Comparing the performance of Long Short Term Memory NN model and ML Classifiers for Predicting the Trend in the Foreign Exchange Market

Sean Gee Zhing, Zhou Zijian, Joe Teddy Yosua Sanjaya,
Huang Linhang, Cao Yuchen

***Abstract*— Foreign Currency Exchange Market (Forex) is a highly volatile complex time series for which predicting the daily trend is a challenging problem. In this study, we investigate the performance of two proposed models — logistic regression of multiple ML classifiers and Long Short Term Memory NN model — in predicting the price direction in the foreign exchange market. The first model will tackle the prediction as a binary classification problem, with uptrend and downtrend outcomes. A large number of basic features driven from the time series data, including technical analysis features are generated using multiple history time windows. Feature selection and extraction are used to find the best subsets for the classification problem. The second model will tackle the prediction as a regression problem, forecasting the short-term trend in the currency Forex using Deep Learning and Reinforcement Learning Algorithm. Six important Forex currency pairs are investigated and the results show mixed results in the daily prediction and in the expected profit through applying a simple trading strategy.**

I. Introduction

This study is about comparing the performance of two different machine learning techniques in predicting the Foreign Exchange (Forex) market trend with the aim of sustainable long-term profits.

The first technique — Classifiers — will treat the problem as a binary classification task, thus we are not trying to predict the actual exchange rate value between two currencies, but rather, if that exchange rate is going to rise or fall. Our trading strategy then is to take one action per day, where this action is either buy or sell.

The second technique — Long Short Term Model —will treat the problem as a regression task, thus we are predicting the price of an exchange rate. From the price we can get the Alpha — strategy's ability to beat the market — which will decide on how much money to buy or sell with respect to the current available assets that we have [1]. Our trading strategy then is to buy or sell at a specific amount that is based on the Alpha.

As attractive as it may sound, we faced many challenges to reach a combination of predictions and trading strategy that are profitable. Major economic downturn is one of the things that exacerbates the performance of the model. In this paper, we look to understand more about machine learning algorithms and trading strategies performance in the highly-volatile forex market.

II. Related Work

Existing research on related topics is important to be identified before discussing the methodology. The use of machine learning techniques and technical analysis on Forex trading has been studied in some previous works [2][3], particularly on how machine learning and quantitative analysis can be used to predict the movements of Forex prices. Long Short Term Memory, a kind of Recurrent Neural Network model which is one of the typical deep learning model, has also been studied in research on how it can be used to do trend prediction and forecasting in Forex trading [4][1].

III. Methodology

The FX pairs that we are investigating are EURUSD, GBPUSD, USDCHF, USDCNY, USDJPY, USDSGD.

A. Data Cleansing

The first part of the research involves collecting and cleaning raw data. We collect the raw data of each FX pair from the Bloomberg Terminal. The data includes "Date", "Open", "High" and "Low". We clean the data by forward-filling NaN values using iloc.fillna. For calculation of technical indicators, we add a new column "Close" for each trading day and fill it in with the opening price of the next day (note that FX markets trade on a 24 hour basis). We also convert the "Date" into "Days since Starting Date". We then label the data with 1 and 0 depending on whether the price goes up or down the following day (1 if the price goes up and 0 if the price goes down, we use the closing price to compare). This helps us quantify the performance of the FX and enable us to trade based on it[2].

B. Feature Engineering

The second part of the research is feature engineering and feature selection. Some of the features that we will generate include momentum technical indicators such as stochastic oscillator, Williams' R indicator, Rate of Change indicator, MACD indicator. Trend technical indicators such as the EMA indicator, and Commodity Channel Index are generated too. We also added some signal processing features such as the slope, average of past two days high, average of past two days low, and average of previous day's high and low [5].

C. Classifiers

The third part of the research involves using classifiers to predict the direction of the price for the forex pair, that is, whether it will go up or down. We do so by training three classifiers on the feature matrix and the labels. The classifiers selected are Support Vector Machine, Random Forest, and XG Boost. For Random Forest and XG Boost, we train each model twice; the first one is to identify the top 10 important features. On the second run, we then train the three classifiers on these features only to increase training speed and also prevent overfitting. The reasoning behind choosing these three classifiers is that they use different algorithms to classify the data, hence they each have their own strengths for different parts of the prediction data[6]. Thus, we can create a stacked classifier that has higher accuracy than each of the three classifiers on its own. The stacking is carried out with training a Logistic Regression model. The Logistic Regression model will assign weights to the predictions of each of the three classification models, and take the weighted average to make its own predictions. This model would have higher accuracy as it combines the strengths of each individual classifier, and reduces overfitting.

D. Long-Short Term Memory Model

The fourth part of the research involves using a Long-Short Term Memory (LSTM) model to predict the price movement of the FX pair. A LSTM model is actually a form of a Recurrent Neural Network (RNN) model. An RNN model works by training a set of randomly chosen weights. The weights are applied to the input values and then adjusted based on whether the output value is close to the actual output value. However, an RNN model is not suitable to be applied in this context, as the training data would span a long period of time. Hence, the effects of certain significant factors in the prediction model would be diminished over time, as the model does not keep any long term memory[4]. A LSTM model is better suited as it can "forget" the memory that does not have any significant impact on the predictions [1].

E. Backtesting

The fifth part of the research involves backtesting our model on the test data. The model performance is then evaluated in terms of Sharpe ratio, and maximum drawdown.

For the classifiers, the trading strategy that we use is that if the model predicts that the next day's closing price will be higher than today's closing price, we all-in on the next day's open price, and exit on the next day's close. The same goes for the opposite direction.

For the LSTM model, the trading strategy that we use is as follows: If the predicted closing price of the next day is higher than today's closing price, we all-in on the next day's open price, and exit on the next day's close. The same goes for the opposite direction.

IV. Results

In this section, we showcase the results of our LSTM model and Classifier model on their accuracy in predicting the Forex Movement and how our trading strategy performs using these models.

FX pair/Accuracy	Random Forest	XGBoost	SVM (kernel=rbf)	Meta classifier
EURUSD	0.51	0.52	0.49	0.52
GBPUSD	0.78	0.83	0.47	0.83
USDCHF	0.76	0.81	0.47	0.80
USDCNY	0.74	0.80	0.76	0.80
USDJPY	0.78	0.82	0.52	0.85
USDSGD	0.50	0.53	0.51	0.51

TABLE I: Accuracy of each classifier for FX pairs

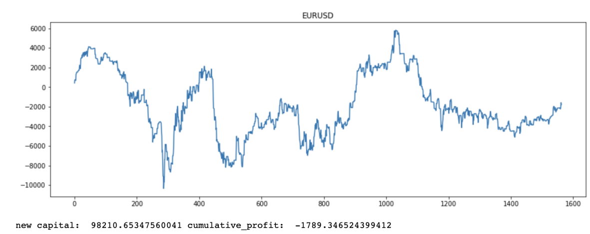

Fig. 1: Profit curve for the classifiers for EURUSD

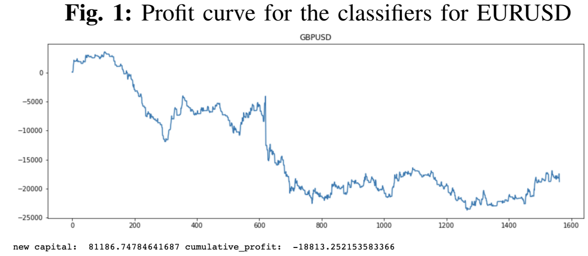

Fig. 2: Profit curve for the classifiers for GBPUSD

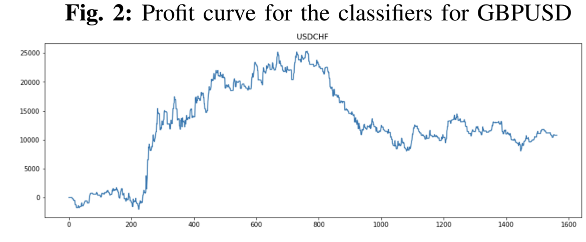

Fig. 3: Profit curve for the classifiers for USDCHF

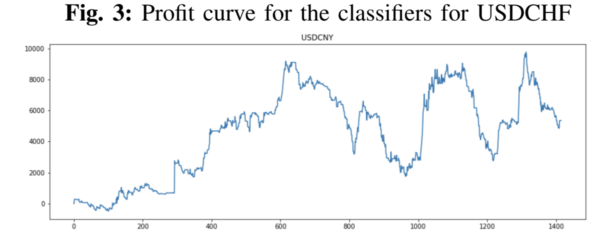

Fig. 4: Profit curve for the classifiers for USDCNY

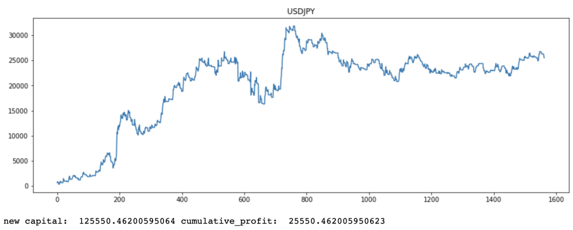

Fig. 5: Profit curve for the classifiers for USDJPY

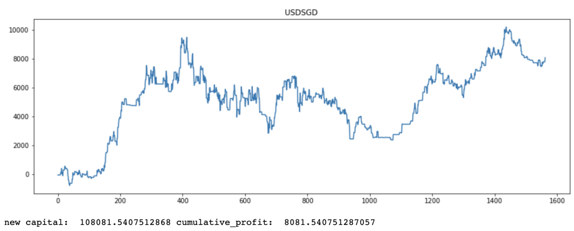

Fig. 6: Profit curve for the classifiers for USDSGD

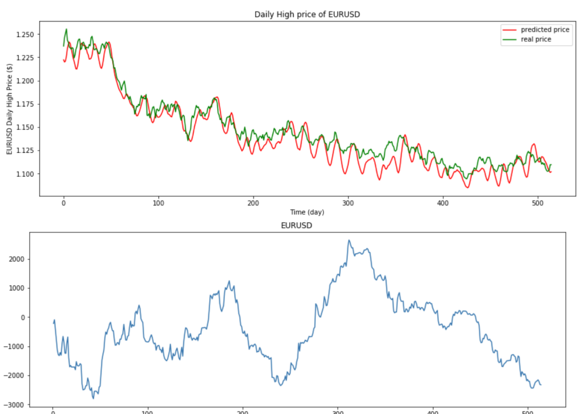

Fig. 7: Prediction curve and profit curve for LSTM for EURUSD

Fig. 8: Prediction curve and profit curve for LSTM for GBPUSD

Fig. 9: Prediction curve and profit curve for LSTM for USDCHF

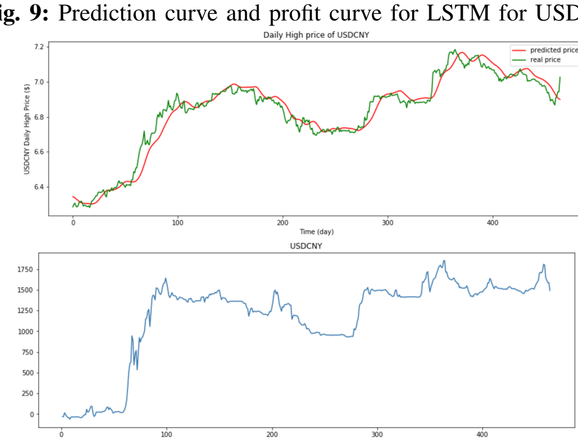

Fig. 10: Prediction curve and profit curve for LSTM for USDCNY

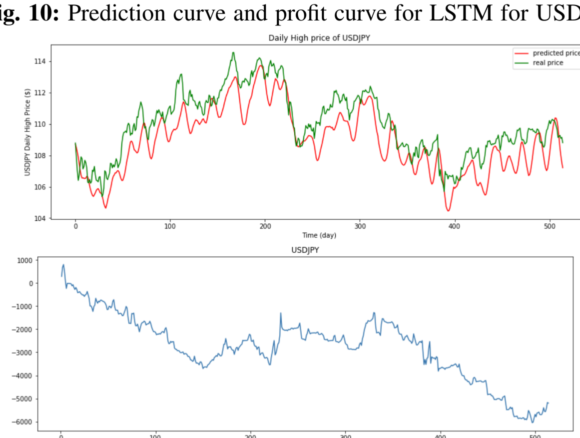

Fig. 11: Prediction curve and profit curve for LSTM for USDJPY

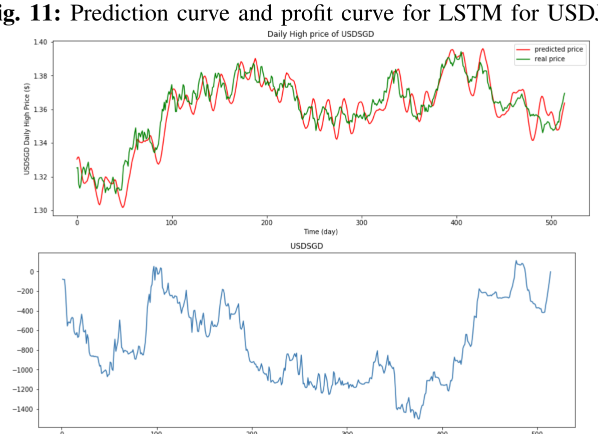

Fig. 12: Prediction curve and profit curve for LSTM for USDSGD

V. Discussion

A. Accuracy and Trading Strategy: Does high accuracy translate to profit?

Average accuracy of the Classifiers model is above 80%. However, with the current simple trading strategy which is to take one action per day - buy or sell, the model performs badly with some forex pairs. Fig. 2 shows an example of how the model can perform negatively (GBPUSD FX pair).

We test the model using the latest 30% of the total data, which is roughly 1600 days, starting with 100000 capital. At the end, the net loss is 18813, roughly 20%. Thus, it is not trivial to generate profit even if we can predict the direction of the market with a relatively high accuracy.

B. Assumptions and anomalies: Is trading strategy bulletproof?

Major economic downturns is an inevitable weakness of the model and its trading strategy. Our models follow that, in the long run, price will always increase. The fact that an economic crisis is almost unpredictable - period wise and magnitude wise - the models could not account for future economic crises and would fail in both accuracy and returns in such situations.

C. Assumptions and anomalies: why is the predicted price produced by LSTM always higher than actual price?

Both LSTM and classifiers produced fairly accurate predictions. While classifiers generated indications on whether price is going up or down, LSTM provided prediction on the exact price. However, the predicted price provided by LSTM is usually slightly higher than the actual price. We suspect that the discrepancy is due to irrational market behaviour during bubbles which misled the model to memorise a forever rising trend of the stock.

D. Additional Notes

In our methodology, we mentioned that we would stack 3 ML classifiers together, namely Random Forest, SVM, XGBoost together to form a meta classifier. However, we omitted the SVM classifier when we were preparing the meta classifier. This is because the SVM classifier has significantly lower accuracy than the other two classifiers, so it will greatly affect the meta classifier accuracy. The reason behind the low accuracy of the SVM classifier is that it attempts to use a hyperplane to separate the data points. In this case, the data points are not separable, so this leads to a low accuracy in the SVM model.

E. Improvements on the ML models

In future studies, the ML classifier models can be improved further by hyperparameter optimisation. One possible method is using GridSearchCV that is provided in scikitlearn.

VI. Conclusion

In conclusion, the study on the two models showcased some interesting results that could pave the way for better machine learning models in this high-frequency and time-sensitive data. We hope that one day that the models could be profitable off the highly-volatile Forex Market.

It is worth noting that being able to predict the direction of the market with high accuracy does not necessarily translate to significant profits. In future iterations, a better trading strategy, perhaps with better risk management, is needed to improve the current returns of the models. With the current trading strategy, the models would be making a loss. One of the suggested trading strategies is to use NR4. The philosophy behind the NR4 pattern is that a volatility contraction is often followed by a volatility expansion. Narrow range days mark price contractions that often precede price expansions. Hence NR4 is an objective criterion for identifying days of decreased range and volatility. Then when we identify the NR4 patterns, we will trade the breakout to get profits. The prediction that we have obtained can significantly improve the performance of NR4 as it can provide a good indication of future NR4 days.

We also expect better accuracy on LSTM predictions by using data with less noise. Weekly data is likely to be a better option as compared to daily data because weekly data generally better captures the trend in the market. Since LSTM works by letting the neuron network study the trend per se, it is convincing that data with a better capability of reflecting trend is going to be more performant.

References

[1] F. Rundo, “Deep LSTM with reinforcement learning layer for financial trend prediction in FX high frequency trading systems,” *Applied Sciences*, vol. 9, no. 20, p. 4460, 2019.

[2] T. Fletcher, F. Redpath, and J. D’Alessandro, “Machine learning in FX carry basket prediction,” in *Proceedings of the World Congress on Engineering*, vol. 2, pp. 1371–1375, Citeseer, 2009.

[3] A. A. Baasher and M. W. Fakhr, “Forex trend classification using machine learning techniques,” in *Proceedings of the 11th WSEAS international conference on Applied computer science*, pp. 41–47, World Scientific and Engineering Academy and Society (WSEAS), 2011.

[4] R. Zanc, T. Cioara, and I. Anghel, “Forecasting financial markets using deep learning,” in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 459–466, IEEE, 2019.

[5] G. Abreu, R. Neves, and N. Horta, “Currency exchange prediction using machine learning, genetic algorithms and technical analysis,” *arXiv preprint arXiv:1805.11232*, 2018.

[6] E. A. Gerlein, M. McGinnity, A. Belatreche, and S. Coleman, “Evaluating machine learning classification for financial trading: An empirical approach,” *Expert Systems with Applications*, vol. 54, pp. 193–207, 2016.