Machine Learning Application in Finance: Predicting Forex direction using Neural Networks

Dorentin Morina
HEC Lausanne
MSc. in Finance
dorentin.morina@unil.ch

***Abstract*—The Foreign Exchange (Forex) market, renowned as the largest and most liquid financial market globally, facilitates the trading of currencies on a colossal scale, surpassing 7 trillion US dollars in daily transactions. This paper delves into the complexities of Forex trading, where participants ranging from financial institutions to individual traders engage in transactions for various motives, including profit maximization and risk mitigation. However, navigating this extremely volatile market demands sophisticated strategies. Institutions have increasingly turned to algorithmic trading to gain an edge, harnessing machine learning techniques to forecast currency price movements. This study focuses on the implementation of neural network models, including Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), leveraging technical and macroeconomic indicators to predict the direction of EUR/USD. The models are trained and evaluated using daily data to provide insights into the efficacy of neural network-based approaches in enhancing Forex trading strategies.**

***Index Terms*—Forex, FX, Finance, Direction prediction, Technical indicators, Macroeconomic indicators, Neural Networks, LSTM, RNN, GRU**

I. Introduction

The Foreign Exchange market is renowned as the largest and most liquid financial market globally, facilitating the trading of currencies across a range of multiple types of investors, from traditional financial institutions to individual traders. This market's high liquidity and volatility, combined with the easy access to leverage, make this market a primary choice for many traders. The market is also used by corporations for their operations needs, and/or for risk mitigation.

Various methods exist to understand a currency pair's current status comprehensively. For instance, employing technical indicators can reveal its prevailing trend, aiding in identifying optimal entry or exit points for positions. Technical indicators play a prominent role in this market, which is also significantly influenced by traders' psychology, thus exerting considerable influence on market dynamics. Discussing the influence of macroeconomic indicators and economic news on the Forex market is crucial. Negative remarks from central bank officials, disappointing economic indicators like CPI or PPI, and changes in interest rates can significantly impact market conditions. Therefore, traders must stay informed about upcoming indicator releases and important speeches or interventions by key figures. With daily trading volumes surpassing seven trillion US dollars, forecasting currency pair movements or prices has become a central focus for quantitative and academic researchers in recent decades. While various methodologies and algorithms have been explored, there's been a notable surge in utilizing technical and macroeconomic/fundamental indicators. Among these approaches, neural networks have emerged as reliable tools for time series predictions, including in Forex, stocks, and commodities markets.

This paper discusses developing nine distinct models to predict EUR/USD movements using daily data comprising technical and macroeconomic indicators. The subsequent sections are structured as follows: Section 2 reviews relevant literature, while Section 3 elaborates on our research question. Section 4 outlines our methodology, including the models used and parameter selection. Section 5 details the dataset sources and the computation of technical indicators. Section 6 delves into the implementation of the code. Section 7 presents the paper's results, and Section 8 concludes the paper and outlines future research directions.

II. Literature Review

As previously mentioned, various factors can cause price fluctuations or currency pair trends. Constructing a Machine Learning Model to predict a currency pair's trend necessitates incorporating additional data beyond the pair's price.

A. Technical indicators

Using technical indicators is common in Forex prediction, and many papers can confirm their effectiveness. Yıldırım, D.C., Toroslu, I.H., & Fiore (2021) utilized seven different technical indicators (MA10, MACD, ROC, Momentum, RSI, Bollinger Bands, and Commodity Channel Index) in their paper entitled 'Forecasting directional movement of Forex using LSTM with technical and macroeconomic indicators.' The LSTM model trained on this data achieved a profit accuracy of around $52.18\% \pm 1.93\%$.

Ling Qi, Matloob Khushi, & Josiah Poon (2021) employed 28 different indicators in their paper entitled 'Event-Driven LSTM for Forex Price Prediction' and obtained robust results. Their best model on 15-minute interval data for the EUR/GBP pair achieved RME 0.006×10^{-3}, RMSE 2.407×10^{-3}, MAE 1.708×10^{-3}, MAPE 0.194%, surpassing previous studies of that time.

B. Macroeconomic indicators

Dagfinn Rime, Lucio Sarno & Elvira Sojli (2009), in their paper entitled 'Exchange Rate Forecasting, Order Flow and Macroeconomic Information' measured the impact of the macroeconomic announcement (GDP advance, GDP preliminary, Nonfarm payroll employment, unemployment rate, CPI, PMI, Michigan sentiment, etc.) on the EUR/USD, GBP/USD, and JPY/USD.

Yıldırım, D.C., Toroslu, I.H., & Fiore (2021), also trained an LSTM model on macroeconomic indicators (monthly inflation rate for European area, Monthly inflation rate for the US area, monthly interest rate in Germany, monthly interest rate in the European area, Daily Fed funds rates, and the close price of the S&P500 and DAX, the German stock index. The LSTM model trained on this data achieved a profit accuracy of around $50.69\& \pm 1.93\%$

Hamed Naderi Semiromi, Stefan Lessmann & Wiebke Peters (2021), in their paper entitled 'News Will Tell: Forecasting Foreign Exchange Rates based on News Story Events in the Economy Calendar' trained three different models (SVM, Random Forest, and XGB) on news story events in the economy calendar. The FX market is easily affected positively or negatively by macroeconomics and economics releases, and they were among the first to include the economic calendar in an FX prediction study.

III. Research Question

This project aims to develop neural network models for technical indicators, macroeconomic indicators, and a combination of both to determine which provides more useful insights into price direction. The plan is to create datasets containing multiple technical indicators commonly used in the Forex market and relevant macroeconomic metrics such as interest and inflation rates.

This paper's project will adopt a method akin to that used in the Yıldırım, D.C., Toroslu, I.H., & Fiore (2021) study. However, the analysis will not be limited to an LSTM model; instead, it will also encompass a baseline RNN model, an LSTM model, and finally, a GRU model. We want to see which model suits the best for each type of data frame and to see at the end which one gives us the best prediction for the future.

IV. Methodology

The models utilized in this paper are derived from artificial neural networks, which have proven highly effective in various time series prediction tasks. Numerous studies that aimed to predict stock price movements or foreign exchange (FX) price directions have demonstrated promising results with neural network models, particularly Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU). These models are especially well-suited for sequential data because they capture temporal dependencies and patterns.

In this section, we will introduce each of these neural network models in detail and provide a brief explanation of their mechanisms and advantages:

A. Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are a class of neural networks specifically designed to recognize patterns in data sequences, such as time series or natural language. RNNs are versatile and can be applied in various configurations, including one-to-one, one-to-many, many-to-many, and many-to-one. Our study focuses on the many-to-one configuration, where multiple input variables are used to predict a single output value. This approach is particularly relevant for predicting the direction of the EUR/USD price movement based on multiple influencing factors. However, standard RNNs suffer from issues like vanishing and exploding gradients, which can hinder their performance on long sequences.

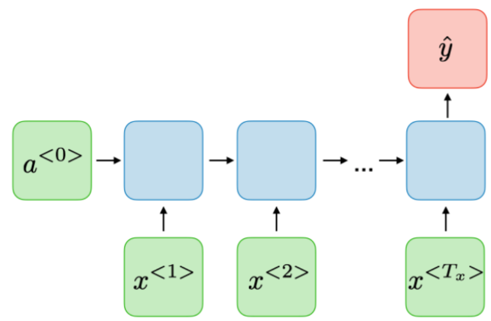

Fig. 1. Architecture of a many-to-one RNN (source: Stanford.edu)

In the many-to-one RNN model, we have a sequence of input data $[x^{(1)}, x^{(2)}, ..., x^{(T)}]$, where T is the length of the sequence, and you want to predict a single output y based on this sequence. At each time step t, the RNN maintains a hidden state called $h^{(t)}$, which captures information from previous time steps and is updated recurrently. It is computed based on the current input $x^{(t)}$ and the previous hidden state $h^{(t-1)}$.

$$h^{(t)} = f(W_h h^{(t-1)} + W_x x^{(t)} + b_h)$$

where W_h is the weight matrix for the recurrent connections, W_x is the weight matrix for the input connections, b_h is the bias vector, and finally, f is the activation function, commonly a non-linear function like hyperbolic tangent (tanh), rectified linear unit (ReLU), or even sigmoid. After processing the entire input sequence, the final hidden state $h^{(T)}$ is used to predict the output y. A loss function measures the discrepancy between the predicted output and the true target value, optimizing the model parameters (weights and biases) with techniques like backpropagation through time or variants of gradient descent to minimize this loss.

B. Long Short-Term Memory (LSTM)

LSTM is a recurrent neural network (RNN) designed to better capture long-term dependencies in sequential data. It is considered an improvement of the simple RNN model. Like a standard RNN, an LSTM model can also be used in a many-to-one configuration where it takes a sequence of input data and produces a single output prediction In an LSTM, there are two

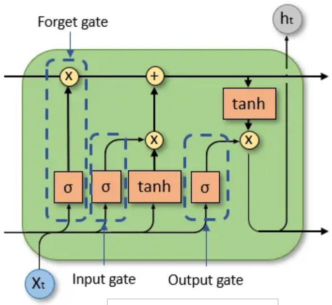

Fig. 2. LSTM Structure

types of states: the hidden states $h^{(t)}$ like in the RNN, but also the cell state $c^{(t)}$. The latter serves as a memory unit, allowing the LSTM to retain information over long sequences. This is not the only new feature of the LSTM, as it also introduces three gates: the input gate, the forget gate, ad the output gate. These gates control the flow of information into and out of the cell state. The model performs several operations at each time step to update the cell and hidden states.

$$f^{(t)} = \sigma(W_f \cdot [h^{(t-1)}, x^{(t)}] + b_f$$

$$i^{(t)} = \sigma(W_i \cdot [h^{(t-1)}, x^{(t)}] + b_i$$

$$\tilde{c} = tanh(W_c \cdot [h^{(t-1)}, x^{(t)}] + b_c$$

$$c^{(t)} = f^{(t)} \cdot c^{(t-1)} + i^{(t)} * \tilde{c}^{(t)}$$

$$o^{(t)} = \sigma(W_o \cdot [h^{(t-1)}, x^{(t)}] + b_o$$

$$h^{(t)} = o^{(t)} \cdot tanh(c^{(t)})$$

where $f^{(t)}$ is the forget gate, $i^{(t)}$ is the input gate, $\tilde{c}$ is the candidate cell state, $c^{(t)}$ is the update cell state, $o^{(t)}$ is the output gate, $h^{(t)}$ is the hidden state update, σ represents the sigmoid activation function, and $tanh$ the hyperbolic tangent activation function. W_f, W_i, W_c, W_o are weight matrices, and b_f, b_i, b_c, b_o are bias vectors.

Like a standard RNN, the final hidden state $h^{(T)}$ is used to predict the output y after processing the entire input sequence. A loss function and a training process are also performed for the standard RNN to minimize the loss between the predicted output and the true target value.

C. Gated Recurrent Unit (GRU)

Gated Recurrent Units (GRUs) are another type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data while addressing some of the limitations of traditional RNNs. Like LSTMs, GRUs also support many-to-one configurations where they take a sequence of input data and produce a single output prediction. In contrast to LSTMs, GRUs have a single hidden state $h^{(t)}$ without an explicit cell state. This simplifies the architecture compared to LSTMs. It introduces two gates: the reset gate

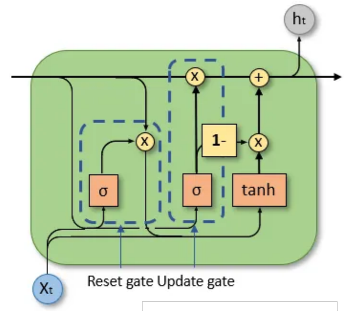

Fig. 3. GRU Structure

$r^{(t)}$ and the update gate $z^{(t)}$. These gates control the flow of information in the network, similar to LSTMs. However, it has a more streamlined architecture with fewer parameters than LSTMs. The final hidden state $h^{(T)}$ is finally used to predict the output y, similar to RNNs and LSTMs.

$$r^{(t)} = \sigma(W_r \cdot [h^{(t-1)}, x^{(t)}] + b_r)$$

$$z^{(t)} = \sigma(W_z \cdot [h^{(t-1)}, x^{(t)}] + b_z)$$

V. DATASET

We decided to base this paper on the EUR/USD FX pair, the most traded currency pair, with almost 28% of the daily volume. The data selected is from January 2019 to end of March 2024. The notebook `data-prep.ipynb` contains all the code for creating the data frame used during this project. It comprises functions that will extract the historical price, compute the technical indicators, and import all the macroeconomic data.

A. Historical Price Dataframe

The main objective of this project is to predict the direction of the price movement of EUR/USD. The EUR/USD prices are needed to pursue this project. As we need to compute some technical indicators later, we need the Open-High-Low-Close (OHLC) price data frame. We import that from Yahoo Finance using the yfinance package.

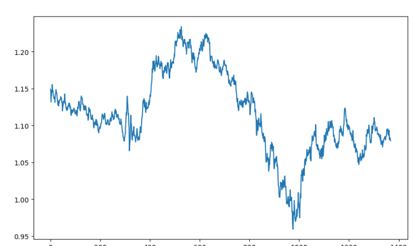

Fig. 4. EURUSD Close price

B. Technical Indicators

The second step was to compute the technical indicators directly with Python. We chose to compute them because there was no real free platform to download them. The function `technical_indicator` was created. It will generate the following list of technical indicators given any OHLC data frame. We selected some of the most used technical indicators in Forex trading.

*1) **Moving Average**:* This technical indicator is very popular in finance. It is used to smooth the price and remove the impact of random and short-term fluctuation. We computed it using a 10-day window.

*2) **Exponential Moving Average**:* It is similar to the Moving Average, except it gives more weight to recent data. It was computed using a 12 and 26-day window.

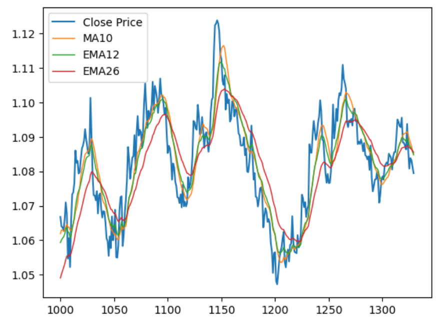

Fig. 5. MA10, EMA12, and EMA26

*3) **Moving Average Convergence/Divergence**:* Traders employ the moving average convergence divergence (MACD) to track the relationship between two moving averages. This involves subtracting a 26-day exponential moving average from a 12-day exponential moving average. It is used as a signal line that helps identify crossovers.

*4) **Average True Range**:* The Average True Range (ATR) is an indicator of market volatility. It computes it by decomposing the entire range of an asset price for that period.

*5) **Relative Strength Index**:* The Relative Strength Index (RSI) is a momentum oscillator. It measures the speed and change of price movements. It lies between 0 and 100. An asset with an RSI above is said to be overbought and oversold when below 30.

*6) **Pivot Point**:* The Pivot Point is an indicator of the market's overall trend. We compute it by taking the average of the previous day's High, Low, and Close prices. If the asset trades above the pivot point, it indicates an ongoing bullish sentiment, while trading below it indicates an ongoing bearish sentiment.

*7) **Commodity Channel Index**:* The Commodity Channel Index (CCI) is a common tool for traders to identify cyclical trends for all assets, not just commodities, as its name might suggest. It measures the difference between the current price and the historical average price.

*8) **Momentum**:* The Momentum is an oscillator indicator of a trend's strength or weakness. It compares previous closing prices.

*9) **Bollinger Bands**:* Bollinger Bands is an indicator that helps traders gauge stock volatility to determine whether they are overvalued or undervalued.

C. Macroeconomic Indicators

The final step was to gather some macroeconomic indicators. All the data was available online, generally on central banks or national statistics offices of countries.

*1) **Inflation rate**:* The CPI measures the price change of goods and services and indicates inflation's level and increase/decrease. The increase of inflation can be equated with a decrease in a currency's buying power, thus leading to a devaluation of the currency.

The concerned countries/regions' Consumer Price Index (CPI) was downloaded via each national bank's website (ECB, FED, etc.). The release date of each CPI was downloaded from investing.com

*2) **Interest rate**:* Interest rate change is a monetary tool that central banks use to implement their monetary policy. By manipulating the monetary supply, cost of borrowing, and availability of money, they can impact the value of one currency against another.

The interest rates of the concerned countries/regions were downloaded online again via each national bank's website.

*3) **Major Indexes**:* The final macroeconomic measure we rely on is the closing prices of major stock indexes in the relevant countries or regions. Stock markets are responsive to economic conditions and trends, making these indexes indicative of a country's economic performance. For the USA, we used the S&P500, the Dow Jones, and the Nasdaq Composite, while for Europe, we used the DAX (Germany), CAC40 (France), and FTSE MIB (Italy). These three countries represent the three biggest economies in the European Union. All the data was downloaded from YahooFinance.

VI. Implementation

In this section, we will outline the process of transforming a dataset from a CSV file into trained machine learning models. This entire project was conducted on the Nuvolos platform, a cloud-based environment. The specifications of the computational resources used are as follows:

- **Processor:** Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz
- **Virtual CPU (vCPU):** 4vCPU
- **Memory RAM:** 16GB

A. Importation

Three dataframes were used during that project. `eurusd_tech.csv`, which contains the close price of the EUR/USD pair and all the technicals indicators. `eurusd_macro.csv` contains the macroeconomic indicators and the close price, while `eurusd_both.csv` contains both technical and macroeconomic indicators. Once we imported them into our environment, it was time make some changes so the machine learning models would understand and would be trained on them. The first was to remove the pandas DataFrame type and have the values in the dataframe. We then convert all the data into float32.

B. Splitting

The second step involved splitting our data into three distinct sets: Training, Validation, and Testing. This division is crucial to ensure the model is well-trained and evaluated properly. The training data is used to fit the model, while the validation data helps monitor the training process and tune the model's hyper-parameters by checking training and validation loss. Finally, the testing data, which the model has never seen before, is used to assess the model's performance and generalization ability. For our project, we decided to test the models on 30 days, validate on 252 days (approximately one year), and train the model with the remaining data.

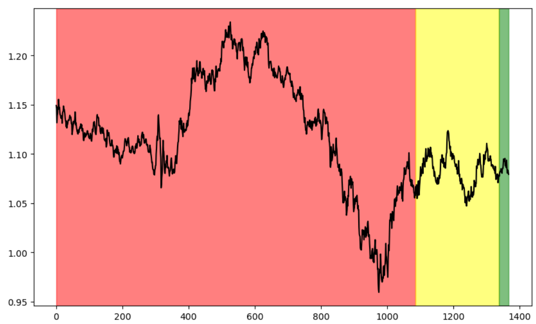

Fig. 6. Training - Validation - Testing Repartition

C. Normalization / Scaling

After splitting the data frame into three categories, we needed to normalize our data frame to improve convergence speed, as normalization helps optimizers converge faster by ensuring that features are on a similar scale. Normalization also avoid that features with larger scales disproportionately influence the model, leading to biases. The data was normalized using the `MinMaxScaler` function of the `sklearn` package. The scaler was initialized with the `train_df`, which contains all the training data to avoid data leakage.

D. Creation of sliding windows matrix

Now that each sub-data frame has been normalized, it is time to construct the sliding window matrix. This step involves creating the X and y matrices, where 'y' represents what we're predicting and 'X' represents the data on which our prediction is based. Typically, predictions are not solely based on the immediately preceding data point, but rather on a sequence of previous data points. I refer to this sequence as `n_past`. For this project, we've chosen 5 as the value for `n_past`. Since a trading week comprises 5 days, we only constructed our models using data from the past week. To illustrate, consider the screenshot below which visually demonstrates this concept. To forecast the price on day 6, we utilize data from days 1 to 5. Likewise, to predict the price on day 7, we utilize data from days 2 to 6, and so forth. This approach resembles a sliding window mechanism.

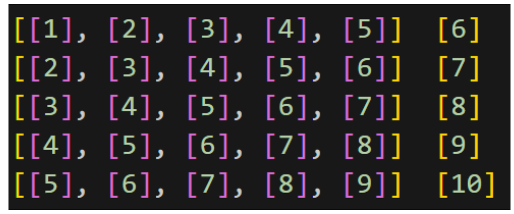

Fig. 7. Sliding Window(n_past=5) - X (left), y (right)

E. Model settings and compilation

As mentioned in the paper, we aim to implement RNN, LSTM, and GRU models to predict the direction of the EUR/USD price. Determining the parameters that will yield an optimal model with the models selected is crucial. Rather than making arbitrary guesses, we implemented a for-loop for each model to systematically find the best parameters for each model and data frame. We arbitrarily selected the following parameters for testing:

- $e = [50, 100]$: the number of epochs
- $bs = [16, 32, 64]$: the batch sizes
- $l = [0.001, 0.01, 0.1]$: the learning rates
- $u = [16, 32, 64]$: the number of hidden units per layer
- $a = [relu, sigmoid, tanh]$: the activation functions

These selections resulted in 1458 possible parameter combinations, each taking at least 5 hours to test per model and data frame. This exhaustive search was essential to identifying the best parameter combination from the chosen set. While further optimization is possible, it would require significantly more computational power and time.

By systematically exploring these parameter combinations, we aim to enhance the predictive accuracy of the models, ensuring that the selected parameters contribute effectively to model performance. This approach underscores the importance of thorough parameter tuning in developing robust predictive models. We finally selected the parameters that achieved the lowest MSE score.

TABLE I
HYPERPARAMETERS - 2 LAYERS RNN/LSTM/GRU

	e	bs	l	u	a
Tech / RNN	100	32	0.01	32 / 16	sigmoid / relu
Macro / RNN	50	16	0.01	32 / 64	sigmoid / relu
Both / RNN	100	32	0.01	32 / 16	sigmoid / relu
Tech / LSTM	50	64	0.1	16 / 16	tanh / sigmoid
Macro / LSTM	100	64	0.1	32 / 16	sigmoid / tanh
Both / LSTM	50	16	0.1	16 / 64	sigmoid / sigmoid
Tech / GRU	100	32	0.001	32 / 64	tanh / sigmoid
Macro / GRU	50	16	0.01	16 / 16	relu / sigmoid
Both / GRU	100	16	0.1	32 / 32	sigmoid / relu

F. Descaling

After making predictions with our models regarding future prices, it's crucial not to overlook another important step. Indeed, to accurately assess the predicted prices, it's imperative to reverse the scaling process applied to both our `y_test` and `y_pred` data. The `y_test` dataset contains the true values we aim to predict, while `y_pred` comprises the values we've forecasted. By descaling these datasets, we restore them to their original scales, enabling more effective comparison and analysis. This step ensures a comprehensive understanding of the predicted prices in their original context.

G. Plotting

After completing all preceding steps, it's beneficial to include some plots to evaluate the performance of our models. This can be achieved by visualizing the training versus validation loss and plotting our predictions against the actual price data. These plots visually represent our model's behavior and provide a quick assessment of its effectiveness. By observing the training versus validation loss plot, we can gauge the model's training progress and identify any signs of overfitting or underfitting. Additionally, comparing our predictions with the actual prices allows us to assess the model's accuracy and identify any discrepancies directly. These visualizations are valuable tools for validating the model's performance and gaining insights into its behavior.

VII. Results

This study aimed to forecast the movement of the EURUSD pair. While accurately predicting the future price remains elusive, forecasting its general direction appears more attainable. We established nine distinct models based on predefined hyperparameters outlined in Section IV-E. Each model underwent training followed by validation using a separate dataset. Subsequently, the models were tested on unseen data to assess their predictive performance.

A. Technical Data

The 2-layers Recurrent Neural Network (RNN) model served as a baseline model and had very low losses during the training and validation phases. The predictions are close to reality. The RMSE score is 0.001001 and the MAE is 0.000843, indicating a good accuracy.

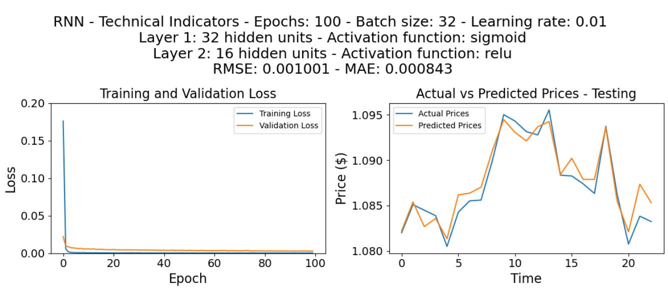

Fig. 8. RNN - Technical Indicators

The 2-layer Long Short Term Memory (LSTM) model also achieved relatively low losses during the training and validation phases but cannot achieve greater accuracy than the baseline model, with an RMSE of 0.00109 and an MAE of 0.000931.

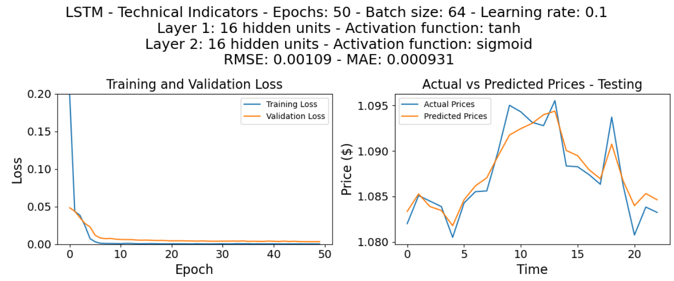

Fig. 9. LSTM - Technical Indicators

The third model, the 2-layer Gated Recurrent Unit (GRU) model, achieved an RMSE score of 0.000845 and an MAE score of 0.000691, which are better than those of the RNN and LSTM models.

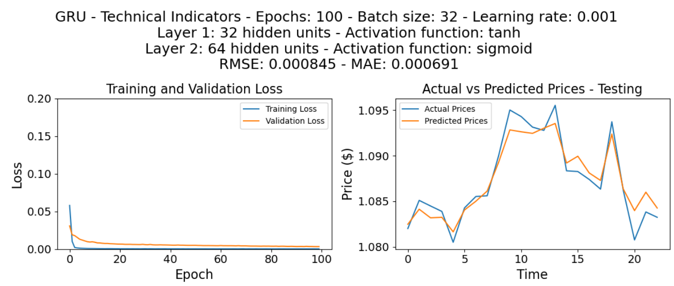

Fig. 10. GRU - Technical Indicators

As a bonus, the three models were tested on three different FX pairs to assess their generalizability and effectiveness across various pairs. This analysis also validated that the models were not overfitted and only useful for EUR/USD data. The CAD/USD, CHF/USD, and GBP/USD pairs were used for this purpose.

The results indicated robust performance across all pairs, although the accuracy was slightly higher for the original EUR/USD pair than the others. Nevertheless, the models demonstrated the potential for providing useful predictions for various currencies, highlighting the applicability of technical indicators in prediction tasks.

Specifically, the GRU model performed best for EUR/USD, GBP/USD, and CAD/USD, while the LSTM model exhibited superior performance for CHF/USD.

TABLE II
RMSE score - Different Currencies

	RNN	**LSTM**	**GRU**
CAD/USD	0.0025	0.00234	**0.00196**
CHF/USD	0.00202	**0.00132**	0.00179
GBP/USD	0.00169	0.00208	**0.00119**
EUR/USD	0.0010	0.00109	**0.00085**

B. Macroeconomic Data

The same procedure was done for the macroeconomic data. The 2-layer RNN model achieved way more accurate prediction with the macroeconomic data with an RMSE score of 0.000801 and an MAE score of 0.000654.

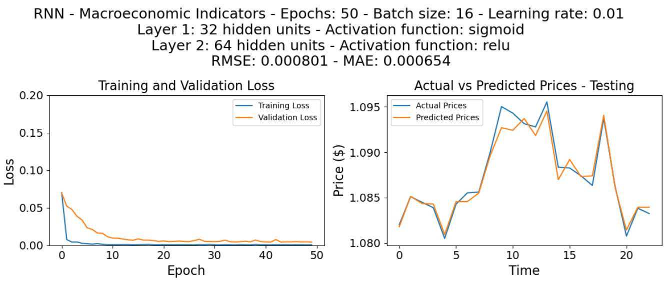

Fig. 11. RNN - Macroeconomic Indicators

The 2-layer LSTM model achieved better scores than the RNN one with an RMSE score of 0.000769, and an MAE score of 0.000567. However, the training and validation losses look less cleaner than before.

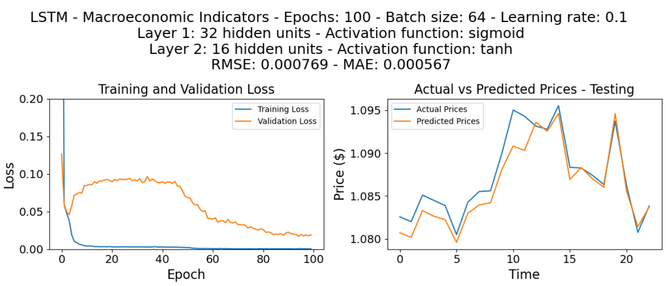

Fig. 12. LSTM - Macroeconomic Indicators

Finally, the 2-layer GRU model achieved again better scores than the two other models with an RMSE score of 0.000624 and an MAE score of 0.000479. The training and validation losses look better with that model.

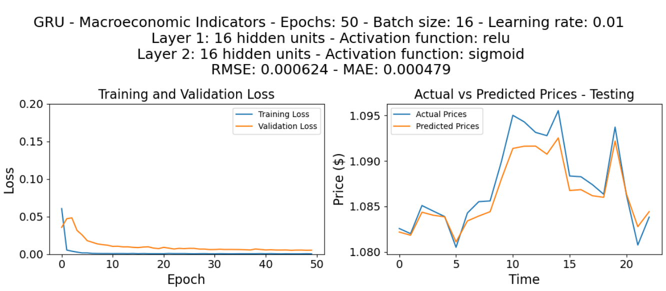

Fig. 13. GRU - Macroeconomic Indicators

C. Both data

The same procedure was used with the data frames containing both indicators. With an RMSE score of 0.001073 and an MAE score of 0.000853, the 2-layer RNN model trained with both indicators can't compete with the RNN model trained with only technical or macroeconomic indicators.

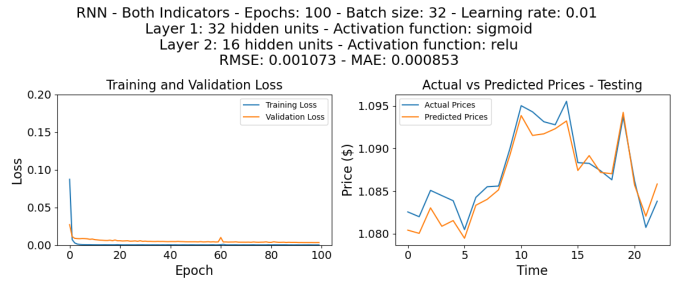

Fig. 14. RNN - Both Indicators

The same conclusion applies to the 2-layer LSTM model, which has an RMSE score of 0.00115 and an MAE score of 0.000929.

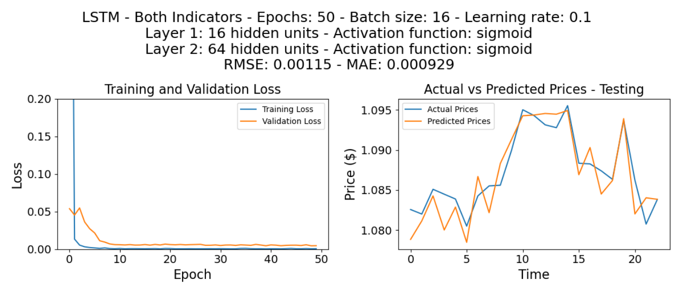

Fig. 15. LSTM - Both Indicators

Once again the 2-layer GRU model achieved an better RMSE score than the two others models, but still can't manage to outperform the 2-layer GRU model trained with different data.

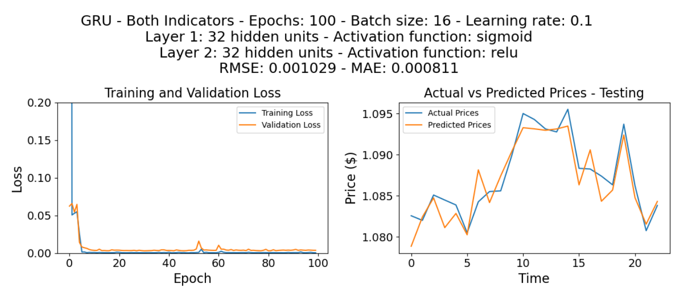

Fig. 16. GRU - Both Indicators

A test involving LSTM and GRU models with other currencies was not conducted due to the absence of comparable number major indices in these countries. Consequently, the disparity in available variables renders it infeasible to apply the same model across different contexts.

D. Direction

Thus far, we've examined the results of price prediction. However, the primary objective of this study isn't to predict the exact price, given its inherent non-stationarity, but rather to forecast its direction. To achieve this, we compared the directional predictions of the models with the actual price movements. Specifically, we assessed when the price increased or decreased and identified which models correctly predicted these movements. Finally, we tallied the correct predictions from each model to determine which one made the most accurate directional forecasts.

TABLE III
DIRECTION PREDICTION ACCURACY

	RNN	LSTM	GRU
Technical Ind.	36.36%	**50.00%**	40.91%
Macro. Ind.	45.45%	**90.91%**	86.36%
Both Ind.	**81.82%**	59.09%	72.73%

The direction prediction achieved using Technical Indicators was the least accurate, with the LSTM model emerging as the most precise, exhibiting a 50% accuracy rate in predicting the correct direction. Remarkably, the LSTM model trained with macroeconomic indicators demonstrated an impressive accuracy rate of over 90% in forecasting price direction. Additionally, the GRU model deserves an honorable mention for achieving accuracy rates of 72% and 86%, which are highly commendable.

VIII. Conclusion

In conclusion, predicting forex prices presents a formidable challenge, with most models demonstrating limited efficacy in real-time applications. This study conducted a comprehensive evaluation on nine distinct models leveraging various neural network architectures and datasets. The results underscore the potential utility of both technical and macroeconomic indicators for forex price prediction, with models trained on macroeconomic data consistently exhibiting lower RMSE errors compared to those relying solely on technical indicators.

Recognizing the inherent non-stationarity of forex markets, the study shifted its focus from precise price prediction to directional forecasting. This strategic pivot enabled the assessment of models based on their capacity to accurately anticipate price movements rather than the precision of their price predictions. Notably, models trained with macroeconomic data consistently outperformed their counterparts in this respect.

An additional finding of note is that the model trained on a merged dataset incorporating both macroeconomic and technical indicators exhibited the highest RMSE score. Despite this, it consistently demonstrated favorable directional predictions, with over 59% accuracy in anticipating price movements. This highlights the potential efficacy of integrating multiple data sources in forex price prediction models, notwithstanding the associated increase in prediction error.

In summary, while the challenge of predicting forex prices remains formidable, the study's findings underscore the potential benefits of incorporating macroeconomic indicators into predictive models. Moving forward, continued exploration and refinement of models leveraging both technical and macroeconomic data hold promise for enhancing forex price prediction accuracy.

Furthermore, avenues for improvement include exploring different technical indicators and macroeconomic indicators, integrating news sentiment analysis, and potentially incorporating sentiment analysis from social media platforms such as X.com. Additionally, conducting extensive parameter tuning could further enhance model performance and robustness. These avenues offer exciting opportunities for advancing the field of forex price prediction and warrant further investigation in future research endeavors.

Appendix

A. *Artifical Intelligence Helper Tools*

At some point, this project (coding, writing, and even the video) was completed with the help of some AI tools.

*1) **ChatGPT 3.5**:* ChatGPT was utilized during this project to assist with various aspects of the work. It helped us address basic questions related to machine learning, correct and debug the code, and provided insights and answers to questions about our methodology and results.

*2) **Copilot**:* Copilot was used as a coding assistant, permitting us to code more efficiently and spend less time on basic tasks like data frame wrangling, plotting, or commenting on the code.

*3) **Grammarly**:* Grammarly was used to write this paper, permitting me to avoid grammar mistakes and structure my thoughts more effectively.

B. *Other Currencies - Actual vs Predicted Prices*

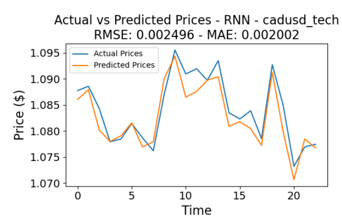

Fig. 17. CAD/USD - RNN - Technical Indicators

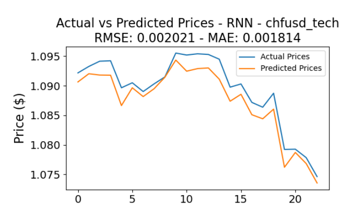

Fig. 18. CHF/USD - RNN - Technical Indicators

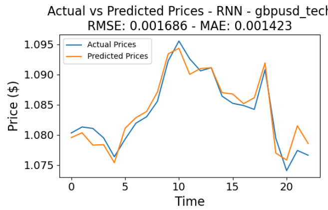

Fig. 19. GBP/USD - RNN - Technical Indicators

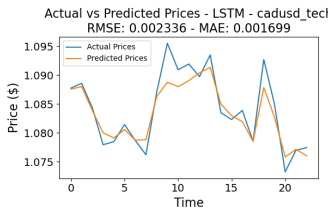

Fig. 20. CAD/USD - LSTM - Technical Indicators

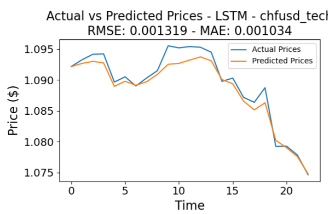

Fig. 21. CHF/USD - LSTM - Technical Indicators

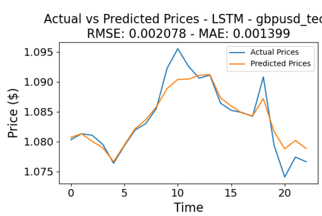

Fig. 22. GBP/USD - LSTM - Technical Indicators

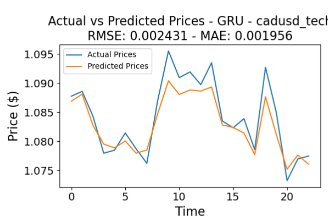

Fig. 23. CAD/USD - GRU - Technical Indicators

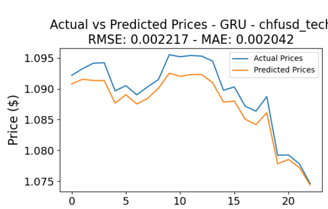

Fig. 24. CHF/USD - GRU - Technical Indicators

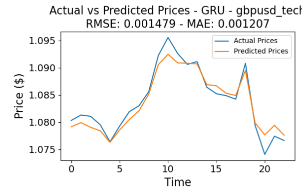

Fig. 25. GBP/USD - GRU - Technical Indicators

REFERENCES

[1] Website: "A beginner's guide to forex trading" https://www.ig.com/en-ch/learn-to-trade/ig-academy/a-beginners-guide-to-forex-trading

[2] Website: "Best indicators for forex trading and how to use them" https://www.ig.com/uk/forex/fx-need-to-knows/forex-indicators

[3] Website: "Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators" https://jfin-swufe.springeropen.com/articles/10.1186/s40854-020-00220-2

[4] Website: "Event-Driven LSTM For Forex Price Prediction" https://arxiv.org/abs/2102.01499

[5] Website: "Cheatsheet - Recurrent Neural Networks" https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

[6] Website: "Understanding LSTM Networks" https://colah.github.io/posts/2015-08-Understanding-LSTMs

[7] Website: "Introduction to Gated Recurrent Unit (GRU)" https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/

[8] Website: "HICP - Overall index, Euro area (changing composition), Monthly" https://data.ecb.europa.eu/data/datasets/ICP/ICP.M.U2.N.000000.4.ANR

[9] Website: "Harmonized Index of Consumer Prices: All Items for United States" https://fred.stlouisfed.org/series/CP0000USM086NEST0

[10] Website: "Eurozone Consumer Price Index (CPI) YoY" https://www.investing.com/economic-calendar/cpi-68

[11] Website: "Download Release Dates for Release: Consumer Price Index" https://alfred.stlouisfed.org/release/downloaddates?rid=10

[12] Website: "Key ECB interest rates" https://data.ecb.europa.eu/main-figures/ecb-interest-rates-and-exchange-rates/key-ecb-interest-rates

[13] Website: "Federal Funds Effective Rate" https://fred.stlouisfed.org/series/DFF

[14] Website: "Moving Average (MA): Purpose, Uses, Formula, and Examples" https://www.investopedia.com/terms/m/movingaverage.asp

[15] Website: "How Is the Exponential Moving Average (EMA) Formula Calculated?" https://www.investopedia.com/ask/answers/122314/what-exponential-moving-average-ema-formula-and-how-ema-calculated.asp

[16] Website: "What Is MACD?" https://www.investopedia.com/terms/m/macd.asp

[17] Website: "Average True Range (ATR) Formula, What It Means, and How to Use It" https://www.investopedia.com/terms/a/atr.asp: :text=Investopedia

[18] Website: "Relative Strength Index (RSI) Indicator Explained With Formula" https://www.investopedia.com/terms/r/rsi.asp

[19] Website: "Pivot Point: Definition, Formulas, and How to Calculate" https://www.investopedia.com/terms/p/pivotpoint.asp

[20] Website: "What Is the Commodity Channel Index (CCI)? How To Calculate" https://www.investopedia.com/terms/c/commoditychannelindex.asp

[21] Website: "What Is a Momentum Indicator? Definition and Common Indicators" https://www.investopedia.com/investing/momentum-and-relative-strength-index

[22] Website: "Bollinger Bands: What They Are, and What They Tell Investors" https://www.investopedia.com/terms/b/bollingerbands.asp