Forex Rate Forecasting Based on Deep Learning Ensembled Predictions.

MSc Research Project
MSc. in Data Analytics

Luis Daniel Medina Salas
x22132627@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Michael Bradford

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Luis Daniel Medina Salas		
Student ID:	X22132627		
Programme:	MSc. in Data Analytics	**Year:**	2024
Module:	MSc. Research Project		
Lecturer:	Michael Bradford		
Submission Due Date:	27th of May 2024		
Project Title:	Forex Rate Forecasting Based on Deep Learning Ensembled Predictions		
Word Count:	4641		
Page Count:	14		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

ABSTRACT

The aim of this paper is to improve foreign exchange rate forecasting by evaluating the capabilities of Recurrent Neural Networks (RNNs), which are known for their ability to analyze sequential data. Four LSTM models with different architectures were created and compared to discover the most suitable architecture based on their performances.

Accurate forecasting of Forex rates is critical across industries, influencing the valuation of foreign investments and driving the dynamics of international trade. Given the complexity and volatility inherent in Forex markets, which are defined by detailed time-series data, traditional forecasting approaches hardly capture the full range of market dynamics and other macroeconomic factors and behaviours.

By overcoming frequent RNN time series predictors that only consider historical data from the predicted variable as input, the models in this paper perform a multidimensional approach that aims to provide a more comprehensive understanding of the factors that influence the Mexican currency (MXN) price against the American dollar (USD). The most distinctive aspect of this approach is the use of the Keras Functional API, which allows more flexible model architectures, including the integration of different input streams to allow for a wider range of factors that influence the MXN. In this case, there are included external macroeconomic variables such as crude oil international prices and the performance of the Mexican stock market.

The provided findings are that by considering the previously mentioned macroeconomic factors correlated to the Mexican currency, into the LSTM models, it was possible to get more reliable predictions with a lower error rate, than the models that do not consider these macroeconomic factors and only consider historical data from the predicted currency.

Keywords:
Forex rate forecasting, time series, Recurrent Neural Network, Keras functional API.

INTRODUCTION

The Forex market has become into one of the most unstable and volatile financial asset markets in recent decades. According to the Bank for International Settlements (BIS) data, the worldwide trading volume in Forex markets was $6.6 trillion per day in April 2019, increasing 30% from April 2016 ($5.1 trillion). Investments in organizations located in different countries and the expansion of globalization have resulted in a substantial rise in the number of Forex transactions. Forex rates have a significant impact on how currency risks and rewards for foreign trading are calculated. To mitigate the risk, the government and policymakers closely monitor currency movements. Therefore, Forex is indeed the most valuable index for international financial markets (Huang et al. [1]).

Price volatility in financial assets is an important concern for researchers, investors, and regulators. Volatility has a clear impact for variable pricing, hedging, portfolio selection, and risk management. As a result, a variety of theoretical and empirical studies have recently been conducted to forecast and simulate volatility in financial assets including the Forex market (Vasilellis and Meade et al. [2]).

Also, macroeconomic factors, such as the inflations and commercial policies play a very important role in the Forex market, as we can find in the paper [3], that the oil price has a high correlation with the Mexican currency, concluding that an increase in oil prices creates an appreciation of the exchange rate. Also, the

Mexican currency and the stock market index share a negative correlation, meaning that an appreciation in the Mexican currency, results in a depreciation in the stock market.

For our research the iShares NAFTARC index prices were used, to feed our models, due to is one of the most important ETFs in Mexico, which is a compilation of stocks of the top 35 biggest companies in the country, providing a good parameter of the stock market in Mexico. As a parameter of crude oil, it was used the West Texas Intermediate (WTI), which is a common oil parameter in the USA, who is the biggest Mexican oil importer.

Vast research has been done about different deep learning models used to predict different financial instruments, or time series data in general. Despite there is plenty of research about correlations between oil prices and different currencies, it is important to consider not all economies are impacted the same way only a small portion has considered introducing the dynamics of macroeconomic factors into a deep learning model for a Mexican Forex prediction. All these factors previously mentioned took us to consider:

'By assembling macroeconomic factors, such as the international oil prices and the Mexican stock market prices, into a deep learning currency price predictor, can we achieve more accurate results than LSTM models that only consider historical data from a single variable to compute their predictions?'

In the following section, Literature Review, we will find previous research that helped to us to understand more about prediction models applied to financial assets and also important findings about macroeconomic factors and their impact into different markets. Followed by the methodology section where will be found the approach used to preprocess the data and create the models, finishing with the evaluation, results, and conclusion.

LITERATURE REVIEW

Considering how unpredictable and nonlinear the forex market is, predicting forex rates is a difficult and complex task. Machine learning (ML) algorithms have been used extensively in recent years to forecast currency rates since they have shown potential in handling noisy, high-dimensional data. In this literature review will be discussed recent studies on machine learning-based FX rate predictions.

1- Deep learning models for FX rate forecasting.

As foreign exchange rates have a direct impact on the revenue of multinational corporations, numerous research have concentrated on projecting foreign exchange rates and have employed ANN models to do so.

In the paper "Foreign exchange rates forecasting with a convolutional neural network," by C. Liu,W. Hou, and D. Liu, (2017), et al.[3], it was used a convolutional neural network to predict the EUR/USD, GBP/USD, and JPY/USD rates (CNN). They provided evidence that this model can handle 2D structural exchange rate data.

In the paper "A new ensemble deep learning approach for exchange rates forecasting and trading," by S. Sun, S. Wang, and Y. Wei, (2020), et al.[4], the authors proposed a novel ensemble deep learning approach based on LSTM and a bagging ensemble learning algorithm. Their empirical findings showed that, in comparison to a conventional LSTM model, their proposed model had much better forecasting accuracy.

In the two previously described papers, it is manifested the use of two different kinds of ANN's. C. Liu,W. Hou, and D. Liu, used a single CNN model, while S. Sun, S. Wang, and Y. Wei used an ensemble between a LSTM model and a bagging learning algorithm, the mentioned ensemble was created not just with the function of forecasting a currency rate but also to outperform a trading strategy. Both authors manifested to have achieved their goals.

In the paper "Comparative Analysis of the Application of Deep Learning Techniques for Forex Rate Prediction," by S. Aryal, D. Nadarajah, D. Kasthurirathna, L. Rupasinghe, C. Jayawardena, (2019), et al.[5], The goal of this paper was to study the ability of three different deep learning models, which were LSTM, CNN, and Temporal Convolution Network (TCN), to predict forex rates, specifically from USD to LKR. The decision to use these models was motivated by their ability to capture temporal dependencies in time series data, a critical aspect of forex rate movements.

The CNN model was identified as the most accurate in predicting the USD/LKR exchange rate, showing its superiority in dealing with financial time series data.

In the paper "A Comparative Study of Deep Neural Network and Statistical Models for Stock Price Prediction," by P. Aithal, M. Geetha, R. Abraham, U. Dinesh Acharya, R. Sagarthey, et al.[6], they compared the prediction abilities of LSTM and CNN deep learning models, with standard statistical models such as ARIMA for stock price forecasting. The study aimed to show which models provided more accurate predictions.

The results showed that CNN outperformed LSTM and ARIMA. They attribute this advantage to CNN's ability to focus on the most recent information, which is critical in the current volatile stock market.

The papers include some weaknesses such as the potential overfitting with deep learning models and the assumption that past stock price trends can accurately predict future behaviours, which may not always be applicable.

The approach of the paper [7], aims to improve the predictability of financial forecasting, by incorporating symbolic processing for a better capture of the noise and seasonality in not stationary financial data. The results show that combining RNNs with symbolic processing can successfully minimize the effects of noise while improving accuracy in complex financial predictions.

The next papers [8] focus into the relationship between different macroeconomic factors and how their behaviours can impact in each other. Determinants of the Long-Term Correlation between Crude Oil and Stock Markets, This paper discovered a significant long-term conditional connection between oil prices and stock markets, which has a positive correlation with the risk-free rate and a negative one to the economic activity and credit risk. These findings highlight the stock market susceptibility to oil variations, as well as the importance of macroeconomic factors in producing these behaviours. One of the limitations in this paper is the short dataset giving results which may not be considered reliable when needed to replicate this same approach in bigger and more complex data.

'On the impacts of oil price fluctuations on European equity markets'[9]: Volatility spillover and hedging effectiveness The authors in this paper highlight methodological innovations in the research of oil price impact to other macroeconomic factors, by developing simple present value models to more complex models such as GARCH and VAR models, which provide a more detailed study of the volatility in the stock market and it responses to oil price variations. At the beginning, this paper discusses cases of study on various locations along the world, but especially focuses on the absence of valuable research addressing Europe, which is the main gap this research tries to fill.

RESEARCH METHODOLOGY & SPECIFICATIONS

In this research it was used the CRISP-DM methodology over other methodology approaches like KDD, due to the data approach and manipulation that is required to achieve the objective. This methodology was chosen due to its structured approach for planning, executing, and evaluating data mining projects, providing a clear roadmap. The CRISP-DM methodology consists of six main steps which are, the business understanding, data understanding, data preparation, modelling, evaluation, and deployment.

DATA UNDERSTANDING.

In this section, it is shown a brief description of the data employed for the modules. Three different datasets were downloaded from [10], consisting in historical data of daily prices of MXN and USD, daily prices of the Crude Oil WTI, and the iShares NAFTRAC daily prices. Each dataset contains the following columns: 'Date', 'Open', 'High', 'Low', 'Close', 'Adj. Close' and 'Volume'. The datasets contain daily observations of a twenty-year period, specifically from the 1st of January 2004 to the 1st of January 2024. For research purposes, only the closing prices of each data set were used as the daily 'close' values, due to their more consistent behaviour of the market sentiments, compared to the high, low, and opening values.

DATA CLEANING AND PREPARATION.

In this section, it is explained the steps and techniques used to clean and prepare the data to be suitable for the LSTM models.

Firstly, for a fast and simple overview of the 3 mentioned datasets (Fig.1), features such as mean, count, standard deviation, minimum, maximum, 25 percentile, 50 percentile, and 75 percentile were computed for each dataset, once these values were obtained, there were discover discrepancies in the number of observations between the datasets. To start with the cleaning process, initially, the 'Date' and 'Close' features were extracted from each dataset and kept in three different data frames, so it was possible to clean each time series data separately.

	Open	High	Low	Close	Adj Close	Volume
count	5210.000000	5210.000000	5210.000000	5210.000000	5210.000000	5210.0
mean	15.224793	15.310365	15.144382	15.225697	15.225697	0.0
std	3.763476	3.792595	3.741213	3.763182	3.763182	0.0
min	9.866500	9.933700	9.834000	9.866500	9.866500	0.0
25%	11.774100	11.835700	11.745850	11.782750	11.782750	0.0
50%	13.495160	13.610200	13.381000	13.503350	13.503350	0.0
75%	18.945140	19.057650	18.857808	18.946193	18.946193	0.0
max	25.315100	25.765341	24.728029	25.336201	25.336201	0.0

Fig 1. MXN vs USD price data frame summary.

Initially, it was expected that the discrepancies in the number of observations between the datasets were a consequence of the missing values in the data frames. Having null values in the MXN vs USD data frame and also in the iShares NAFTRAC data frame. After visualizing a dispersion plot of the missing values (Fig.2 and 3), it was discovered that the null values were dispersed in small gaps along the time series data. So, it was decided to use the forward fill method to handle these missing values, considering precious values could provide a feasible estimate for the forward values.

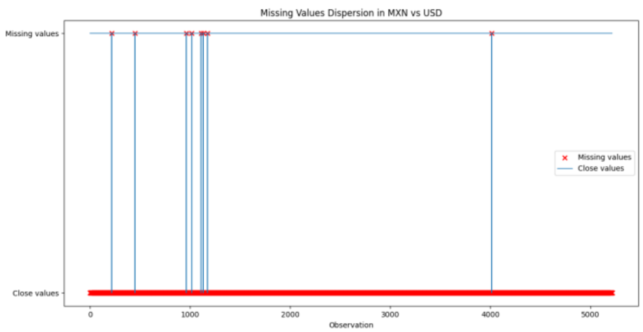

Fig 2. Missing values MXN vs USD.

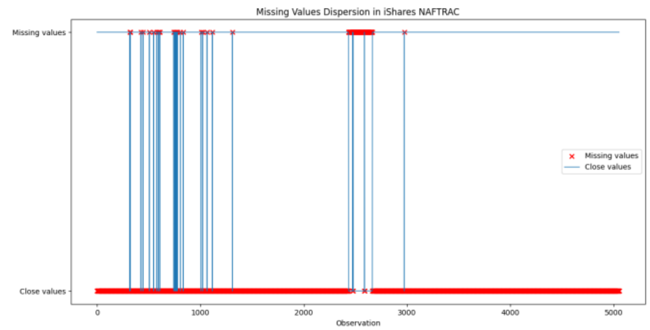

Fig 3. Missing values iShares NAFTRAC

Then, after visualizing a boxplot of the Crude Oil WTI data frame (Fig.4), an outlier was found. Finally, the outlier was replaced with the closest next value inside the 25th percentile range.

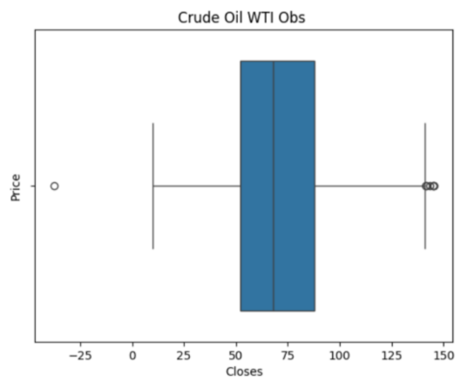

Fig 4. Boxplot Crude oil observations.

Finally, the three data frames were merged into a single data frame using the MXN vs USD ‘Date’ values as reference index, after doing this merging, it was possible to find that there were some new null values in this single new data frame, this can be attributed to the fact that as our three data sets belong to different financial markets, they may not be perfectly be aligned due to the difference of trade market days and even different bank holidays. To handle these null values the forward fill method was used again, but also the backward fill method, due to the missing values at the be very beginning of the Crude Oil WTI observations period.

Finally, a cleaned and merged data frame (Fig.5), was achieved, with 5,218 daily price observations and 25 attributes Fig. Critical parameters like the mean and standard deviation did not suffer considerable variations after cleaning compared to the initial mean and standard deviation.

	Close	Close_MXN_MARKET	Close_CrudeOil
count	5218.000000	5218.000000	5218.000000
mean	15.220540	37.052566	70.126451
std	3.764126	12.174708	22.394433
min	9.866500	8.800000	10.010000
25%	11.756250	29.702500	52.185000
50%	13.494760	40.799999	68.095000
75%	18.944512	45.889999	87.925000
max	25.336201	57.750000	145.290000

Fig 5. Merged dataset summary.

For the data preparation, the process began by applying an Augmented Dickey-Fuller test (ADF), to check for stationarity in data frame features, as stationary time series data shows more consistent statistical attributes over time, it is easier for the models to do a better analysis. After applying the ADF test it was found that the MXN vs USD prices and the iShares NAFTRAC prices are not stationary. After these results, it was decided to turn the data into stationary data by applying the differencing method, which consists of calculating the difference between consecutive observations, the linear trends can be removed.

After achieving stationarity in our data, we faced a concern, since the fact that after our data transformation, the correlation between the attributes was reduced significantly (Fig.6). This commonly happens after doing a data transformation as the one was done in our data preparation, but considering one of the peaks of this research is to implement into the models, important associated data that can help to get a reliable forex predictor, it was decided to continue with the original data. Regardless of the not stationary in two features of our data, the LSTM is well known for being able to handle data where trends and patterns are present, Fig 6.

```
Correlation Matrix:
                          Close  Close_MXN_MARKET  Close_CrudeOil  Close_diff
Close                  1.000000          0.722298       -0.332747    0.014512
Close_MXN_MARKET       0.722298          1.000000        0.182460   -0.011541
Close_CrudeOil        -0.332747          0.182460        1.000000   -0.015174
Close_diff             0.014512         -0.011541       -0.015174    1.000000
Close_MXN_MARKET_diff -0.006753          0.007828       -0.014819   -0.256980

                       Close_MXN_MARKET_diff
Close                              -0.006753
Close_MXN_MARKET                    0.007828
Close_CrudeOil                     -0.014819
Close_diff                         -0.256980
Close_MXN_MARKET_diff               1.000000
```

Fig 6. Correlation values of merged data frame.

Then, the feature engineering techniques of lag features and rolling windows were applied to the merged data frame. The lag features correspond to 1 day, 5 days, and 20 days periods for the 3 variables, as the observations on the datasets correspond to daily prices from Monday to Friday trading days, a week period is equivalent to 5 observations and a month period corresponds to 20 observations approximately. This technique created a new column for each lag period for the prices of each variable.

For the rolling window, window sizes of 5 and 20 observations were applied, again for the 3 variables, for the same reason as the lag features, creating 2 new columns for each variable, with the mean and standard deviation along the window sizes. Adopting these 2 techniques, the LSTM models can have a better understanding of patterns and trends in the data and achieve better performances.

Then main data frame was scaled, so all values would be between 0 and 1, by doing the scaling it possible to optimize the model learning process, avoiding that the largest values in the data frame would be dominant compared to the small values.

Finally, the consolidated data frame was divided into a training data set and a testing data set, with a ratio of 70% to 30%.

MODELLING.

In this section, there is a description of the modelling and configuration of the LSTM models used in this research.

General features:
The four models considered in this research, used 10 steps per sequence as input for the LSTM layers, while the final output was 1 step, meaning that the past 10 observations are used to predict the next day's price. The models were trained with 100 epochs, implementing Early stopping as a callback function in Keras was implemented in all the models, in order to reduce the risk of overfitting and use only the necessary number of epochs, by stopping the training if the loss does not improve after 10 epochs (patience of 10). The models were also fitted with the 'Adam' optimizer algorithm which combines the best properties of the AdaGrad and RMSProp algorithms, minimizing the loss function during the training and Mean Squared Error (MSE) was chosen as the loss function parameter. Finally, a validation spit of 0.2 was implemented, meaning that 20% of the training data was used to evaluate the model while it was in the training process. Also, all models were tested with a batch size of 8.

Baseline LSTM model.

Firstly, a baseline model was created with the purpose of comparing it against the other LSTM models with ensembled predictions into the architecture. This model only used the MXN vs USD values and its features, as input data for the training and predictions. The model is formed by a single LSTM layer with 50 neurons, followed by a dense layer with a single neuron at the end to compute the single predicted value. The model architecture is shown in Fig.7 closing price next day.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_layer (LSTM)           (None, 50)                11800

 output_layer (Dense)        (None, 1)                 51

=================================================================
Total params: 11851 (46.29 KB)
Trainable params: 11851 (46.29 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig 7. Baseline LSTM Model architecture

LSTM Model No.1

This model uses the merged data frame that contains the MXN vs USD prices, the iShares NAFTRAC prices, and the Crude Oil WTI prices, as single input for a LSTM layer with 50 neurons, followed by a dense layer with a single neuron at the end to compute a predicted value. The model architecture is shown in Fig.8.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 50)	15000
dense (Dense)	(None, 1)	51

Total params: 15051 (58.79 KB)
Trainable params: 15051 (58.79 KB)
Non-trainable params: 0 (0.00 Byte)

Fig 8. LSTM 1 Model architecture

LSTM Model No.2

This model is a multi-input LSTM model that uses 3 different data frames (MXN vs USD prices, the iShares NAFTRAC prices, and the Crude Oil WTI prices), with the aim of using each data frame as input for a different LSTM layer, being 3 LSTM layers in total. Each LSTM layer has 50 neurons. Once the LSTM layers processed the data, the results were ensembled into a unified feature by a concatenate layer, then a dense layer with 50 neurons followed the concatenate layer, ensembled and transformed the features extracted from the previous layers, to feed a final single neuron dense layer to compute a predicted value. The model architecture is shown in Fig.9.

Layer (type)	Output Shape	Param #	Connected to
mxn_usd_input (InputLayer)	[(None, 10, 8)]	0	[]
oil_input (InputLayer)	[(None, 10, 8)]	0	[]
market_input (InputLayer)	[(None, 10, 8)]	0	[]
mxn_usd_lstm (LSTM)	(None, 50)	11800	['mxn_usd_input[0][0]']
oil_lstm (LSTM)	(None, 50)	11800	['oil_input[0][0]']
market_lstm (LSTM)	(None, 50)	11800	['market_input[0][0]']
merge_layer (Concatenate)	(None, 150)	0	['mxn_usd_lstm[0][0]', 'oil_lstm[0][0]', 'market_lstm[0][0]']
dense_layer (Dense)	(None, 50)	7550	['merge_layer[0][0]']
output (Dense)	(None, 1)	51	['dense_layer[0][0]']

Total params: 43001 (167.97 KB)
Trainable params: 43001 (167.97 KB)
Non-trainable params: 0 (0.00 Byte)

Fig 9. LSTM 2 Model architecture

LSTM Model No.3

Just as LSTM Model No.2 is a multi-input LSTM model that uses 3 different data frames (MXN vs USD prices, the iShares NAFTRAC prices, and the Crude Oil WTI prices). This model has 6 LSTM layers (50 neurons per layer), meaning 2 LSTM per data frame, which are designed to extract temporal features from the data, enabling the model to adapt and learn from these features. A LSTM Model No. 2, once the LSTM layers processed the data, the results were ensembled into a unified feature by a concatenate layer, then a dense layer with 50 neurons follows the concatenate layer, combining and transforming the features extracted from the previous layers, to feed a final single neuron dense layer to compute a predicted value. The model architecture is shown in Fig.10.

```
Layer (type)                  Output Shape        Param #   Connected to
==========================================================================================
mxn_usd_input (InputLayer)    [(None, 10, 8)]     0         []

oil_input (InputLayer)        [(None, 10, 8)]     0         []

market_input (InputLayer)     [(None, 10, 8)]     0         []

mxn_usd_lstm_1 (LSTM)         (None, 10, 50)      11800     ['mxn_usd_input[0][0]']

oil_lstm_1 (LSTM)             (None, 10, 50)      11800     ['oil_input[0][0]']

market_lstm_1 (LSTM)          (None, 10, 50)      11800     ['market_input[0][0]']

mxn_usd_lstm_2 (LSTM)         (None, 50)          20200     ['mxn_usd_lstm_1[0][0]']

oil_lstm_2 (LSTM)             (None, 50)          20200     ['oil_lstm_1[0][0]']

market_lstm_2 (LSTM)          (None, 50)          20200     ['market_lstm_1[0][0]']

merge_layer (Concatenate)     (None, 150)         0         ['mxn_usd_lstm_2[0][0]',
                                                             'oil_lstm_2[0][0]',
                                                             'market_lstm_2[0][0]']

dense_layer (Dense)           (None, 50)          7550      ['merge_layer[0][0]']

output (Dense)                (None, 1)           51        ['dense_layer[0][0]']

==========================================================================================
Total params: 103601 (404.69 KB)
Trainable params: 103601 (404.69 KB)
Non-trainable params: 0 (0.00 Byte)
```

Fig 10. LSTM 3 Model architecture

RESULTS AND EVALUATION

In this section will be shown the performances and results of the four prediction models.

All models were evaluated on test data using the next metrics:
Mean Absolute Error (MAE): It measures the average of the absolute difference between the actual values and the predicted values, and all the individual errors have the same weight.
Mean Squared Error (MSE): It measures the average of the squared difference between the actual values and the predicted values, giving more weight to larger errors.
Root Mean Squared Error (RMSE): This is the squared root of the MSE, it gives the results in the same units as the predicted variable, in this case, MXN.

Figs. 11, 12, 13, and 14, show the training loss and validation loss of the Baseline LSTM model, LSTM 1, LSTM 2, and LSTM 3 respectively. It is possible to see that the training loss in the four models tends to decrease over time, indicating the model is learning, despite this, it is possible to see many spikes along the validation loss, indicating the model's performance was more volatile through those epochs. Models 1 and 2 seem to have smaller spikes than the others, while the baseline model has more severe spikes and the training and validation loss lines, do not seem to converge as much as in the other 3 models.

Fig.11. Training loss – Baseline LSTM Model.

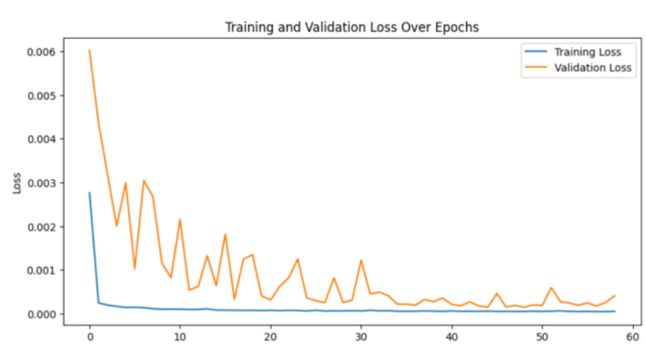

Fig.12. Training loss – LSTM Model No.1.

Fig.13. Training loss – LSTM Model No.2.

Fig.14. Training loss – LSTM Model No.3.

Figures 15, 16, 17, and 18 show the predictions of the models on test data. The four models show good performances. There is not any important gap between the actual and predicted values. Only Model 3 shows some small gaps, but in general, the predictions still being fairly close to the actual values.

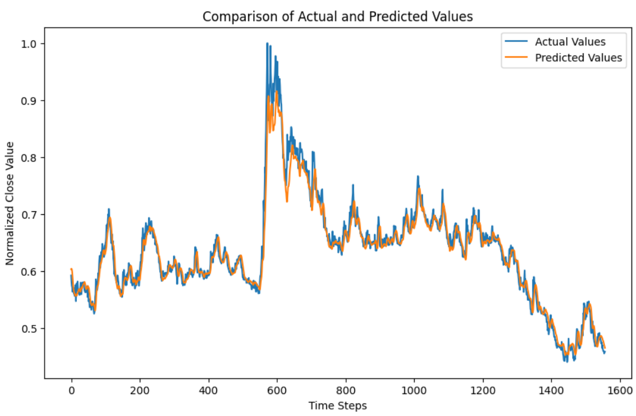

Fig.15. Baseline LSTM Model Predictions.

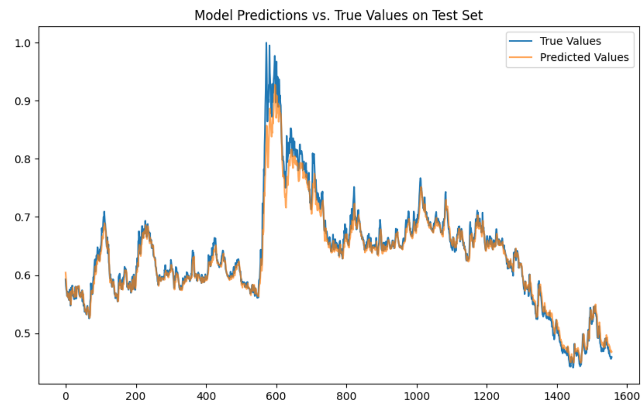

Fig.16. LSTM Model No.1 Predictions.

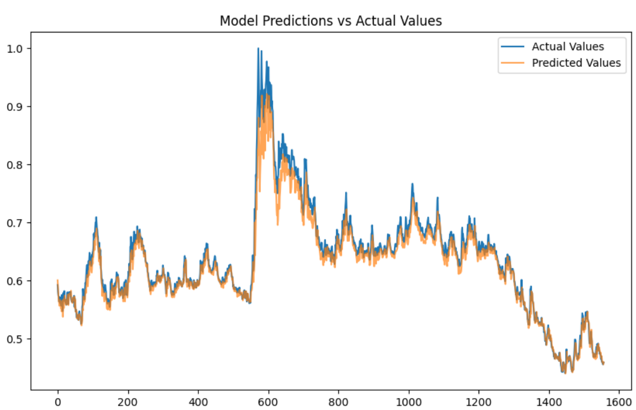

Fig.17. LSTM Model No.2 Predictions.

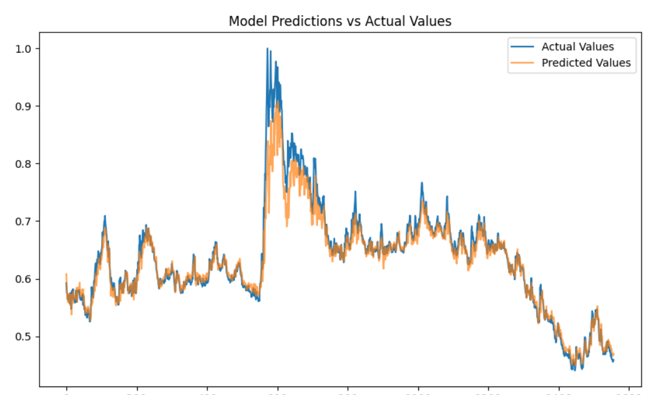

Fig.18. LSTM Model No.3 Predictions.

In the table, Fig.19, there are shown the metric evaluation results. The table shows how Model 2 has the lowest MSE, which indicates it has the smallest average of the squares of the errors so and considering the magnitude of the error. Also, Model 2 has the smallest RMSE, indicating that when considering the magnitude of errors it has the least. The model with the lowest MAE is model 1, indicating in the average, it has the lowest absolute errors in average. Despite being the more complex model, with a larger number of LSTM layers, Model 3 in comparison with the other models, has the worst performance, with the largest MAE, MSE, and RMSE, this might be attributed to the larger number of LSTM layers, while more layers result in a higher learning capacity, it might also lead to an overfitting by over learning from the training data, as a model architecture is more complex, it requires more complex techniques to setup the hyperparameters.

MODEL	MAE	MSE	RMSE
Baseline	0.0129	0.0004	0.0198
Model 1	0.009427074	0.0002424	0.015569102
Model 2	0.010030028	0.00023494	0.015327825
Model 3	0.015526705	0.00082493	0.028721608

Fig.19. Compilated results table.

% of Improvement vs Baseline Model			
MODEL	MAE	MSE	RMSE
Model 1	27%	39%	21%
Model 2	22%	41%	23%
Model 3	-20%	-106%	-45%

Fig.20. Models' improvement percentage.

CONCLUSIONS AND FUTURE WORK.

The goal of this research was achieved, because it was possible to compare different LSTM models with different architectures, and more importantly, to compare different multi-input models against single-input models. As we can see in the table.20 the end Models 1 and 2 could overcome the baseline model results, both getting a lower MSE by 39% and 41% than the baseline model respectively. It is important to remark that LSTM model 1 is a single input model, but it considered the iShares NAFTRAC and Crude Oil WTI into this single input, while model 2 has ensembled predictions from different LSTM layers. Finally, we can say that assembling these macroeconomic factors into the models, and learning from their features and trends, had a clear impact on the results, improving the performances in Forex forecasting. The results of this paper aim to provide valuable insights for investors, financial analysts, and regulators, hence improving decision-making methods in international investment and trade.

For future works, other macroeconomic factors, such as inflation rate or interest rates, can be implemented as inputs to the models and test how they affect the performances. Other popular Deep learning techniques can be applied such as Convolutional Neural Networks (CNN) and Temporal Convolutional networks (TCN). Different optimizers, hyperparameters, and architectures for the LSTM models can be implemented. A different crude oil standard instead of the WTI one, can be implemented. Even this same approach of ensembled predictions can be applied to different financial instruments, such as stock market equities and exchange-traded funds (ETFs).

REFERENCES

1. W. Huang, K. K. Lai, Y. Nakamori, and S. Wang, “Forecasting foreign exchange rates with artificial neural networks: a review,” International Journal of Information Technology & Decision Making, vol. 3, no. 1, pp. 145–165, 2004.
2. G. A. Vasilellis and N. Meade, “Forecasting volatility for portfolio selection,” Journal of Business Finance & Accounting, vol. 23, no. 1, pp. 125–143, 1996.
3. C. Liu, W. Hou, and D. Liu, “Foreign exchange rates forecasting with convolutional neural network,” Neural Processing Letters, vol. 46, no. 3, pp. 1095–1119, 2017.
4. S. Sun, S. Wang, and Y. Wei, “A new ensemble deep learning approach for exchange rates forecasting and trading,” Advanced Engineering Informatics, vol. 46, Article ID 101160, 2020.
5. Aryal, S., Nadarajah, D., Kasthurirathna, D., Rupasinghe, L. and Jayawardena, C., 2019, December. Comparative analysis of the application of Deep Learning techniques for Forex Rate prediction. In *2019 international conference on advancements in computing (ICAC)* (pp. 329-333). IEEE.
6. Aithal, P.K., Acharya, U.D., Geetha, M., Sagar, R. and Abraham, R., 2022, May. A comparative study of deep neural network and statistical models for stock price prediction. In *2022 3rd International Conference for Emerging Technology (INCET)* (pp. 1-5). IEEE.
7. Giles, C.L., Lawrence, S. and Tsoi, A.C., 2001. Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine learning*, *44*, pp.161-183.
8. Yang, L., Yang, L., Ho, K.C. and Hamori, S., 2019. Determinants of the long-term correlation between crude oil and stock markets. *Energies*, *12*(21), p.4123.
9. Arouri, M.E.H., Jouini, J. and Nguyen, D.K., 2012. On the impacts of oil price fluctuations on European equity markets: Volatility spillover and hedging effectiveness. *Energy Economics*, *34*(2), pp.611-617.
10. Datasets sources.
    https://finance.yahoo.com/quote/MXN%3DX
    https://finance.yahoo.com/quote/NAFTRACISHRS.MX?.tsrc=fin-srch
    https://finance.yahoo.com/quote/CL=F?.tsrc=fin-srch