precision Metrics for the Training and Test Sets

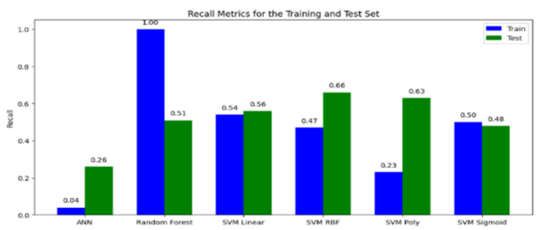

Figure 14. Recall Metrics for the Training and Test Sets.

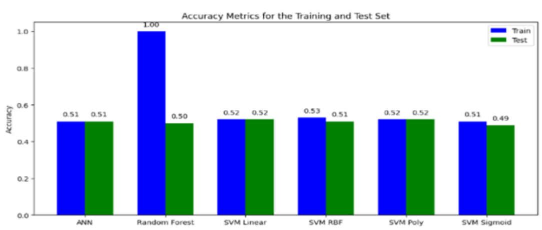

Figure 15. Accuracy Metrics for the Training and Test Sets.

Precision: ANN achieved balanced precision, while Random Forest showed perfect training performance but overfitting during testing. SVM kernels provided moderate precision, with the Linear and RBF kernels showing better balance than Poly and Sigmoid.

Recall: Random Forest dominated recall metrics in training but failed to generalize. ANN and SVM configurations performed consistently, with ANN excelling in capturing the dominant class.

Accuracy: Random Forest's overfitting resulted in a stark accuracy drop from 1.00 (train) to 0.50 (test). ANN and SVM models maintained stable accuracy between 0.51 and 0.53 across both datasets, indicating reliability for trend prediction.

The comparison of the performance of the various model architectures can be seen in Figures 12, 13,14 and 15.

CONCLUSION

The study provides insights into the performance trade-offs between linear and non-linear models for financial trend forecasting. ARIMA demonstrated competence in linear trend detection but struggled with non-linearities, underscoring its limitations for dynamic financial datasets. ANN excelled at capturing non-linear patterns, particularly during abrupt trend shifts, making it a more suitable choice for volatile markets. While Random Forest delivered exceptional training performance, its overfitting highlighted the need for regularization and tuning. SVM kernels, particularly Linear and RBF, offered balanced but moderate performance, suggesting potential with further optimization.

In conclusion, a hybrid modeling approach could leverage ARIMA's strengths in linear trends and ANN's capabilities for non-linear patterns to improve forecasting accuracy. Furthermore, careful parameter tuning and feature engineering for Random Forest and SVM models could enhance their robustness and applicability to financial datasets. These findings emphasize the importance of selecting and customizing models based on the unique characteristics of financial time series data.

LIMITATIONS

Limited computational power restricted the ability to perform extensive feature engineering, preventing the creation of more complex, generalized, and specific features that could have enhanced model performance. Additionally, the analysis was conducted using only 8 features, whereas up to 122 possible features could have been derived through domain knowledge and advanced feature engineering techniques, potentially improving the models' ability to capture intricate patterns in the data. It is important to note that this study is intended strictly for educational purposes and should not be applied to commercial applications yet. Any risks or decisions based on this analysis are the sole responsibility of the user.

REFERENCES

[1] Fuller, N. et al. (2024) What is forex trading? - forex trading "Learn to trade the market, Learn to Trade The Market 'Learn Price Action Trading with Nial Fuller'. Available at: https://www.learntotradethemarket.com/forex-university/introduction-what-is-forex-trading (Accessed: 23 November 2024).

[2] Abbas, S. (no date) 11 best forex indicators all traders should know in 2024 - XS, Discover the 11 best forex trading indicators for 2024. Learn how to use reliable tools like RSI, MACD, and more to enhance your trading strategy. Available at: https://www.xs.com/en/blog/best-forex-indicators/ (Accessed: 23 November 2024).

[3] IBM (2024) What is supervised learning? IBM. Available at: https://www.ibm.com/topics/supervised-learning (Accessed: 23 November 2024).

[4] GeeksforGeeks. (2024, October 10). Support Vector Machine Algorithm. Retrieved from https://www.geeksforgeeks.org/support-vector-machine-algorithm/

[5] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.

[6] https://media.geeksforgeeks.orgwp-content/uploads20201211162942/Capture.JPG

[7] https://media.geeksforgeeks.orgwp-content/uploads20201211181531/Capture.JPG

[8] https://media.geeksforgeeks.orgwp-content/uploads20201211190544/Capture.JPG

[9] https://media.geeksforgeeks.orgwp-content/uploads20201211191138/Capture.JPG

[10] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[11] Ali, Jehad & Khan, Rehanullah & Ahmad, Nasir & Maqsood, Imran. (2012). Random Forests and Decision Trees. International Journal of Computer Science Issues (IJCSI). 9.

[12] Mabrouk, N., Chihab, M., & Chihab, Y. (2022). Trading Strategy in the Forex Market based on Linear and Non-linear Machine Learning Algorithms. 81–85. https://doi.org/10.5220/0010728800003101

[13] https://media.datacamp.com/legacy/image/upload/v1718113325/image_7f309c633f.png

[14] Sherstinsky, A. (n.d.). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. https://www.linkedin.com/in/alexsherstinsky

[15] https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[16] https://miro.medium.com/v2/resize:fit:640/format:webp/1*go8PHsPNbbV6qRiwpUQ5BQ.png

[17] Hayes, A. (2024, July). Autoregressive integrated moving average (ARIMA) prediction model, Investopedia. Available at: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp (Accessed: 22 November 2024).

[18] (No date) What are autoregressive models? - ar models explained - AWS. Available at: https://aws.amazon.com/what-is/autoregressive-models/#:~:text=Autoregression is a statistical technique, function of its past values. (Accessed: 22 November 2024).

[19] Nau, R. (no date) Introduction to ARIMA: Nonseasonal Models. Available at: https://people.duke.edu/%7ernau/411arim.htm (Accessed: 22 November 2024).

[20] https://d1.awsstatic.com/FORMULA33.73e02af03ffa7c17562929db41034d6b048352be.png

[21] Uhlig, S., Alkhasli, I., Schubert, F., Tschöpe, C., & Wolff, M. (2023). A review of synthetic and augmented training data for machine learning in ultrasonic non-destructive evaluation. In Ultrasonics (Vol. 134). Elsevier B.V. https://doi.org/10.1016/j.ultras.2023.107041

[22] E. Soria Olivas (Ed.), Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, IGI Global (701 E. Chocolate Avenue Hershey Pennsylvania 17033 USA), Hershey, PA, 2010.

[23] C.C. Aggarwal Data Classification: Algorithms and Applications CRC Press, Boca Raton, FL (2015)

[24] Author links open overlay panelSaeed Mouloodi a et al. (2022) Feedforward backpropagation

artificial neural networks for predicting mechanical responses in complex nonlinear structures: A study on a long bone, Journal of the Mechanical Behavior of Biomedical Materials. Available at: https://www.sciencedirect.com/science/article/abs/pii/S1751616122000078 (Accessed: 22 November 2024).

[25] Murat H., S. (2006). A brief review of feed-forward neural networks. Communications Faculty Of Science University of Ankara, 50(1), 11–17. https://doi.org/10.1501/commua1-2_0000000026

[26] Nguyen, Thuy & Xuan, Vuong. (2018). FoRex Trading Using Supervised Machine Learning. International Journal of Engineering & Technology. 7. 400. 10.14419/ijet.v7i4.15.23024.

[27] Dautel, Alexander & Härdle, Wolfgang Karl & Lessmann, Stefan & Seow, Hsin-Vonn. (2020). Forex exchange rate forecasting using deep recurrent neural networks. Digital Finance. 2. 10.1007/s42521-020-00019-x.

[28] Macedo, Luis & Godinho, Pedro & Alves, Maria. (2020). A Comparative Study of Technical Trading Strategies Using a Genetic Algorithm. Computational Economics. 55. 10.1007/s10614-016-9641-9.

[29] Huang, Wei & Lai, Kin Keung & Nakamori, Yoshiteru & Wang, Shouyang. (2004). Forecasting Foreign Exchange Rates with Artificial Neural Networks: A Review. International Journal of Information Technology and Decision Making. 3. 145-165. 10.1142/S0219622004000969.

[30] Ayitey Junior, Michael & Appiahene, Peter & Appiah, Obed & Ninfaakang, Christopher. (2023). Forex market forecasting using machine learning: Systematic Literature Review and meta-analysis. Journal of Big Data. 10. 10.1186/s40537-022-00676-2.

[31] Rabbi, Md. Fazle & Hasan, Mahmudul & Dhonno, Fahmida & Sultana, Ayrin & Abedin, Mohammad. (2022). Foreign Currency Exchange Rate Prediction Using Long Short-Term Memory, Support Vector Regression and Random Forest Regression. 10.1007/978-3-030-83799-0_8.

[32] Kamruzzaman, Joarder & Sarker, Ruhul & Ahmad, Iftekhar. (2003). SVM based models for predicting foreign currency exchange rates. 557-560. 10.1109/ICDM.2003.1250976.

[33] Baasher, Areej & Fakhr, Mohamed. (2011). Forex trend classification using machine learning techniques. Proceedings of the 11th WSEAS International Conference on Applied Computer Science. 41-47.

[34] How to trade EUR/USD Forex Online (no date) Saxo. Available at: https://www.home.saxo/learn/guides/forex/how-to-trade-the-eurusd-forex-pair#:~:text=Why%20does%20the%20EUR%2FUSD,is%20their%20respective%20interest%20rates. (Accessed: 23 November 2024).