FORECASTING DIRECTIONAL MOVEMENT OF FOREX DATA USING LSTM WITH TECHNICAL AND MACROECONOMIC INDICATORS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENİZ CAN YILDIRIM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MAY 2019

Approval of the thesis:

FORECASTING DIRECTIONAL MOVEMENT OF FOREX DATA USING LSTM WITH TECHNICAL AND MACROECONOMIC INDICATORS

submitted by **DENİZ CAN YILDIRIM** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Halit Oğuztütün
Head of Department, **Computer Engineering** ______________

Prof. Dr. İsmail Hakkı Toroslu
Supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Prof. Dr. Tolga Can
Computer Engineering, METU ______________

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU ______________

Assoc. Prof. Dr. Osman Abul
Computer Engineering, TOBB ETU ______________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Deniz Can Yıldırım

Signature :

ABSTRACT

FORECASTING DIRECTIONAL MOVEMENT OF FOREX DATA USING LSTM WITH TECHNICAL AND MACROECONOMIC INDICATORS

Yıldırım, Deniz Can
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. İsmail Hakkı Toroslu

May 2019, 69 pages

Foreign Exchange is known as Forex or FX is a financial market where currencies are bought and sold simultaneously. Forex is the largest financial market with more than $5 trillion volume. It is a decentralized market that is operational 24 hours in a day other than weekends which makes different from other markets. Fundamental and Technical Analysis are the two techniques that are commonly used in predicting the future prices in Forex. Fundamental Analysis concentrates on the economical, social and political factors that can cause to price moving higher, lower or staying the same. Technical analysis, on the other hand, is based on only the price to predict the future price movements. It studies the effect of the price movement by using technical indicators.

In this thesis, a model that uses LSTM with technical and macroeconomic indicators is proposed to forecast directional movement of Forex data. It is based on the two LSTM models that learn the effects of both indicators individually. The predictions of two LSTM models are combined according to the predefined set of rules in order to determine the final decision. The experiments are conducted on EUR/USD currency

pair to forecast 1-day, 3-days and 5-days ahead and promising results are succeeded.

Keywords: time series, forex, directional movement forecasting, technical indicator, macroeconomic indicator, lstm

ÖZ

FOREX VERİSİ HAREKETLENME YÖNÜNÜN TEKNİK VE MAKROEKONOMİK GÖSTERGELER KULLANILARAK LSTM İLE TAHMİNLENMESİ

Yıldırım, Deniz Can
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. İsmail Hakkı Toroslu

Mayıs 2019 , 69 sayfa

Döviz alım satım piyasası, yaygın olarak bilinen isimleriyle Forex veya FX, para birimlerinin aynı anda alınıp satıldığı bir finansal piyasadır. Dünyanın en büyük piyasası olup 5 trilyon doların üzerinde hacime sahiptir. Merkezi olmayan bir pazar olması, hafta sonları dışında günde 24 saat saat açık olması Forex'i diğer piyasalardan farklı kılmaktadır. Temel ve Teknik analiz, Forex'te gelecekteki fiyatları tahmin etmekte yaygın olarak kullanılan iki tekniktir. Temel analiz, fiyatın yükselmesine, düşmesine veya aynı kalmasına neden olabilecek ekonomik, sosyal ve politik faktörlere odaklanır. Öte yandan teknik analiz, gelecekteki fiyat hareketlerini öngörmek için yalnızca fiyatları kullanmaktadır. Teknik göstergeler kullanarak, fiyat hareketlerinin etkisini incelemektedir.

Bu tezde, Forex verilerinin yönlü hareketini tahmin etmek için teknik ve makroekonomik göstergelerle LSTM'yi kullanan bir model önerilmiştir. Her iki göstergenin etkilerini ayrı ayrı öğrenen iki LSTM modeline dayanmaktadr. İki LSTM modelinin öngörüleri, nihai kararı belirlemek için öntanımlı kurallar dizisine göre birleştirilir.

Deneyler, 1 gün, 3 gün ve 5 gün sonrasını öngörmek için EUR / USD döviz çifti üzerinde gerçekleştirilmiş ve umut verici sonuçlar elde edilmiştir.

Anahtar Kelimeler: zaman serisi, forex, hareketlenme yönü tahmini, teknik gösterge, makroekonomik gösterge, lstm

to all my beloved ones

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. İsmail Hakkı Toroslu for giving me the opportunity to work with him during my thesis. Thanks to his guidance, friendly and supportive attitude, I find a way to complete this study. It was a great pleasure to discuss the progress and ideas with him.

I am deeply grateful to my family, without them I couldn't accomplish this far. My dearest parents Gülay and Ensar show me their endless love and support that give me confidence in my entire life. I also wish to send my special thanks for my sister Duygu for being a source of motivation at all times by teaching me the value of working hard.

Last but not least, I would like to thank love of my life Pelin. Her presence is always with me and I feel her sincere love that makes me stay strong in this period.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
MA	Moving Average
MACD	Moving Average Convergence Divergence
ROC	Rate of Change
RSI	Relative Strength Index
RS	Relative Strength
BB	Bollinger Bands
CCI	Commodity Channel Index
EUR	Euro
USD	American Dollar

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

Foreign Exchange is known as Forex or FX is a financial market where currencies are bought and sold simultaneously. Forex is the largest financial market with more than $5 trillion volume. It is a decentralized market that is operational 24 hours in a day other than weekends which makes different from other markets.

The characteristics of Forex show differences compared to other markets. Such differences bring advantages to Forex users for profitable trading. Some of these features are no commissions, no middlemen, no fixed lot size, low transaction cost, high liquidity, almost instantaneous transactions, low margin and high leverage, a 24-hour market, no insider trading, limited regulation and online trading and etc.

There are two types of analysis made on financial time series to predict future values in Forex: Fundamental Analysis and Technical Analysis. The former uses macroeconomic factors and the latter uses historical data to forecast the future price movement.

We see that there are two kinds of issues in financial time series forecasting problem in our literature survey. While the first one is related with prediction on the exact value of data in the next time slot, the other one concentrates on the direction prediction of the price. In Forex environment, forecasting directional movement has a significant importance to make profits from transactions. Identifying directional movement is the problem that we address in this study.

1.2 Proposed Method & Contributions

Forecasting the directional movement problem in financial time series could be approached with two different perspective:

- Firstly, predict the future value of price than compare it with past values and decide the direction ($regression$),
- Directly determine the directional movement of the price ($classification$).

Our initial study uses the regression method to determine the directional movement. In this approach, we try to predict the exact value of the prices. After comparison with the past values made by using these predictions, we reach the decision whether there will be increase or decrease on the future price. With the help of the method introduced in [5], we examine the effect of the regression on the directional movement prediction in financial time series.

Since our primarily concern is on the directional movement, increase and decrease could be thought as class labels and we can focus on developing an algorithm that can be used to forecast these classes. By this way, we succeed to forecast the increase/decrease value directly.

In this thesis, we propose the combination of multiple LSTM models in order to forecast the directional movement of EUR/USD currency pair in Forex environment with regard to explanation above. The prediction periods contain 1-day, 3-days and 5-days ahead.

Our hybrid model is composed of **macroeconomic LSTM model** and **technical LSTM model** which is named by the types of inputs they take. We aim to investigate the effects of these variables on the directional movement separately. Learning different aspects of data by two different models enables us to improve our classification accuracy in a significant manner. **Macroeconomic LSTM model** explores the several financial factors consisting of interest rates, FED funds rate, inflation rates, S&P 500 and DAX market indexes. Each of these variables has an important role on the trend of EUR/USD currency pair. This could be interpreted as fundamental analysis on price data. Another model is called as **technical LSTM model** which is responsible for taking advantage of technical analysis. Technical analysis is based on technical indicators which are mathematical formulas used for predicting future price action. MA, MACD, ROC, Momentum, RSI, BB and CCI indicators are used as a feature set by our model.

Our approach introduces another class label called **no_action** together with $increase$ and $decrease$. This label means the changes remaining between certain thresholds are negligible and requires no action at all. Only when a difference between data points is greater/less than the threshold, corresponding data point will be labeled as increase/decrease. Otherwise, we treat this data point as unaltered.

Introducing new class labels leads us to define new performance metric named as **profit_accuracy**. This metric is related to the number of increases and decreases of the predicted labels. We could interpret this metric as that gives us the ratio of number of profitable transactions over the total transactions. We simply define profitable transaction as right prediction of decrease and increase classes. Due to the fact that labeled and predicted class no_action have no contribution on the profit/loss of the transaction, we need this brand new concept.

We present another procedure on the usage of predictions of macroeconomic LSTM and technical LSTM models. We determine the final output of the system by using these predictions according to the predefined set of rules (details are explained in Chapter 5).

Our proposed model brings different approaches together to forecast financial time series. Our contributions yield with the way of combining these approaches which are listed below:

- In this study, we explore the fundamental and technical analysis individually. Since both analysis have own characteristics, we manage to learn different aspects of the EUR/USD pair by developing different LSTM models,

- We develop an algorithm that finds the optimal threshold value for labeling data points. At the end, we get balanced class distribution for our dataset,

- We define the new performance metric $profit\_accuracy$ that gives us profitable transaction ratio,
- Final contribution is the establishment of the set of rules for combining individual predictions of macroeconomic LSTM and technical LSTM models.

1.3 The Outline of the Thesis

In Chapter 2, related studies on financial time series prediction are thoroughly examined.

Chapter 3 explains the background information on Forex, LSTM and technical indicators. Additionally, the dataset used in experiments is presented.

Chapter 4 introduces the proposed algorithm to handle directional movement prediction problem.

In Chapter 5, pre-processing and post-processing techniques are explained in detail. Moreover, it covers the results of the experiments and classification performances of proposed model.

Chapter 6 discusses the result of the experiments and gives insight about the future directions.

CHAPTER 2

RELATED WORK

2.1 Forecasting in Financial Time Series

L. Zhang et al. [6] proposed state frequency memory recurrent network, modification of LSTM, in order to forecast stock price. By decomposing hidden states of memory cells into multiple frequency components, they manage to learn trading patterns of these frequencies. They use state-frequency components to make predictions on future value of prices through a nonlinear regression. They used stock prices from several sectors and made experiments to forecast one, three and five days later. They compared results with LSTM and ARIMA in terms of mean square error. They obtained errors 5.57, 17.00 and 28.90 respectively for different steps which were better than the other models.

In [7], J. Patel et al. developed a two stage fusion structure to predict future values of stock market index with 1-10, 15 and 30 days ahead using ten technical indicators. In the first stage, SVR was applied to these inputs and results fed into ANN (SVR-ANN), SVR(SVR-SVR) and RF(SVR-ANN) models in the second stage. They made comparison between fusion model and stand alone ANN, SVR and RF separately. They reported that the fusion model made significant improvement over stand alone models.

F. Tay and L. Cao [8] examined SVM on forecasting future values of five real future contracts data and compared results with BPNN. Several performance metrics are used in the experiments, namely Normalized Mean Squared Error (NMSE), Mean Absolute Error (MAE), Directional Symmetry (DS) and Weighted Directional Symmetry (WDS). They concluded that SVM showed better performances than BPNN.

E. Guresen et al. [9] explores several ANN models on predicting stock market index. These models were Multi-layer Perceptron (MLP), Dynamic Artificial Neural Network (DAN2) and the hybrid neural networks with generalized Autoregressive Conditional Heteroscedasticity (GARCH). They used mean square error (MSE) and Mean Absolute Deviate (MAD). Results showed that MLP performance was slightly better than DAN2 and GARCH-MLP and GARCH-DAN2 had the worst results.

In [10], B. Weng et al. developed a financial expert system by using ensemble methods (neural network regressing ensemble (NNRE), support vector regression ensemble (SVRE), boosted regression tree (BRT) and random forest regression (RFR)) to predict 1 day ahead stock price. Not only market prices and technical indicators but also financial news, Google Trends and number unique visitors of Wikipedia pages were given as inputs. They also investigated the effect of PCA on the performances. They reported that the ensembles with PCA performed better than ensembles without PCA. They also mentioned that BRT and RFR were the best and SVRE was the worst in terms of mean absolute percentage error.

2.2 Directional Forecasting in Financial Time Series

In [11], W. Huang et al. examined forecasting weekly stock market movement direction with SVM. They compared SVM with Linear Discriminant Analysis, Quadratic Discriminant Analysis and Elman Back-propagation Neural Networks. They also proposed a combining model i.e, SVM with other classifiers. They used not only NIKKEI 225 index but also macroeconomic variables as feature for their model. Their direction calculation was based on the first order difference natural logarithmic transformation and directions were either increase or decrease. SVM outperformed the other models with the accuracy of 73% and combining model was the best with the accuracy of 75%.

M. Qiu and Y. Song [5] developed an genetic algorithm (GA) based optimized artificial neural network to predict the direction of the next day's price of the stock market index. GA was used for optimizing the initial weights and bias of the model. Two types of input sets that were generated by using several technical indicators on daily price of the Nikkei 225 index and fed into the model. They got accuracies 60.87% in the first set and 81.27% in the second set.

Y. Kara et al. [12] used artificial neural network and support vector machine for comparing their performances on the direction of stock price index movement prediction. Ten technical indicators were given as input to the model. They concluded that ANN with accuracy of 75.74% performed significantly better than SVM with accuracy of 71.52%.

In [13], S. Galeshchuk and S. Mukherjee investigated the performance of Convolutional Neural Network (CNN) on prediction of the direction of change in Forex. They used Euro and USD (EUR/USD), Pound and USD (GBP/USD) and USD and Japanese Yen (USD/JPY) daily closing rate and compare the results of CNN with their baseline models and SVM. While they had accuracy around 65% in baseline models and SVM, they reported around 75% in the proposed CNN model.

In [14] J. Patel et al. compared performances of four classifiers (ANN, SVM, Random Forest and Naive Bayes) on predicting the direction of stock price index with two approaches. In the first approach, they used ten technical indicator's values as input with different parameter settings of classifiers. Their prediction accuracies were changing between the range 0.7331 and 0.8359. In the other approach, they represented same ten technical indicators' results as directions (up and down) which were given as inputs to the classifiers. By this way, they enhanced accuracies about fifteen percent for all classifiers. Although their experiments were about short term prediction, the direction period was not explicitly explained.

M. Ballings et al. [15] evaluated ensemble methods (Random Forest, AdaBoost and Kernel Factory) against to neural networks, logistic regression, SVM and K-nearest neighbor for predicting one year ahead. They used different stock market domains in their experiments. According to the median area under curve (AUC) scores, Random Forest presented the best performance followed by SVM, Random Forest and Kernel Factory.

In his thesis [16], Q. Gao studied the stock market forecasting in six different domains by using LSTM. He tried to predict the next three hours with using hourly historical stock data. The model was trained to classify three classes namely increasing between 0%-1%, increasing above 1% and not increasing (less than 0%). His accuracy results were changing between 49.75% and 59.5%. He also built stock trading simulator for testing his model on real world stock trading activity. With this simulator, he managed to make profit on all six stock domains with an average 6.89%.

D. Nelson et al. [17] examined the LSTM on predicting future 15 minutes trend of stock prices with using technical indicators. They used 175 technical indicators (external technical analysis library) and open, close, minimum, maximum and volume of price as inputs to the model. They compared their model with baseline consisting of multi layer perceptron, random forest and a pseudo-random model. The accuracies of LSTM among different stocks are between 53% and 55.9%. They concluded that LSTM performed significantly better than baseline models according to Kruskal-Wallis test.

X. Zhong and D. Enke [18] investigated three dimensionality reduction techniques applied to the ANN for forecasting daily direction of the S&P 500 Index ETF (SPY). Principal component analysis (PCA), fuzzy robust principal component analysis (FR-PCA) and kernel-based principal component analysis (KPCA) are used for reducing the number of features. Their experiments indicated that ANN with PCA performed slightly better than other two techniques.

In [19], H. Hu et al. introduced improved sine cosine algorithm (ISCA) for optimizing weights and biases of BPNN in order to predict directions of open stock prices of S&P 500 and Dow Jones Industrial Average Indices. By using Google Trends data in addition to the open, high, low, close price and trading volume in their experiments, they obtained 86.81% hit ratio for S&P 500 index and 88.98% hit ratio for Dow Jones Industrial Average Index.

In [20], K.Kim investigated SVM on predicting the direction of stock price index with different parameter settings. He also compared the result of SVM with BPNN and case-based reasoning models. He used multiple technical indicators as inputs to the models. He concluded that SVM outperformed the other models with accuracy of 57.8313% while the other models were having 54.7332% and 51.9793% respectively.

CHAPTER 3

BACKGROUND ON FOREX, LSTM AND TECHNICAL INDICATORS & DATASET REPRESENTATION

3.1 Introduction to Forex

Foreign Exchange is known as Forex or FX is a financial market where currencies are bought and sold simultaneously. Forex is the largest financial market with more than $5 trillion volume. It is a decentralized market that is operational 24 hours in a day other than weekends which makes different from other markets [21, 22].

The characteristics of Forex show differences compared to other markets. Such differences bring advantages to Forex users for profitable trading. Some of these features are no commissions, no middlemen, no fixed lot size, low transaction cost, high liquidity, almost instantaneous transactions, low margin and high leverage, a 24-hour market, no insider trading, limited regulation and online trading and etc [21].

The profit/loss calculation can be made by taking difference of final value and initial value of currency pair. If currency pair increases and trader goes long or currency pair decreases and trader goes short, trader will make profit from that transaction. Otherwise, trader will lose. For example, if EUR/USD at 1.1500 and trader goes long $10000. When the position was closed in 1.1550, the trader would gain `10000 * (1.1550 - 1.1500) = $50`. When the position was closed in 1.1450, the trader would lose `10000 * (1.1500 - 1.1450) = $50`.

3.1.1 The Forex Terminology

In this section, definitions of commonly used concepts and terms in Forex environment are explained [23, 24, 21]:

- `Base Currency:` is also called transaction currency is the first currency in the currency pair. To illustrate, EUR/USD pair, EUR is the base currency.

- `Quote Currency:` is the second currency in the currency pair. To illustrate, EUR/USD pair, USD is the quote currency.

- `Currency Strength:` is the the future value of currency and can be defined against other currencies.

- `Being Long (Going Long):` means buying the base currency or selling the quote currency in the currency pair.

- `Being Short (Going Short):` means selling the base currency or buying the quote currency in the currency pair.

- `Pip:` is the abbreviation of "percentage of point" defined as smallest amount of change occurs in currency. In general, pip corresponds to four decimal points (0.0001) of that currency.

- `Pipette:` is the fractional pip corresponds to five decimal points (0.00001). In other words, 1 pip equal 10 pipettes.

- `Leverage:` is the use of borrowed money in making transactions. A leverage 1:100 indicates that if one opens a position of volume 1, actual transaction volume will be 100. At the end of using leverage, one can either gain or loss 100 times of that volume.

- `Margin:` is the borrowed money for the trader that is supplied by broker to make investment using leverage. In this way, one can multiply his/her gaining or losses. `Margin call` is another term that a rule formed by broker to protect company's money by narrowing the losses of the trader in a leveraged position.

- `Bid Price:` is the price that trader can sell the base currency.
- `Ask Price:` is the price that trader can buy the base currency.
- `Spread:` is the difference between ask and bid prices. Lower spread means that trader can make profits in small price changes. Spread value is dependent to market volatility and liquidity.
- `Stop Loss:` is an order for trader to sell currency when it reaches a specified price. This term prevent trader from large amount of loss.
- `Take Profit:` is an order for trader to close the open position for a gain when the price reaches a specific value. This term guarantees the profit of trader without concern about the changing in market price.
- `Market Order:` is an order which is committed instantly with the current price.
- `Swap:` is the simultaneous buy and sell actions on the currency with same amount at a forward exchange rate. It saves dealers from fluctuations of interest rates of base and quote currencies. In continuous conditions of base currency having higher interest rate and quote currency having lower interest rate, positive swap will occur. Negative swap will occur in the case of opposite conditions.

3.1.2 Fundamental and Technical Analysis

Fundamental and Technical Analysis are the two techniques that are commonly used in predicting the future prices in Forex. While the first one is based on the economic factors, the other is related with price actions [21].

`Fundamental Analysis:` concentrates on the economical, social and political factors that can cause to price moving higher, lower or staying the same [25, 21]. These factors are also called as macroeconomic factors and economic data reports, interest rate levels, monetary policy and international trade/investment flows are some examples of them [22].

`Technical Analysis`: is based on only the price to predict the future price movements [26]. It studies the effect of the price movement. Technical analysis mainly uses open, high, low, close and volume data to either predict market direction or generate sell and buy signals [21]. It is based on these there premises: "Market action discounts everything", "Price move in trends" and "History repeats itself" [25]. Chart analysis and price analysis using technical indicators are the two main approaches of technical analysis. While the former is used to detect patterns in price charts, the latter is used to predict future price action [22].

3.2 Long short Term Memory (LSTM)

Long short Term Memory (LSTM) was proposed by S. Hochreiter and J. Schmidhuber in 1997. LSTM is a recurrent neural network architecture that was designed to overcome vanishing gradient problem exists in the conventional RNNs [27]. The errors between layers tend to vanish or blow up which causes to oscillating weights or incredibly long convergence time. The initial LSTM structure solves this problem by introducing constant error carrousel (CEC). By this way, the architecture ensures the constant error flow between the self-connected units [1].

The figure 3.1 shows memory cell of the initial LSTM structure. This memory cell consists of input gate and output gate. While the input gate decides which information should be kept or updated in memory cell, output gate allows which information should be output.

This standard LSTM was extended with introducing the new feature called forget gate [2]. The forget gate is responsible of resetting the memory state that contains outdated information. The figure 3.2 represents the structure of LSTM with forget gate.

Furthermore, peephole connections and the full back-propagation through time (BPTT) training are final features that were included in the LSTM architecture [28, 3]. With these modifications, the architecture is named as Vanilla LSTM [3] shown in the figure 3.3.

LSTM offers an effective and scalable model for learning problems that includes

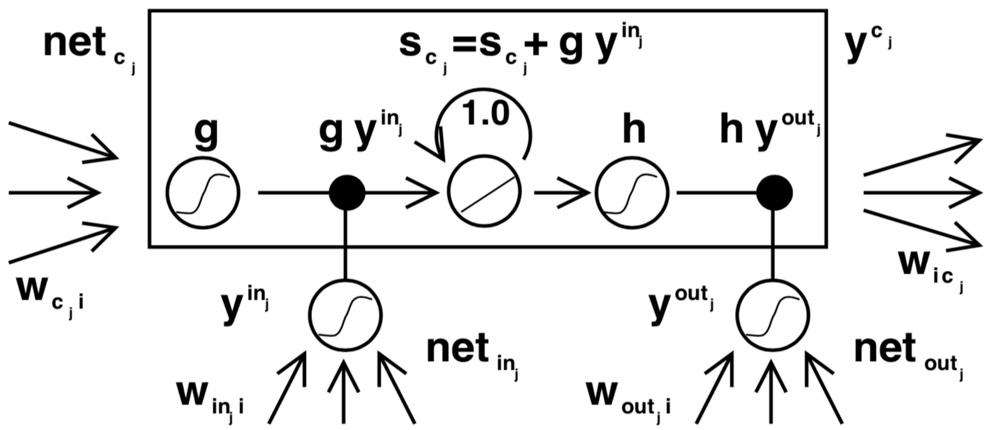

Figure 3.1: Initial LSTM Structure [1] (Architecture of the memory cell).

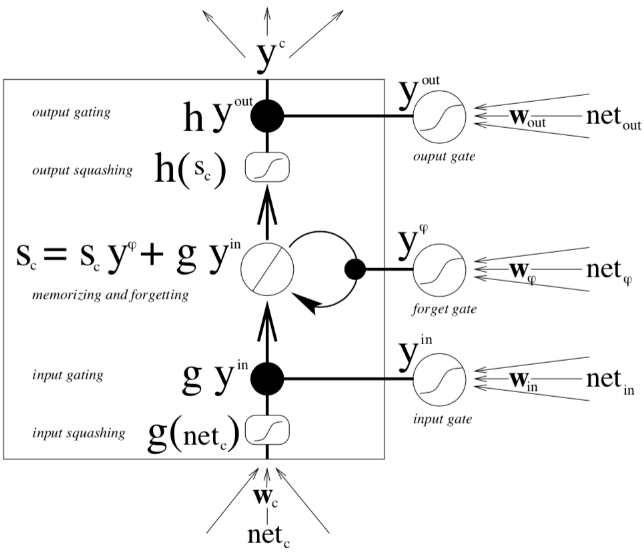

Figure 3.2: Memory cell with forget gate of the extended LSTM [2].

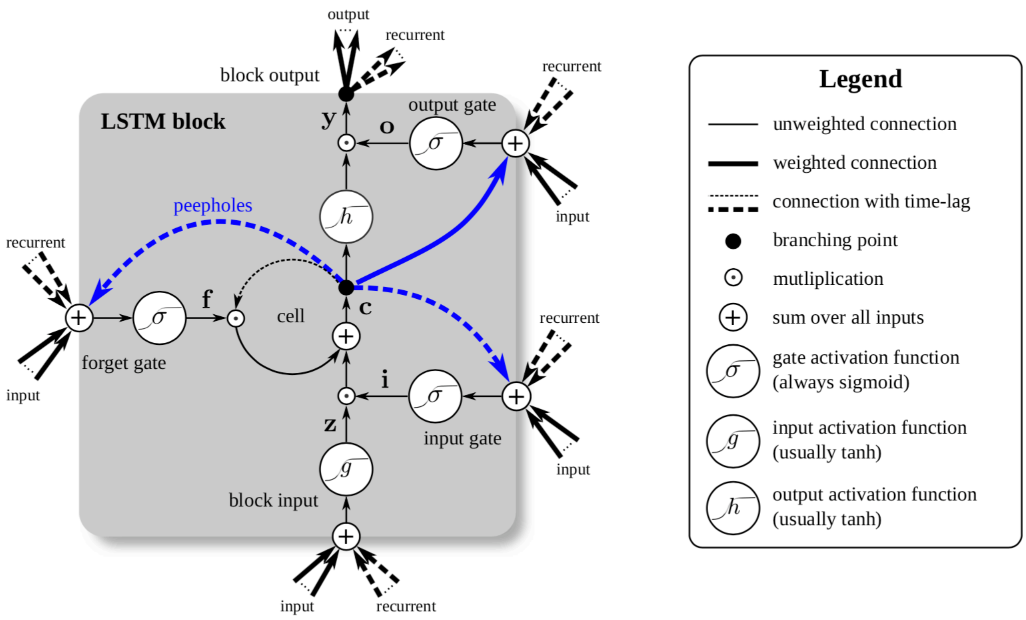

Figure 3.3: Vanilla LSTM [3].

sequential data [3]. It has been used in many different fields by researchers including handwriting recognition (A. Graves et al. [29] and V. Pham et al. [30]), and generaton (A. Graves [31]), language modeling (W. Zaremba et al. [32]) and translation (M. Luong et al. [33]), acoustic modeling of speech (H. Sak et al. [34]), speech synthesis (Y. Fan et al. [35]), protein secondary structure prediction (S. Kaae et al. [36]), analysis of audio (E. Marchi et al. [37]) and video data (J. Donahue et al. [38]) [3].

3.3 Technical Indicators

Technical Indicator is a price time series data that results from mathematical formula(s) applied to another time series data [39]. These formulas can use price data's close, open, high and low values or volumes information. Technical Indicators can be applied to the any domain that carrying out a trade in an open market is possible namely stocks, futures, commodities, Forex etc. They are solid assistants for identifying future price trend and measuring volatility [22]. By analyzing historical data, they can help to forecast future prices.

According to their functionality, technical indicators can be grouped in three categories: lagging, leading and volatility indicator. Lagging indicators also referred as trend indicators follow the past price action. MA and MACD are the examples of lagging indicators. Leading indicators also known as momentum based indicators try to predict future price trend direction and shows rate of changes in the price. ROC and RSI are examples of the leading indicators. Volatility based indicators measure the volatility level happened in the price. BB and Average True Range (ATR) are the examples of volatility based indicators.

In the following sections, the detailed information about technical indicators used in the thesis is explained.

3.3.1 Moving Average (MA)

Moving Average (MA) is a trend following or lagging indicator that smooths the prices by averaging them in a specified period. In this way, MA can help to filter out the noise. Not only MA manages to identify trend direction but also determines potential support and resistance levels [39]. The formulation of MA is given as below:

$$SMA = \frac{SUM(Close, N)}{N} \tag{31}$$

- N is the period,
- $Close$ is the close price,
- $SUM(Close, N)$ is the sum of close prices in periods of N,
- SMA is simple moving average.

3.3.2 Moving Average Convergence Divergence (MACD)

Moving Average Convergence Divergence (MACD) is a momentum oscillator that was developed by Gerald Appel in the late 1970s. It is a trend following indicator that uses the short and long term exponential moving averages of prices [40]. MACD uses

Figure 3.4: Moving Average of EUR/USD [4].

short term moving average to identify price changes quickly and long term moving average to emphasize trend [22]. The formulation of the MACD indicator is:

$$MACDLine = EMA(Close, N1) - EMA(Close, N2) \tag{32}$$

$$EMA = Close - EMA(previous) * C + EMA(previous) \tag{33}$$

where

$$C = \frac{2}{N+1} \tag{34}$$

- C is exponential smoothing constant,
- N is the total number of periods in a simple moving average to be roughly approximated by the EMA,
- $Close$ is the close price,
- EMA and $EMA(previous)$ are the current and the previous period's EMA values,
- $N1$ and $N2$ are the short and long term moving average periods ($N1 = 12$ and $N2 = 26$ are in default),
- $MACDLine$ is the MACD line.

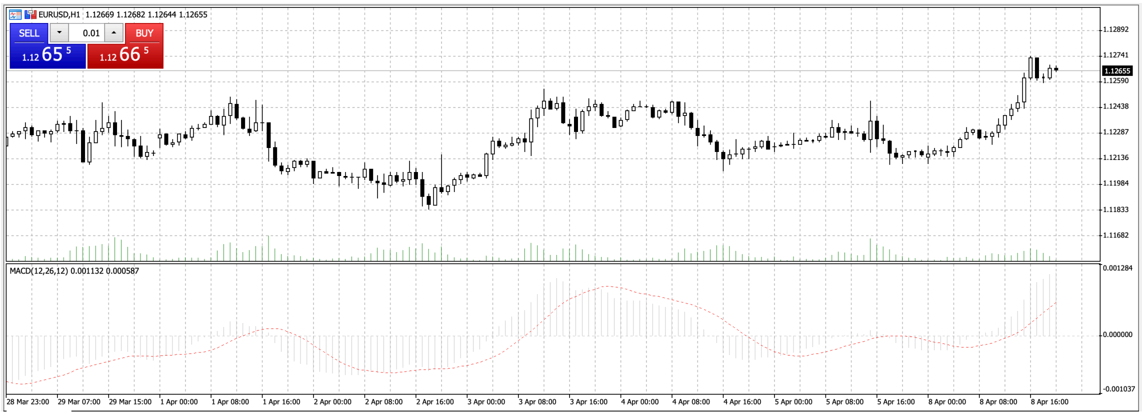

Figure 3.5: MACD Line of EUR/USD [4].

3.3.3 Rate Of Change (ROC)

Rate of Change (ROC) is the momentum oscillator that defines velocity of price. This indicator measures percentage of direction by calculating ratio between current close price and close prices of specified period ago [22]. The formulation of ROC is defined as:

$$ROC = \frac{Close - Close(previous, N)}{Close(previous, N)} * 100 \tag{35}$$

- N is the period,
- $Close$ and $Close(previous, N)$ are close price and close price of N periods ago,
- ROC is the Rate of Change value.

3.3.4 Momentum

Momentum measures the amount of change happened in the price in a specified period [41]. It is a leading indicator that shows rises and falls in the price or remains stable when the the current trend prevails. Momentum is defined by constantly calculating differences of prices for a set time interval [25]. The formulation of Momentum

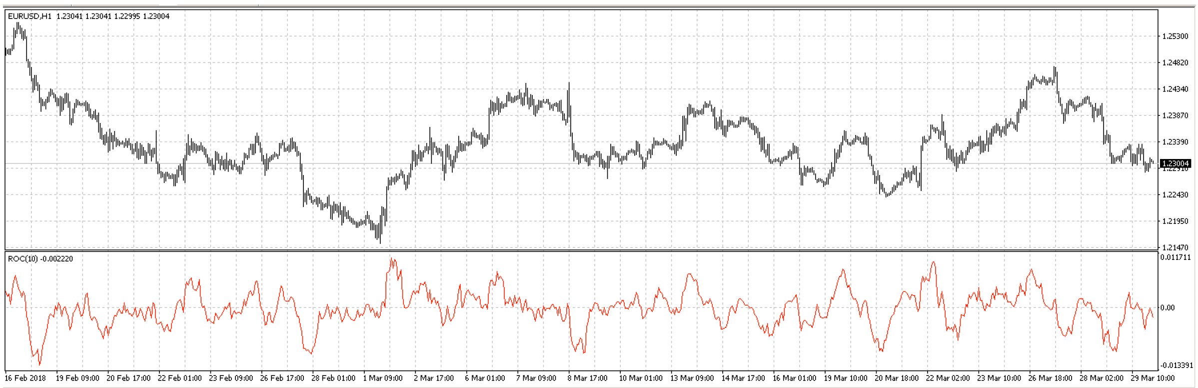

Figure 3.6: ROC of EUR/USD [4].

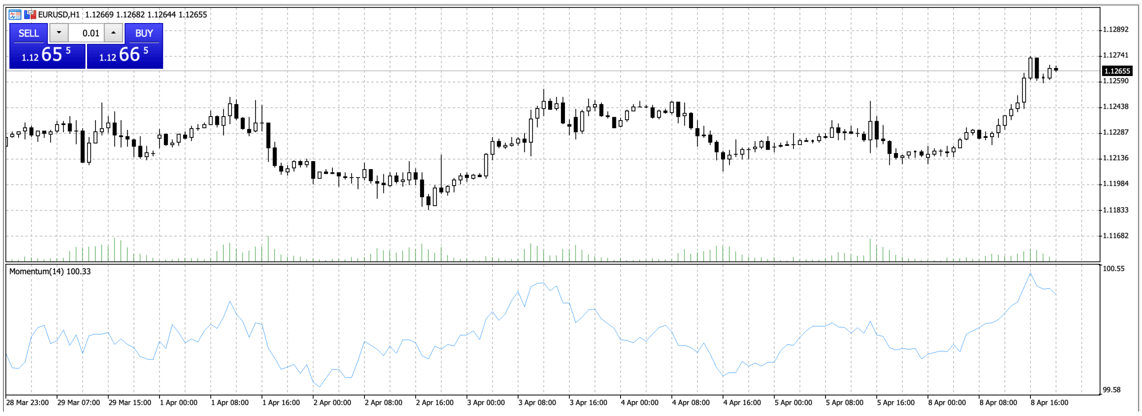

Figure 3.7: Momentum of EUR/USD [4].

is defined as:

$$Momentum = Close - Close(previous, N) \tag{36}$$

- N is the period,
- $Close$ and $Close(previous, N)$ are close price and close price of N periods ago,
- $Momentum$ is the Momentum line.

3.3.5 Relative Strength Index (RSI)

Relative Strength Index (RSI) is a momentum indicator that was developed by J. Welles Wilder in 1978. RSI is based on the ratio between average gain and average loss which is called Relative Strength (RS) [42, 22].

RSI is an oscillator that its values are changing between 0 and 100. It determines overbought and oversold levels in the prices. Formulation is defined as follows with default parameter period equals to 14:

$$AverageGain = \frac{AverageGain(previous) * (N-1) + CurrentGain}{N} \tag{37}$$

$$AverageLoss = \frac{AverageLoss(previous) * (N-1) + CurrentLoss}{N} \tag{38}$$

$$RS = \frac{AverageGain}{AverageLoss} \tag{39}$$

$$RSI = 100 - \frac{100}{1 + RS} \tag{310}$$

- N is the period,
- $AverageGain(Previous)$ and $AverageLoss(Previous)$ are the previous period's average gain and loss,
- $CurrentGain$ and $CurrentLoss$ are the positive and negative absolute difference values between the current and previous period's close price,
- $AverageGain$ and $AverageLoss$ are the current average gain and loss in N periods,
- RS is the relative strength,
- RSI is the relative strength index value.

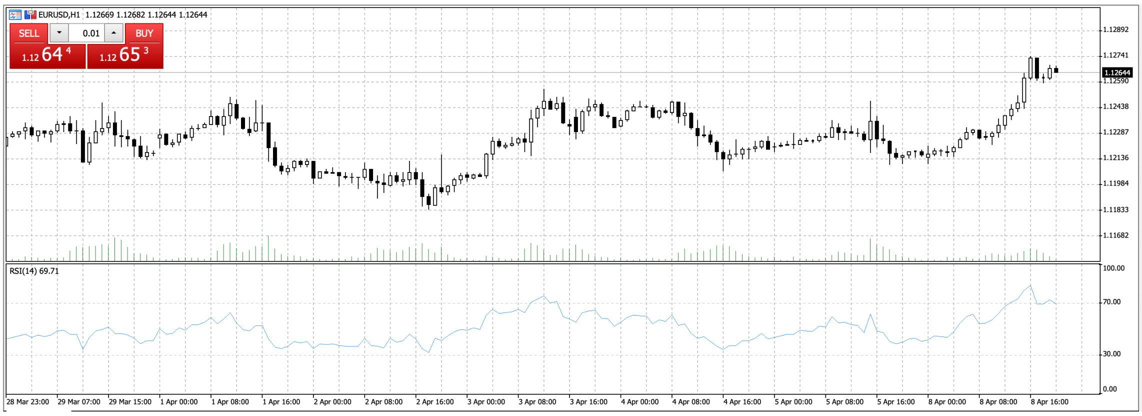

Figure 3.8: RSI of EUR/USD [4].

3.3.6 Bollinger Bands (BB)

Bollinger Bands (BB) are volatility based indicator that were developed by John Bollinger in the 1980s. BB are three bands that provide relative definitions of high and low according to base [43].

While the middle band is the moving average with a specific period, the upper and lower bands are calculated by the standard deviation of the price which are placed above and below of the medium band. The distance between the bands depends on the volatility of the price [43, 44]. In the formulation below, default parameters for period and standard deviation multiplier i.e, 20 and 2, are used:

$$MiddleBand = SMA(Close, 20) \tag{311}$$

$$UpperBand = MiddleBand + SD(Close, 20) * 2 \tag{312}$$

$$LowerBand = MiddleBand - SD(Close, 20) * 2 \tag{313}$$

where

$$SD = \sqrt{\frac{\sum (x_i - \mu)^2}{N}} \tag{314}$$

- SD is the standard deviation,

Figure 3.9: Bollinger Bands of EUR/USD [4].

- x is the data point, μ is the average of the data points, N is the number of data points,
- $SMA(Close, 20)$ is the simple moving average of the close price with period of 20,
- SD is the standard deviation of the close price with period of 20,
- $MiddleBand$ $UpperBand$ and $LowerBand$ correspond to the middle band, upper band and lower band of the BB.

3.3.7 Commodity Channel Index (CCI)

Commodity Channel Index (CCI) is a momentum based indicator that was developed by Donald Lambert in 1980. CCI takes the principle that current prices should be examined with the recent past prices not the distant past ones in order not to confuse present patterns [45]. This indicator can be used to alert a new trend or warn against extreme conditions. Moreover, CCI identifies overbought and oversold conditions [24]. The formulation of CCI is represented as:

$$CCI = \frac{TypicalPrice - SMA(TypicalPtice, N)}{MeanDeviation * L} \tag{315}$$

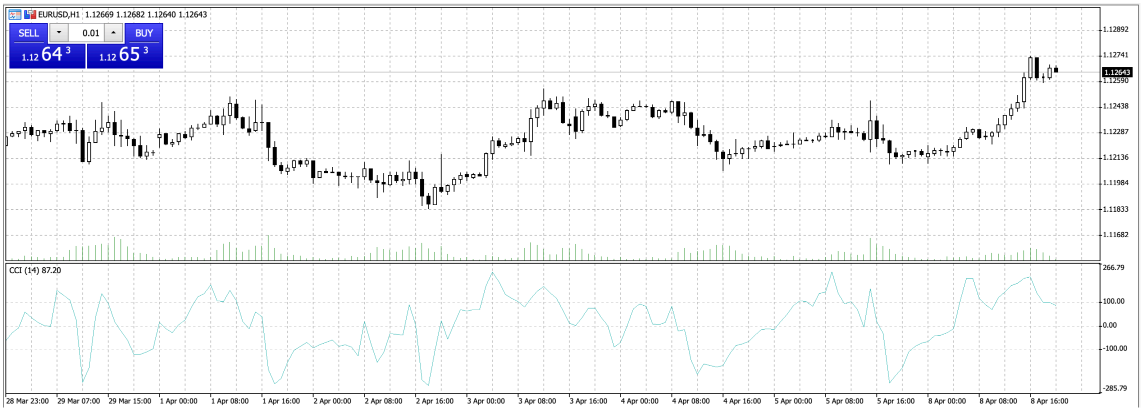

Figure 3.10: CCI of EUR/USD [4].

where

$$TypicalPrice = \frac{High + Low + Close}{3} \tag{316}$$

$$MeanDeviation = \frac{\sum |TypicalPrice - SMA(TypicalPrice, N)|}{N} \tag{317}$$

- N is the period,
- $TypcialPrice$ is the typical price,
- $SMA(TypicalPrice, N)$ is the simple moving average of typical price with a period of N,
- $MeanDeviation$ is the mean deviation,
- L is the Lambert coefficient equals to 0.015.

3.4 Dataset Representation

The interest rates and inflation rates are fundamental indicators for strength of the economy. In the case of low interest rates, individuals are tend to buy investment tools that makes economy stronger. In the opposite case, economy will become fragile. If supply does not meet demand, inflation will occur and higher interest rates take place in this situation [46].

Table 3.1: The macroeconomic data and currency pair used in dataset.

Term	Explanation
$Close(EURUSD)$	Daily Close value of Euro/Dollar currency pair.
$Open(EURUSD)$	Daily Open value of Euro/Dollar currency pair.
$High(EURUSD)$	Daily High value of Euro/Dollar currency pair.
$Low(EURUSD)$	Daily Low value of Euro/Dollar currency pair.
$Inflation\ Rate_{EU}$	Monthly Inflation Rate for European Area
$Inflation\ Rate_{USA}$	Monthly Inflation Rate for United States of America Area
$Interest\ Rate_{GER}$	Monthly Interest Rate in Germany
$Interest\ Rate_{EU}$	Monthly Interest Rate in European Area
$FED\ Funds\ Rate$	Daily Fed Funds Rates
$Close(S\&P\ 500)$	Standard&Poor - Daily Close American Stock Market Index
$Close(DAX)$	Daily Close German Stock Index

German and USA have powerful economies which have high impacts on the currencies. DAX is the German stock index which is decisive on EUR, on the other hand S&P 500 is one the American stock index which directs USD.

In the light of these preliminary information, we decided to construct our dataset with seven macroeconomic factors and close, open, high and low values of the EUR/USD currency pair which are retrieved from [47, 48, 49, 50, 51]. Therefore, we had opportunity to observe the relationship between the currency pair and the macroeconomic factors.

This dataset is created with values beginning from January 2013 until January 2018. This 5-year period contains 1234 data points where markets are open. There are **613** increase and **620** decrease data points. The explanations of each field of the dataset are summarized in Table 3.1. Monthly inflation rates and interest rates are depopulated so as to ensure every values become daily. Table 3.2 shows the values of macroeconomic variables and EUR/USD currency pair and summarizes our dataset.

Table 3.2: The Dataset

Date	DAX	S&P 500	FED Funds Rate	Inflation Rate (EU)	Inflation Rate (USA)	Interest Rate (GER)	Interest Rate (EU)	EUR/USD Close	EUR/USD Open	EUR/USD High	EUR/USD Low
Wed, 2 Jan 2013	7.778,779785	1.462,420044	0,17	2,0	1,9	1,51	2,87	1,3186	1,3205	1,3302	1,3157
Thu, 3 Jan 2013	7.756,439941	1.459,369995	0,17	2,0	1,9	1,51	2,87	1,3048	1,3187	1,3192	1,3046
Fri, 4 Jan 2013	7.776,370117	1.466,469971	0,16	2,0	1,9	1,51	2,87	1,3069	1,3048	1,3091	1,2999
Mon, 7 Jan 2013	7.732,660156	1.461,890015	0,16	2,0	1,9	1,51	2,87	1,3117	1,3072	1,3121	1,3018
Tue, 8 Jan 2013	7.695,830078	1.457,150024	0,15	2,0	1,9	1,51	2,87	1,3083	1,3116	1,3142	1,3058
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
Thu, 21 Dec 2017	13.109,740234	2.684,570068	1,42	1,4	1,8	0,3	1,09	1,1873	1,1871	1,1889	1,1848
Fri, 22 Dec 2017	13.072,790039	2.683,340088	1,42	1,4	1,8	0,3	1,09	1,1859	1,1874	1,1878	1,1817
Wed, 27 Dec 2017	13.070,019531	2.682,620117	1,42	1,4	1,8	0,3	1,09	1,1888	1,1859	1,1912	1,1855
Thu, 28 Dec 2017	12.979,940430	2.687,540039	1,42	1,4	1,8	0,3	1,09	1,1943	1,1888	1,1961	1,1888
Fri, 29 Dec 2017	12.917,639648	2.673,610107	1,33	1,4	1,8	0,3	1,09	1,1998	1,1943	1,2030	1,1937

CHAPTER 4

HYBRID LSTM MODEL: COMBINATION OF MACROECONOMIC INDICATOR MODEL AND TECHNICAL INDICATOR MODEL

4.1 Hybrid Model

Using LSTM structure, we construct hybrid model for forecasting the directional movement in the EUR/USD currency pair. Our hybrid model consists of two separate LSTM models that learn different parameter settings for different input sets. These models are called "macroeconomic lstm model" and "technical lstm model" which will thoroughly be explained in sections 4.1.1 and 4.1.2 respectively.

The main flow for hybrid model shown in the Figure 4.1 can be defined as follows:

1. Pre-processing dataset,
2. Training "macroeconomic lstm model" and post-processing results,
3. Training "technical lstm model" and post-processing results,
4. Combining these models by applying different strategies for using their individual results.

4.1.1 Macroeconomic LSTM Model

This LSTM model (ME_LSTM) is built for investigating effects of macroeconomics factors on the price movement of EUR/USD currency pair. These factors which are explained in detail in section 3.4 are listed below:

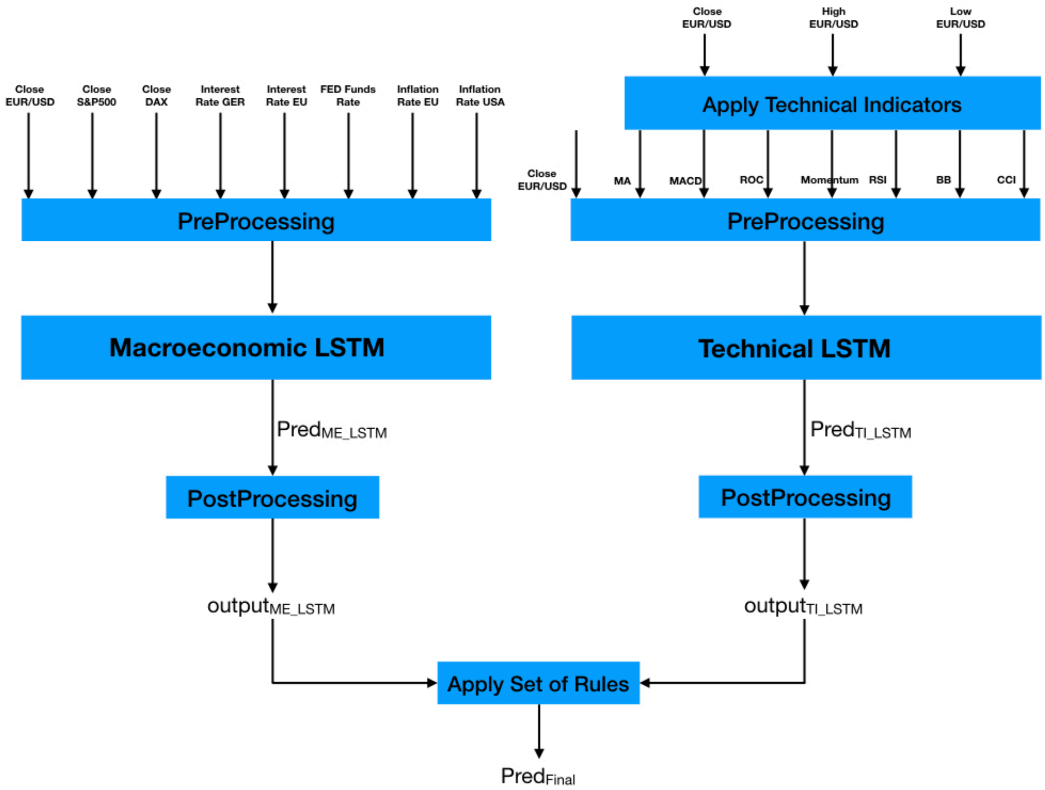

Figure 4.1: Hybrid LSTM Model. Macroeconomic LSTM model is on the left and Technical Indicator LSTM is on the right.

- Interest rates for Germany and EU,
- Fed funds rate,
- Inflation rates in EU and USA,
- Close value of S&P 500 market index,
- Close value of DAX market index.

After pre-processing stage, ME_LSTM model is trained using all these macroeconomic factors together with the close value of EUR/USD.

4.1.2 Technical LSTM Model

This LSTM model (TI_LSTM) is formed by using technical indicators in order to observe their effects on the price movement of EUR/USD currency pair. These technical indicators are listed below:

- MA with a period of 10,
- MACD with short and long term periods of 12 and 26 respectively,
- ROC with a period of 2,
- Momentum with a period of 4,
- RSI with a period of 10,
- BB with period of 20,
- CCI with a period of 20.

After pre-processing stage, TI_LSTM model is trained using these seven technical indicators together with close value of EUR/USD currency pair. They are calculated by using the close, high and low values of EUR/USD currency pair.

Table 4.1: LSTM parameter settings.

LSTM Cell	Learning Rate	Epoch	Batch Size	Loss Function	Optimizer	Activation
64	0.01	1024	32	Categorical Crossentropy	Adamax	Softmax

4.1.3 Macroeconomic and Technical LSTM Model

This LSTM model (ME_TI_LSTM) is formed by using both macroeconomic and technical indicators to see the effects of combined indicators.

After pre-processing stage, ME_TI_LSTM is trained using macroeconomic variables and technical indicators mentioned above together with close value of EUR/USD currency pair.

4.1.4 Training a Classifier

We train Macroeconomic LSTM, Technical Indicator LSTM and Macroeconomic and Technical Indicator LSTM models with using the settings shown in the Table 4.1. The dataset is split into training and test sets, with the ratio of 80% and 20% respectively. Training phase is carried out with different number of iterations (50, 100 and 150).

CHAPTER 5

EXPERIMENTS

In this chapter, we describe our experiments that we carried out. Firstly, we will mention about the pre-processing part which will describe how we set labels in detail. Secondly, post-processing part will be explained which is responsible for evaluation policies. The third section is going to be about experiment environment. After that, we will go through experiments. Experiments involves 1-day, 3-days and 5-days ahead predictions of directional movement of EUR/USD currency pair. We will introduce results of individual LSTM models as baselines and make comparison with our proposed hybrid model. Moreover, we built another LSTM model as a baseline which takes both macroeconomic variables and technical indicators as inputs for training. We also compare our approach with this model. For validation of our proposed model, we will interpret the performances on 3-days and 5-days ahead forecasting. We also mention about the number of total transactions we made on test data for each experiment which shows the percentage of successful transaction. The final step will be about experiments conducted on the extended dataset for only hybrid model and we will summarize the results of all experiments in detail.

5.1 Pre-processing

Before training our classifier, we apply a couple of processing on data. These pre-processing techniques involve labeling data and normalization.

5.1.1 Data Labeling

Our data points are labeled based on the histogram analysis and entropy approach. At the end of these operations, we divide data points in three classes which are namely:

- $Class\_inc$: **increase** which is greater than certain threshold,
- $Class\_dec$: **decrease** which is less than certain threshold,
- $Class\_noact$: **no change** which is in the remaining between [-threshold, threshold].

5.1.1.1 Histogram Analysis

Histogram analysis was made on the close price of EUR/USD currency pair for visualizing the price changes occur in data ($Close(t) - Close(t-n)$). We placed close differences into 10 bins in a sorted manner and counted the number of differences in each bin. We determined the upper value of threshold τ by setting an early cut-off value (85%) to these differences that prevent us from looking all data points. In this way, this analysis helps us in calculating threshold τ value that defines class labels by applying Algorithm 1.

5.1.1.2 Threshold Calculation

Threshold was calculated based on entropy approach. Entropy is related with the distribution of data and the formulation is defined as 5.1.1.2.

$$Entropy = -\sum \mathbf{p}_i * \log \mathbf{p}_i \tag{51}$$

In order to get balanced distribution, we calculate entropy of class distribution in an iterative way. By limiting iteration number to upper bound of threshold found in histogram analysis, we try to find final threshold τ which maximizes entropy. The Algorithm 2 shows the details of our approach.

Algorithm 1 Histogram Analysis.

procedure HISTOGRAM_ANALYSIS($close\_diff$) ▷ close(t) - close(t - n)
$hist\_vals, bin\_edges \leftarrow perform\_histogram(close\_diff, 10)$ ▷ hist_vals, bin_edges are sorted arrays
$temp\_sum \leftarrow 0$
$sum\_hist\_vals \leftarrow sum(hist\_vals)$
while $i \leq |hist\_vals|$ **do**
$temp\_sum \leftarrow temp\_sum + hist\_vals[i]$
if $temp\_sum/sum\_hist\_vals > 0.85$ **then**
break
$i \leftarrow i + 1$
$threshold\_upper\_bound \leftarrow bin\_edges[i]$
return $threshold\_upper\_bound$

Algorithm 2 Threshold Calculation.

procedure CALCULATE_THRESHOLD($close\_diff$) ▷ close(t) - close(t - n)
$threshold\_upper\_bound \leftarrow HISTOGRAM\_ANALYSIS(close\_diff)$
$temp\_threshold \leftarrow 0$
$best\_entropy = -Infinity$
while $temp\_threshold < threshold\_upper\_bound$ **do**
$labels \leftarrow \{0\}$
$indexes\_incr \leftarrow indexOf(close\_diff > temp\_threshold)$
$indexes\_decr \leftarrow indexOf(close\_diff < temp\_threshold)$
$labels\{indexes\_incr\} \leftarrow 2$
$labels\{indexes\_decr\} \leftarrow 1$
$entropy \leftarrow calculate\_entropy(labels)$
if $entropy > best\_entropy$ **then**
$best\_entropy \leftarrow entropy$
$threshold \leftarrow temp\_threshold$
$temp\_threshold \leftarrow temp\_threshold + 0.00001$
return $threshold$

5.1.2 Normalization

We applied Robust Scaler from scikit learn API in Python for normalization. Removing median and scaling data with respect to interquantile range (1st and 3rd quantile) make this method robust to outliers [52].

5.2 Post-processing

We can divide post-processing section into two part according to set of conditions/rules they have. We called these as Stage-1 and Stage-2 processing. The former is responsible for changing labels of test data and the latter determines the final output our proposed model.

5.2.1 Stage-1 Processing

Before testing our model, we manipulated the labels of our test data according to the specific conditions below:

- testing data point is labeled as $class\_noact$,
- the original state of data is increase/decrease and,
- the model's prediction is the same as the original state.

If these conditions were met, we changed the label of the data to the original state. Otherwise, no alteration was made.

5.2.2 Stage-2 Processing

The purpose of this processing is to determine the final output. We combined the predictions of ME_LSTM and TI_LSTM models with predefined set of rules which are listed below:

Table 5.1: Sample Table for profit_accuracy calculation.

	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	-	False_dec_noact	False_inc_noact
True(dec)	-	True_dec	False_inc_dec
True(inc)	-	False_dec_inc	True_inc

- If one of the model' prediction was `class_noaction`, final decision would be `class_noaction`,
- If both models agreed on the label, we set the final decision as the prediction of individual's,
- If their predictions were different, we chose the final decision as the one whose prediction was made with the highest probability. In case of having same probability for predictions, we preferred the prediction of TI_LSTM model.

5.3 Experiment Environment

We conducted the experiments on a Mac-Book Pro with a 2.7GHz dual-core Intel Core i5 processor, 8GB memory and 256GB disk space.

The implementation of our proposed model was made in Python by using deep learning library Keras [53] and machine learning library scikit-learn.

5.4 A Performance Metric

We introduced a new performance metric to measure our proposed method. The **profit_accuracy** is the accuracy that is related to the number of increases and decreases of the predicted labels. We could interpret this metric as that gives us the ratio of number of profitable transactions over the total transactions which is defined by using Table 5.1:

$$ProfitAccuracy = \frac{True\_dec + True\_inc}{False\_dec\_noact + False\_inc\_noact + True\_dec + False\_inc\_dec + False\_dec\_inc + True\_inc} \quad (52)$$

5.5 Preliminary Experiments

Our initial study uses the regression method to determine the directional movement. In this approach, we try to predict the exact value of the prices. After comparison with the past values made by using these predictions, we reach the decision whether there will be increase or decrease on the future price. With the help of the method introduced in [5], we examine the effect of the regression on the directional movement prediction in financial time series.

The datasets consist of historical close values of EUR/GBP, GBP/USD, USD/CHF, GBP/CHF, EUR/USD and EUR/CHF currency pairs. These 15-minute period contains 2499 data points beginning from 25 November 2015 to 31 December 2015.

In these experiments, we try to measure the accuracy in terms of directional symmetry. Using Equation 53, several conditions are created with multiplication or addition of custom values to all data points. After that, manipulated data points are given as features to the model with normalization and without normalization. Experiments are conducted for 1-day, 3-days, 5-days, 10-days and 20-days ahead prediction of currency pairs' direction. Even if this model exhibits a good performance in terms of mean square error, it has poor accuracy for directional symmetry around 50%. In Table 5.2, we can summarize the average accuracy values of six currency pairs for all cases in detail.

$$f(x_i) = C * x_i + n \tag{53}$$

- x_i is the close value of currency pair,
- C is the multiplicative constant,
- n is the additive constant.

5.6 The Dataset Statistics

After applying labeling algorithm, we obtain a balanced distribution of three classes over the dataset. This algorithm calculates different threshold values for each period

Table 5.2: Preliminary Experiments. Average accuracy values of six currency pairs.

	Normalization			Without Normalization
Multiplicative - Additive Constants	C: 10^4 - n: 0	C: 10^5 - n: 0	C: 10^5 - n: 10	C: 10^5 - n: 10
Step-1	49.33%	49.33%	49.67%	52.50%
Step-3	49.33%	49.33%	49.67%	49.50%
Step-5	49.33%	49.17%	49.67%	50.17%
Step-10	49.17%	49.17%	49.50%	50.17%
Step-20	49.00%	49.00%	49.33%	48.83%

Table 5.3: The Dataset Statistics. (Training - Test sets respectively)

	Threshold	**# of no_action**	**# of decrease**	**# of increase**
1-Day Ahead	0.0023	**412** (334 - 78)	**400** (327 - 73)	**402** (310 - 92)
3-Days Ahead	0.0040	**413** (317 - 96)	**414** (357 - 57)	**385** (295 - 90)
5-Days Ahead	0.0055	**400** (311 - 89)	**422** (370 - 52)	**388** (287 - 101)

and forms different sets of class distribution. For predictions of different periods, calculated thresholds and corresponding number of data points (explicitly via training and test set) in each class are represented in Table 5.3.

This table indicates us class distributions of training and test data have different characteristics compared to the all data. While class $decrease$ has high ratio in training set and low ratio in test set, class $increase$ has opposite behavior. Class no_action is, on the other hand, more stable in both sets. This is due to the fact that the split is made between training and test sets without shuffling the dataset in order to preserve the order of data points.

Table 5.4: ME_LSTM Model: 1-day ahead prediction results.

Iteration=50				Iteration=100				Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)		Pred(no_act)	Pred(dec)	Pred(inc)		Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	0	**38**	**0**	True(no_act)	55	**3**	**7**	True(no_act)	0	**27**	**14**
True(dec)	0	**113**	**0**	True(dec)	55	**16**	**7**	True(dec)	0	**80**	**19**
True(inc)	0	**92**	**0**	True(inc)	73	**10**	**17**	True(inc)	0	**64**	**39**

Table 5.5: ME_LSTM Model: 1-day ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	46.50%	243/243
Iteration=100	55.00%	60/243
Iteration=150	48.97%	243/243
Average	**50.16%**	**182/243**

5.7 Forecasting 1 Day Ahead

In this section, we present one day ahead directional forecasting on EUR/USD currency pair for macroeconomic lstm, technical lstm, macroeconomic and technical lstm and our proposed model. In the following, the profit_accuracy and number of predicted transactions are reported for each experiment conducted.

5.7.1 Macroeconomic LSTM Model Results

Here, we express the results of me_lstm model on 1-day ahead predictions. This model sometimes made no predictions on specific classes and showed huge variance on the number of transactions. The experiments' results are elaborated in the following.

In Table 5.4, we present the confusion matrices for each iteration that show the distribution of predicted classes. We see profit_accuracy results have small variance and we get **50.16%** $\pm$ **4.37%** on average. Additionally, the average predicted transaction number is **182** and this corresponds to **74.90%** of test data. These results are summarized in Table 5.5.

Table 5.6: TI_LSTM Model: 1-day ahead prediction results.

Iteration=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	22	**13**	**14**
True(dec)	12	**48**	**28**
True(inc)	20	**39**	**47**

Iteration=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	26	**12**	**12**
True(dec)	19	**46**	**23**
True(inc)	37	**32**	**36**

Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	21	**14**	**7**
True(dec)	21	**47**	**24**
True(inc)	24	**38**	**47**

Table 5.7: TI_LSTM Model: 1-day ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	50.26%	189/243
Iteration=100	50.93%	161/243
Iteration=150	53.11%	177/243
Average	**51.43%**	**175.67/243**

5.7.2 Technical LSTM Model Results

Here, we express the the result of ti_lstm model on 1-day ahead prediction. Predicted class distribution over test data was balanced for this model. The experiments' results are elaborated in the following.

In Table 5.6 confusion matrices show the distribution of predicted classes. We see profit_accuracy results are close in each iteration and we get **51.43%** $\pm$ **1.49%** on average. Additionally, the average predicted transaction number is **175.67** and this corresponds to **72.29%** of test data. These results are summarized in Table 5.7.

5.7.3 Macroeconomic and Technical LSTM Model Results

Here, we express the the result of me_ti_lstm model on 1-day ahead prediction. We observed that this model's predictions were tend to one class in some cases and number of transactions show some variances for each iterations. The experiments' results are elaborated in the following.

In Table 5.8, we present the confusion matrices for each iteration that show the distribution of predicted classes. We see profit_accuracy results have small variance and

Table 5.8: ME_TI_LSTM Model: 1-day ahead prediction results.

Iteration=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	4	**34**	**2**
True(dec)	2	**104**	**3**
True(inc)	6	**82**	**6**

Iteration=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	31	**18**	**1**
True(dec)	27	**68**	**3**
True(inc)	46	**40**	**9**

Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	15	**30**	**2**
True(dec)	13	**84**	**5**
True(inc)	20	**67**	**7**

Table 5.9: ME_TI_LSTM Model: 1-day ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	47.62%	231/243
Iteration=100	55.40%	139/243
Iteration=150	46.67%	195/243
Average	**49.89%**	**188.33/243**

we get **49.89**% ± **4.79**% on average. Additionally, the average predicted transaction number is **188.33** and this corresponds to **77.50**% of test data. These results are summarized in Table 5.9.

5.7.4 Hybrid LSTM Model Results

In this section, we show our proposed method's 1-day ahead prediction results. We compare the results with models me_lstm, ti_lstm and me_ti_lstm. We combine the result of different iteration combinations of baseline models that forms our proposed model.

Stage-1 processing to baseline models produces two nine sub-tables based on the modification of true labels with respect to me_lstm and ti_lstm results. Table 5.10 and Table 5.11 consist of nine sub-tables showing confusion matrices built during testing process. The profit_accuracy values and the number of transactions for each cases are summarized in Table 5.12.

In general, we observe that predicted class distribution is well except for the cases when $Iteration1$ equals to 100. The total number of decrease and increase predictions are changing in the range $[26, 137]$. In the average, we get that value equals to **92**

Table 5.10: Hybrid Model - Confusion Matrices (modification based on me_lstm): 1-day ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	24	**14**	**0**
True(dec)	23	**71**	**19**
True(inc)	59	**7**	**26**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	25	**13**	**0**
True(dec)	32	**72**	**9**
True(inc)	69	**9**	**14**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	21	**17**	**0**
True(dec)	35	**62**	**16**
True(inc)	62	**5**	**25**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	65	**0**	**0**
True(dec)	63	**10**	**5**
True(inc)	83	**2**	**15**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	63	**1**	**1**
True(dec)	66	**8**	**4**
True(inc)	88	**2**	**10**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	61	**1**	**3**
True(dec)	69	**5**	**4**
True(inc)	79	**2**	**19**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	29	**11**	**1**
True(dec)	26	**60**	**13**
True(inc)	60	**5**	**38**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	28	**9**	**4**
True(dec)	39	**53**	**7**
True(inc)	70	**9**	**24**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	23	**12**	**6**
True(dec)	37	**51**	**11**
True(inc)	60	**4**	**39**

i.e., transaction ratio is $92/243 = 37.86\%$ for both cases. Moreover, we obtain the average profit_accuracy over nine cases are **72.69**$\%$ $\pm$ **3.06**$\%$ and **73.48**$\%$ $\pm$ **5.64**$\%$ for me_lstm and ti_lstm based modified hybrid models respectively where 3.06 and 4.37 represent standard deviation. When we analyze the results, we reach the followings below:

- Predictions of ti_lstm and hybrid model are more balanced than me_lstm and me_ti_lstm,

- Baseline models make more transactions (by a number of **90** out of 243 on average),

- Hybrid model predicts more accurately in a significant amount (**22.60**$\%$ better on average).

Table 5.11: Hybrid Model - Confusion Matrices (modification based on ti_lstm): 1-day ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	35	**0**	**14**
True(dec)	12	**57**	**19**
True(inc)	59	**7**	**40**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	38	**0**	**12**
True(dec)	19	**60**	**9**
True(inc)	69	**9**	**27**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	35	**0**	**7**
True(dec)	21	**55**	**16**
True(inc)	62	**5**	**42**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	45	**1**	**3**
True(dec)	76	**7**	**5**
True(inc)	90	**2**	**14**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	47	**1**	**2**
True(dec)	77	**7**	**4**
True(inc)	93	**2**	**10**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	38	**3**	**1**
True(dec)	81	**7**	**4**
True(inc)	90	**2**	**17**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	39	**3**	**7**
True(dec)	21	**54**	**13**
True(inc)	55	**5**	**46**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	43	**2**	**5**
True(dec)	29	**52**	**7**
True(inc)	65	**9**	**31**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	33	**4**	**5**
True(dec)	29	**52**	**11**
True(inc)	58	**4**	**47**

Table 5.12: Hybrid Model: 1-day ahead predictions.

Iterations		Hybrid Model - modification based on me_lstm		Hybrid Model - modification based on ti_lstm	
me_lstm	**ti_lstm**	**profit_accuracy**	**# of transactions**	**profit_accuracy**	**# of transactions**
50	50	70.80%	137/243	70.80%	137/243
50	100	73.50%	117/243	74.36%	117/243
50	150	69.60%	125/243	77.60%	125/243
100	50	78.13%	32/243	65.63%	32/243
100	100	69.23%	26/243	65.38%	26/243
100	150	70.59%	34/243	70.59%	34/243
150	50	76.56%	128/243	78.13%	128/243
150	100	72.64%	106/243	78.30%	106/243
150	150	73.17%	123/243	80.49%	123/243
Average		**72.69%**	**92/243**	**73.48%**	**92/243**

Table 5.13: ME_LSTM Model: 3-days ahead prediction results.

Iteration=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	0	**4**	**39**
True(dec)	0	**17**	**47**
True(inc)	1	**13**	**122**

Iteration=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	22	**24**	**11**
True(dec)	12	**59**	**11**
True(inc)	27	**37**	**40**

Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	1	**45**	**6**
True(dec)	1	**87**	**9**
True(inc)	5	**82**	**7**

5.8 Forecasting 3 Days Ahead

In this section, we present three days ahead directional forecasting on EUR/USD currency pair for macroeconomic lstm, technical lstm, macroeconomic and technical lstm and our proposed model. These experiments could be thought as validation for our proposed model. In this perspective, we will discuss the performances of main experiments in the Section 5.11. In the following, the profit_accuracy and number of predicted transactions are reported for each experiment conducted.

5.8.1 Macroeconomic LSTM Model Results

Here, we express the results of me_lstm model on 3-days ahead predictions. This model sometimes made rare predictions on specific classes and performed different number of transactions for each iterations. The experiments' results are elaborated in the following.

In Table 5.13, we present the confusion matrices for each iteration that show the distribution of predicted classes. We see profit_accuracy results have high variance and we get **50.56**% $\pm$ **9.41**% on average. Additionally, the average predicted transaction number is **220** and this corresponds to **90.54**% of test data. These results are summarized in Table 5.14.

5.8.2 Technical LSTM Model Results

Here, we express the the results of ti_lstm model on 3-days ahead predictions. Predicted class distribution over test data was balanced for this model. The experiments results are elaborated in the following.

Table 5.14: ME_LSTM Model: 3-days ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	57.44%	242/243
Iteration=100	54.40%	182/243
Iteration=150	39.83%	236/243
Average	**50.56%**	**220/243**

Table 5.15: TI_LSTM Model: 3-days ahead prediction results.

Iteration=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	34	**24**	**7**
True(dec)	24	**47**	**8**
True(inc)	28	**50**	**21**

Iteration=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	26	**23**	**11**
True(dec)	12	**56**	**11**
True(inc)	25	**49**	**30**

Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	35	**21**	**8**
True(dec)	22	**48**	**8**
True(inc)	40	**34**	**27**

In Table 5.15 confusion matrices show the distribution of predicted classes. We see profit_accuracy results have small variance and we get **47.49**% $\pm$ **4,04**% on average. Additionally, the average predicted transaction number is **161** and this corresponds to **66.26**% of test data. These results are summarized in Table 5.16.

5.8.3 Macroeconomic and Technical LSTM Model Results

Here, we express the the result of me_ti_lstm model on 3-days ahead prediction. We observed that this models predictions were tend to one class in some cases and number of transactions show some variances for each iterations. The experiments results are elaborated in the following.

Table 5.16: TI_LSTM Model: 3-days ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	43.31%	157/243
Iteration=100	47.78%	180/243
Iteration=150	51.37%	146/243
Average	**47.49%**	**161/243**

Table 5.17: ME_TI_LSTM Model: 3-days ahead prediction results.

Iteration=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	2	**49**	**0**
True(dec)	1	**101**	**0**
True(inc)	6	**84**	**0**

Iteration=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	5	**45**	**0**
True(dec)	4	**94**	**2**
True(inc)	8	**80**	**5**

Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	31	**37**	**0**
True(dec)	15	**70**	**0**
True(inc)	33	**57**	**0**

Table 5.18: ME_TI_LSTM Model: 3-days ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	43.16%	234/243
Iteration=100	43.81%	226/243
Iteration=150	42.68%	164/243
Average	**43.22%**	**208/243**

In Table 5.17, we present the confusion matrices for each iteration that show the distribution of predicted classes. We see profit_accuracy results are so close and we get **43.22**% $\pm$ **0.56**% on average. Additionally, the average predicted transaction number is **208** and this corresponds to **85.60**% of test data. These results are summarized in Table 5.18.

5.8.4 Hybrid LSTM Model Results

In this section, we show our proposed method's 3-days ahead prediction results. We compare the results with models me_lstm, ti_lstm and me_ti_lstm. We combine the result of different iteration combinations of baseline models that forms our proposed model.

Stage-1 processing to baseline models produces two nine sub-tables based on the modification of true labels with respect to me_lstm and ti_lstm results. Table 5.19 and Table 5.20 consist of nine sub-tables showing confusion matrices built during testing process. The profit_accuracy values and the number of transactions for each cases are summarized in Table 5.21.

The total number of decrease and increase predictions are changing in the range

Table 5.19: Hybrid Model - Confusion Matrices (modification based on me_lstm): 3-days ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	20	**3**	**20**
True(dec)	32	**25**	**7**
True(inc)	54	**27**	**55**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	24	**0**	**19**
True(dec)	22	**34**	**8**
True(inc)	42	**37**	**57**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	22	**1**	**20**
True(dec)	33	**28**	**3**
True(inc)	63	**24**	**49**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	52	**3**	**2**
True(dec)	39	**42**	**1**
True(inc)	81	**11**	**12**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	48	**6**	**3**
True(dec)	27	**46**	**9**
True(inc)	83	**8**	**13**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	51	**2**	**4**
True(dec)	40	**37**	**5**
True(inc)	89	**2**	**13**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	38	**8**	**6**
True(dec)	42	**50**	**5**
True(inc)	77	**9**	**8**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	33	**14**	**5**
True(dec)	24	**65**	**8**
True(inc)	73	**6**	**15**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	36	**11**	**5**
True(dec)	38	**54**	**5**
True(inc)	73	**8**	**13**

$[63, 155]$. In the average, we get that value equals to **103.44** i.e., transaction ratio is $103.44/243 = 42.57\%$ for both cases. Moreover, we obtain the average profit_accuracy over nine cases are **67.95**% $\pm$ **7.31**% and **68.67**% $\pm$ **8.35**% for me_lstm and ti_lstm based modified hybrid models respectively. When we analyze the results, we reach the followings below:

- Predictions of ti_lstm and hybrid model are more balanced than me_lstm and me_ti_lstm,
- Baseline models make more transactions (by a number of **92.89** out of 243 on average),
- Hybrid model predicts more accurately in a significant amount (**21.09**% better on average).

Table 5.20: Hybrid Model - Confusion Matrices (modification based on ti_lstm): 3-days ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	41	**23**	**1**
True(dec)	28	**44**	**7**
True(inc)	37	**27**	**35**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	37	**21**	**2**
True(dec)	20	**51**	**8**
True(inc)	31	**37**	**36**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	43	**18**	**3**
True(dec)	30	**45**	**3**
True(inc)	45	**24**	**32**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	57	**6**	**2**
True(dec)	36	**42**	**1**
True(inc)	79	**11**	**9**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	52	**3**	**5**
True(dec)	26	**44**	**9**
True(inc)	80	**8**	**16**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	59	**3**	**2**
True(dec)	34	**39**	**5**
True(inc)	87	**2**	**12**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	54	**4**	**7**
True(dec)	25	**49**	**5**
True(inc)	78	**9**	**12**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	45	**4**	**11**
True(dec)	12	**59**	**8**
True(inc)	73	**6**	**25**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	52	**4**	**8**
True(dec)	22	**51**	**5**
True(inc)	73	**8**	**20**

Table 5.21: Hybrid Model: 3-days ahead predictions.

Iterations		Hybrid Model - modification based on me_lstm		Hybrid Model - modification based on ti_lstm	
me_lstm	**ti_lstm**	**profit_accuracy**	**# of transactions**	**profit_accuracy**	**# of transactions**
50	50	58.39%	137/243	57.66%	137/243
50	100	58.71%	155/243	56.13%	155/243
50	150	61.60%	125/243	61.60%	125/243
100	50	76.06%	71/243	71.83%	71/243
100	100	69.41%	85/243	70.59%	85/243
100	150	79.37%	63/243	80.95%	63/243
150	50	67.44%	86/243	70.93%	86/243
150	100	70.80%	113/243	74.34%	113/243
150	150	69.79%	96/243	73.96%	96/243
Average		**67.95%**	**103.44/243**	**68.67%**	**103.44/243**

Table 5.22: ME_LSTM Model: 5-days ahead prediction results.

Iteration=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	2	**38**	**0**
True(dec)	0	**100**	**0**
True(inc)	5	**95**	**2**

Iteration=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	0	**32**	**12**
True(dec)	0	**80**	**9**
True(inc)	0	**75**	**34**

Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	5	**35**	**0**
True(dec)	1	**99**	**0**
True(inc)	8	**91**	**3**

5.9 Forecasting 5 Days Ahead

In this section, we present five days ahead directional forecasting on EUR/USD currency pair for macroeconomic lstm, technical lstm, macroeconomic and technical lstm and our proposed model. These experiments could also be thought as a validation for our proposed model. In this perspective, we will discuss the performances of main experiments in the Section 5.11. In the following, the profit_accuracy and number of predicted transactions are reported for each experiment conducted.

5.9.1 Macroeconomic LSTM Model Results

Here, we express the results of me_lstm model on 5-days ahead predictions. We observed that this model's predictions were tend to one class in some cases and number of transactions are close for each iterations. The experiments results are elaborated in the following.

In Table 5.22, we present the confusion matrices for each iteration that show the distribution of predicted classes. We see profit_accuracy results are close and we get **45.08**% $\pm$ **1.88**% on average. Additionally, the average predicted transaction number is **235** and this corresponds to **97.11**% of test data. These results are summarized in Table 5.23.

5.9.2 Technical LSTM Model Results

Here, we express the the results of ti_lstm model on 5-days ahead predictions. Predicted class distribution over test data was balanced for this model. The experiments results are elaborated in the following.

Table 5.23: ME_LSTM Model: 5-days ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	43.40%	235/242
Iteration=100	47.11%	242/242
Iteration=150	44.74%	228/242
Average	**45.08%**	**235/242**

Table 5.24: TI_LSTM Model: 5-days ahead prediction results.

Iteration=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	15	**26**	**9**
True(dec)	16	**56**	**9**
True(inc)	24	**53**	**34**

Iteration=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	23	**26**	**9**
True(dec)	13	**54**	**12**
True(inc)	30	**56**	**19**

Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	30	**18**	**6**
True(dec)	18	**50**	**11**
True(inc)	30	**54**	**25**

In Table 5.24 confusion matrices show the distribution of predicted classes. We see profit_accuracy results have small variance and we get **45.11**% $\pm$ **3.37**% on average. Additionally, the average predicted transaction number is **175.67** and this corresponds to **72.29**% of test data. These results are summarized in Table 5.25.

5.9.3 Macroeconomic and Technical LSTM Model Results

Here, we express the the result of me_ti_lstm model on 5-days ahead prediction. We observed that this models predictions were tend to neglect one class (which is $increase$) and number of transactions show huge variances for each iterations. The experiments results are elaborated in the following.

Table 5.25: TI_LSTM Model: 5-days ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	48.13%	187/242
Iteration=100	41.48%	176/242
Iteration=150	45.73%	164/242
Average	**45.11%**	**175.67/242**

Table 5.26: ME_TI_LSTM Model: 5-days ahead prediction results.

Iteration=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	52	**16**	**0**
True(dec)	38	**33**	**0**
True(inc)	71	**29**	**3**

Iteration=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	11	**34**	**0**
True(dec)	8	**87**	**0**
True(inc)	17	**84**	**1**

Iteration=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	26	**26**	**0**
True(dec)	9	**79**	**0**
True(inc)	35	**66**	**1**

Table 5.27: ME_TI_LSTM Model: 5-days ahead result summary.

	profit_accuracy	**# of transactions**
Iteration=50	44.44%	81/242
Iteration=100	42.72%	206/242
Iteration=150	46.51%	172/242
Average	**44.56%**	**153/242**

In Table 5.26, we present the confusion matrices for each iteration that show the distribution of predicted classes. We see profit_accuracy results are close and we get **44.56%** $\pm$ **1.90%** on average. Additionally, the average predicted transaction number is **153** and this corresponds to **63.22%** of test data. These results are summarized in Table 5.27.

5.9.4 Hybrid LSTM Model Results

In this section, we show our proposed method's 5-days ahead prediction results. We compare the results with models me_lstm, ti_lstm and me_ti_lstm. We combine the results of different iteration combinations of baseline models that forms our proposed model.

Stage-1 processing to baseline models produces two nine sub-tables based on the modification of true labels with respect to me_lstm and ti_lstm results. Table 5.28 and Table 5.29 consist of nine sub-tables showing confusion matrices built during testing process. The profit_accuracy values and the number of transactions for each cases are summarized in Table 5.30.

The total number of decrease and increase predictions are close to each other chang-

Table 5.28: Hybrid Model - Confusion Matrices (modification based on me_lstm): 5-days ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	32	**8**	**0**
True(dec)	27	**67**	**6**
True(inc)	78	**10**	**14**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	37	**3**	**0**
True(dec)	25	**68**	**7**
True(inc)	86	**6**	**10**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	32	**8**	**0**
True(dec)	33	**62**	**5**
True(inc)	84	**7**	**11**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	33	**5**	**6**
True(dec)	25	**59**	**5**
True(inc)	72	**7**	**30**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	32	**2**	**5**
True(dec)	27	**57**	**5**
True(inc)	79	**7**	**23**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	37	**4**	**3**
True(dec)	33	**52**	**4**
True(inc)	78	**10**	**21**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	32	**8**	**0**
True(dec)	27	**67**	**6**
True(inc)	77	**9**	**16**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	38	**2**	**0**
True(dec)	25	**68**	**7**
True(inc)	85	**7**	**10**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	34	**6**	**0**
True(dec)	35	**60**	**5**
True(inc)	84	**7**	**11**

ing in the range $[89, 112]$. In the average, we get that value equals to **98.44** i.e., transaction ratio is $98.44/242 = 40.68\%$ for both cases. Moreover, we obtain the average profit_accuracy over nine cases are **79.73**% $\pm$ **2.15**% and **79.11**% $\pm$ **2.48**% for me_lstm and ti_lstm based modified hybrid models respectively. When we analyze the results, we reached the followings below:

- Predictions of ti_lstm and hybrid model are more balanced than me_lstm and me_ti_lstm,
- Baseline models make more transactions (by a number of **89.45** out of 242 on average),
- Hybrid model predicts more accurately in a significant amount (**34.50**% better on average).

Table 5.29: Hybrid Model - Confusion Matrices (modification based on ti_lstm): 5-days ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	42	**0**	**8**
True(dec)	16	**59**	**6**
True(inc)	79	**10**	**22**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	49	**0**	**9**
True(dec)	13	**59**	**7**
True(inc)	86	**6**	**13**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	48	**0**	**6**
True(dec)	18	**56**	**5**
True(inc)	83	**7**	**19**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	42	**2**	**6**
True(dec)	17	**59**	**5**
True(inc)	71	**7**	**33**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	50	**3**	**5**
True(dec)	17	**57**	**5**
True(inc)	76	**7**	**22**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	51	**1**	**2**
True(dec)	22	**53**	**4**
True(inc)	75	**10**	**24**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	41	**0**	**9**
True(dec)	17	**58**	**6**
True(inc)	78	**9**	**24**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	49	**0**	**9**
True(dec)	13	**59**	**7**
True(inc)	86	**7**	**12**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	49	**0**	**5**
True(dec)	19	**55**	**5**
True(inc)	85	**7**	**17**

Table 5.30: Hybrid Model: 5-days ahead predictions.

Iterations		Hybrid Model - modification based on me_lstm		Hybrid Model - modification based on ti_lstm	
me_lstm	**ti_lstm**	**profit_accuracy**	**# of transactions**	**profit_accuracy**	**# of transactions**
50	50	77.14%	105/242	77.14%	105/242
50	100	82.98%	94/242	76.60%	94/242
50	150	78.49%	93/242	80.65%	93/242
100	50	79.46%	112/242	82.14%	112/242
100	100	80.81%	99/242	79.80%	99/242
100	150	77.66%	94/242	81.91%	94/242
150	50	78.30%	106/242	77.36%	106/242
150	100	82.98%	94/242	75.53%	94/242
150	150	79.78%	89/242	80.90%	89/242
Average		**79.73%**	**98.44/242**	**79.11%**	**98.44/242**

Table 5.31: The Extended Dataset Statistics. (Training - Test sets respectively)

	Threshold	# of no_action	# of decrease	# of increase
1-Day Ahead	0.0022	**497** (438 - 59)	**515** (464 - 51)	**507** (465 - 42)
3-Days Ahead	0.0040	**507** (451 - 56)	**527** (476 - 51)	**483** (438 - 45)
5-Days Ahead	0.0054	**503** (448 - 55)	**532** (483 - 49)	**480** (432 - 48)

5.10 Experiments on Extended Dataset

For validation purpose, we extend our dataset from 1 January 2018 to 1 April 2019. In this way, new dataset has 1539 data points which contains **761** increase and **777** decrease in overall. Applying labeling algorithm, we form a dataset which has balanced distribution of three classes. Table 5.31 represents statistics of extended dataset in detail:

The extended dataset is split into training and test sets, with the ratio of 90% and 10% respectively. We report the 1-day, 3-days and 5-days ahead prediction results of our hybrid model on the extended dataset in the subsections below.

5.10.1 Forecasting 1 Day Ahead

In this section, we show our proposed method's 1-day ahead prediction results. We combine the results of different iteration combinations of baseline models that forms our proposed model.

Stage-1 processing to baseline models produces two nine sub-tables based on the modification of true labels with respect to me_lstm and ti_lstm results. Table 5.32 and Table 5.33 consist of nine sub-tables showing confusion matrices built during testing process. The profit_accuracy values and the number of transactions for each cases are summarized in Table 5.34.

Table 5.32: Hybrid Model (modification based on me_lstm on extended dataset) - Confusion Matrices: 1-day ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	19	**10**	**4**
True(dec)	27	**31**	**12**
True(inc)	23	**11**	**15**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	17	**12**	**4**
True(dec)	19	**36**	**15**
True(inc)	20	**8**	**21**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	24	**6**	**3**
True(dec)	28	**36**	**6**
True(inc)	33	**2**	**14**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	31	**9**	**1**
True(dec)	27	**33**	**9**
True(inc)	27	**5**	**10**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	28	**13**	**0**
True(dec)	20	**34**	**15**
True(inc)	25	**4**	**13**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	33	**7**	**1**
True(dec)	33	**29**	**7**
True(inc)	33	**3**	**6**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	20	**10**	**3**
True(dec)	25	**38**	**10**
True(inc)	21	**11**	**14**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	16	**15**	**2**
True(dec)	19	**39**	**15**
True(inc)	20	**9**	**17**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	23	**6**	**4**
True(dec)	30	**37**	**6**
True(inc)	31	**5**	**10**

The total number of decrease and increase predictions are changing in the range $[53, 97]$. In the average, we get that value equals to **77.33** i.e., transaction ratio is $77.33/152 = 50.88\%$ for both cases. Moreover, we obtain the average profit_accuracy over nine cases are **62.94**$\%$ $\pm$ **6.13**$\%$ and **65.54**$\%$ $\pm$ **8.30**$\%$ for me_lstm and ti_lstm based modified hybrid models respectively.

5.10.2 Forecasting 3 Days Ahead

In this section, we show our proposed method's 3-days ahead prediction results. We combine the results of different iteration combinations of baseline models that forms our proposed model.

Stage-1 processing to baseline models produces two nine sub-tables based on the modification of true labels with respect to me_lstm and ti_lstm results. Table 5.35 and Table 5.36 consist of nine sub-tables showing confusion matrices built during testing process. The profit_accuracy values and the number of transactions for each cases are summarized in Table 5.37.

Table 5.33: Hybrid Model (modification based on ti_lstm on extended dataset) - Confusion Matrices: 1-day ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	26	**5**	**11**
True(dec)	22	**24**	**12**
True(inc)	21	**11**	**20**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	20	**3**	**11**
True(dec)	15	**29**	**15**
True(inc)	21	**8**	**30**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	29	**3**	**5**
True(dec)	25	**34**	**6**
True(inc)	31	**2**	**17**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	34	**0**	**8**
True(dec)	23	**26**	**9**
True(inc)	28	**5**	**19**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	26	**0**	**8**
True(dec)	18	**26**	**15**
True(inc)	29	**4**	**26**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	33	**0**	**4**
True(dec)	32	**26**	**7**
True(inc)	34	**3**	**13**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	25	**4**	**13**
True(dec)	20	**28**	**10**
True(inc)	21	**11**	**20**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	21	**2**	**11**
True(dec)	14	**30**	**15**
True(inc)	20	**9**	**30**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	31	**1**	**5**
True(dec)	23	**36**	**6**
True(inc)	30	**5**	**15**

Table 5.34: Hybrid Model (on extended dataset): 1-day ahead predictions.

Iterations		Hybrid Model - modification based on me_lstm		Hybrid Model - modification based on ti_lstm	
me_lstm	**ti_lstm**	**profit_accuracy**	**# of transactions**	**profit_accuracy**	**# of transactions**
50	50	55.42%	83/152	53.01%	83/152
50	100	59.38%	96/152	61.46%	96/152
50	150	74.63%	67/152	76.12%	67/152
100	50	64.18%	67/152	67.16%	67/152
100	100	59.49%	79/152	65.82%	79/152
100	150	66.04%	53/152	73.58%	53/152
150	50	60.47%	86/152	55.81%	86/152
150	100	57.73%	97/152	61.86%	97/152
150	150	69.12%	68/152	75.00%	68/152
Average		**62.94%**	**77.33/152**	**65.54%**	**77.33/152**

Table 5.35: Hybrid Model (modification based on me_lstm on extended dataset) - Confusion Matrices: 3-days ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	30	**0**	**7**
True(dec)	31	**17**	**3**
True(inc)	35	**12**	**17**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	32	**0**	**5**
True(dec)	40	**11**	**0**
True(inc)	38	**7**	**19**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	25	**0**	**12**
True(dec)	35	**15**	**1**
True(inc)	36	**14**	**14**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	16	**1**	**6**
True(dec)	21	**31**	**11**
True(inc)	32	**12**	**22**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	18	**1**	**4**
True(dec)	35	**22**	**6**
True(inc)	36	**6**	**24**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	13	**0**	**10**
True(dec)	31	**24**	**8**
True(inc)	34	**13**	**19**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	32	**1**	**0**
True(dec)	30	**38**	**6**
True(inc)	40	**2**	**3**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	28	**5**	**0**
True(dec)	42	**24**	**8**
True(inc)	42	**1**	**2**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	29	**4**	**0**
True(dec)	37	**26**	**11**
True(inc)	38	**3**	**4**

The total number of decrease and increase predictions are changing in the range $[40, 83]$. In the average, we get that value equals to **56.89** i.e., transaction ratio is $56.89/152 = 37.43\%$ for both cases. Moreover, we obtain the average profit_accuracy over nine cases are **65.38**% $\pm$ **8.95**% and **62.43**% $\pm$ **6.42**% for me_lstm and ti_lstm based modified hybrid models respectively.

5.10.3 Forecasting 5 Days Ahead

In this section, we show our proposed method's 5-days ahead prediction results. We combine the results of different iteration combinations of baseline models that forms our proposed model.

Stage-1 processing to baseline models produces two nine sub-tables based on the modification of true labels with respect to me_lstm and ti_lstm results. Table 5.38 and Table 5.39 consist of nine sub-tables showing confusion matrices built during testing process. The profit_accuracy values and the number of transactions for each cases are summarized in Table 5.40.

Table 5.36: Hybrid Model (modification based on ti_lstm on extended dataset) - Confusion Matrices: 3-days ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	32	**9**	**0**
True(dec)	37	**24**	**3**
True(inc)	27	**12**	**8**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	34	**7**	**0**
True(dec)	43	**16**	**0**
True(inc)	33	**7**	**12**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	28	**8**	**0**
True(dec)	38	**27**	**1**
True(inc)	30	**14**	**6**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	25	**12**	**4**
True(dec)	20	**33**	**11**
True(inc)	24	**12**	**11**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	29	**9**	**3**
True(dec)	30	**23**	**6**
True(inc)	30	**6**	**16**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	26	**7**	**3**
True(dec)	27	**31**	**8**
True(inc)	25	**13**	**12**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	36	**0**	**5**
True(dec)	25	**33**	**6**
True(inc)	41	**2**	**4**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	37	**0**	**4**
True(dec)	31	**20**	**8**
True(inc)	44	**1**	**7**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	33	**0**	**3**
True(dec)	32	**23**	**11**
True(inc)	39	**3**	**8**

Table 5.37: Hybrid Model (on extended dataset): 3-days ahead predictions.

Iterations		Hybrid Model - modification based on me_lstm		Hybrid Model - modification based on ti_lstm	
me_lstm	**ti_lstm**	**profit_accuracy**	**# of transactions**	**profit_accuracy**	**# of transactions**
50	50	60.71%	56/152	57.14%	56/152
50	100	71.43%	42/152	66.67%	42/152
50	150	51.79%	56/152	58.93%	56/152
100	50	63.86%	83/152	53.01%	83/152
100	100	73.02%	63/152	61.90%	63/152
100	150	58.11%	74/152	58.11%	74/152
150	50	82.00%	50/152	74.00%	50/152
150	100	65.00%	40/152	67.50%	40/152
150	150	62.50%	48/152	64.58%	48/152
Average		**65.38%**	**56.89/152**	**62.43%**	**56.89/152**

Table 5.38: Hybrid Model (modification based on me_lstm on extended dataset) - Confusion Matrices: 5-days ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	28	**2**	**0**
True(dec)	26	**39**	**6**
True(inc)	39	**4**	**8**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	23	**7**	**0**
True(dec)	33	**32**	**6**
True(inc)	35	**7**	**9**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	27	**2**	**1**
True(dec)	29	**29**	**13**
True(inc)	35	**3**	**13**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	24	**2**	**1**
True(dec)	25	**43**	**4**
True(inc)	36	**8**	**9**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	18	**8**	**1**
True(dec)	34	**34**	**4**
True(inc)	30	**13**	**10**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	22	**4**	**1**
True(dec)	27	**32**	**13**
True(inc)	33	**5**	**15**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	24	**2**	**2**
True(dec)	24	**42**	**5**
True(inc)	40	**4**	**9**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	17	**9**	**2**
True(dec)	34	**33**	**4**
True(inc)	33	**6**	**14**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	22	**4**	**2**
True(dec)	26	**31**	**14**
True(inc)	35	**5**	**13**

The total number of decrease and increase predictions are close each other in the range $[59, 70]$. In the average, we get that value equals to **65.44** i.e., transaction ratio is $65.44/152 = 43.06\%$ for both cases. Moreover, we obtain the average profit_accuracy over nine cases are **70.66**% $\pm$ **6.61**% and **66.50**% $\pm$ **5.74**% for me_lstm and ti_lstm based modified hybrid models respectively.

5.11 Summary

Overall, we summarize all experiments' result in Table 5.41. In one day ahead predictions, we observed that individual models have slightly better profit_accuracy than me_ti_lstm by 0.91%, but less transaction number ratio by 3.91% on average. Moreover, when we combine the predictions of individual models, our proposed model reaches the profit_accuracy **73.09%** (22.30% improvement) on average by reducing transaction number ratio by 35.73%.

In three days ahead predictions, we observed that individual models have better

Table 5.39: Hybrid Model (modification based on ti_lstm on extended dataset) - Confusion Matrices: 5-days ahead. `Iteration1` and `Iteration2` corresponds to iteration number of me_lstm and ti_lstm respectively.

Iteration1=50 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	31	**2**	**5**
True(dec)	23	**34**	**6**
True(inc)	39	**4**	**8**

Iteration1=50 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	30	**1**	**9**
True(dec)	25	**23**	**6**
True(inc)	36	**7**	**15**

Iteration1=50 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	34	**1**	**7**
True(dec)	22	**23**	**13**
True(inc)	35	**3**	**14**

Iteration1=100 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	32	**2**	**4**
True(dec)	19	**40**	**4**
True(inc)	34	**8**	**9**

Iteration1=100 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	30	**1**	**9**
True(dec)	24	**26**	**4**
True(inc)	28	**13**	**17**

Iteration1=100 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	34	**2**	**6**
True(dec)	18	**27**	**13**
True(inc)	30	**5**	**17**

Iteration1=150 Iteration2=50			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	31	**3**	**4**
True(dec)	18	**40**	**5**
True(inc)	39	**4**	**8**

Iteration1=150 Iteration2=100			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	28	**3**	**9**
True(dec)	24	**26**	**4**
True(inc)	32	**6**	**20**

Iteration1=150 Iteration2=150			
	Pred(no_act)	Pred(dec)	Pred(inc)
True(no_act)	33	**3**	**6**
True(dec)	17	**27**	**14**
True(inc)	33	**5**	**14**

Table 5.40: Hybrid Model (on extended dataset): 5-days ahead predictions.

Iterations		Hybrid Model - modification based on me_lstm		Hybrid Model - modification based on ti_lstm	
me_lstm	**ti_lstm**	**profit_accuracy**	**# of transactions**	**profit_accuracy**	**# of transactions**
50	50	79.66%	59/152	71.19%	59/152
50	100	67.21%	61/152	67.21%	61/152
50	150	68.85%	61/152	60.66%	61/152
100	50	77.61%	67/152	73.13%	67/152
100	100	62.86%	70/152	61.43%	70/152
100	150	67.14%	70/152	62.86%	70/152
150	50	79.69%	64/152	75.00%	64/152
150	100	69.12%	68/152	67.65%	68/152
150	150	63.77%	69/152	59.42%	69/152
Average		**70.66%**	**65.44/152**	**66.50%**	**65.44/152**

profit_accuracy than me_ti_lstm by 5.81%, but less transaction number ratio by 7.20% on average. Moreover, the hybrid model presents a performance of **68.31%** (19.29% improvement) on average by reducing transaction number ratio by 35.83%.

In five days ahead predictions, we observed that individual models have slightly better profit_accuracy than me_ti_lstm by 0.54% and high transaction number ratio by 21.48% on average. Moreover, the hybrid model presents an exceptional performance of **79.42%** (34.33% improvement) by reducing transaction number by 32.72%.

The more analysis about the results is listed in the following:

- ME_LSTM: The profit_accuracy of 3-days ahead predictions is slightly better than 1-day ahead predictions (by just 0.40%). Both are 5.48% and 5.08% higher than 5-days ahead predictions respectively. The transaction number ratio is getting higher when we forecast the further and 87.52% on average,
- TI_LSTM: The profit_accuracy is decreasing when we extend the prediction period and in the range [45.11% - 51.43%]. The transaction number ratio is the same for 1-day and 5-days ahead predictions and higher than 3-days ahead predictions by 6.03%,
- ME_TI_LSTM: The profit_accuracy of 1-day ahead predictions is the highest with 49.89%. Additionally, the profit_accuracy of 5-days ahead prediction is 1.34% higher than 3-days ahead predictions. The transaction number ratio over test data is changing and 75.44% on average,
- Hybrid Model: Interestingly, the performance of profit_accuracy is highest in 5-days ahead predictions. Additionally, 1-day ahead predictions is 5.17% higher than 3-days predictions. The transaction number ratio over test data is changing and close to each other (40.37% on average).

When we compare the performances of hybrid model on main dataset and extended dataset, we see that there are decreases on the profit_accuracy and changing values on the transaction number ratio. We represent this comparison in Table 5.42 for different periods. Initial observation is that the behavior of hybrid model on extended dataset is the same with the main dataset. In other words, the best performance occurs on

Table 5.41: Summary of all experiments conducted on main dataset.

	1-day ahead		3-days ahead		5-days ahead	
	profit_accuracy	**# of transactions**	**profit_accuracy**	**# of transactions**	**profit_accuracy**	**# of transactions**
ME_LSTM	50.16%	74.90%	50.56%	90.54%	45.08%	97.11%
TI_LSTM	51.43%	72.29%	47.49%	66.26%	45.11%	72.29%
ME_TI_LSTM	49.89%	77.50%	43.22%	85.60%	44.56%	63.22%
Hybrid LSTM	**73.09%**	37.86%	**68.31%**	42.57%	**79.42%**	40.68%

Table 5.42: Performance Comparison of Hybrid Model.

	1-day ahead		3-days ahead		5-days ahead	
	profit_accuracy	**# of transactions**	**profit_accuracy**	**# of transactions**	**profit_accuracy**	**# of transactions**
Main Dataset	73.09%	37.86%	68.31%	42.57%	79.42%	40.68%
Extended Dataset	64.24%	50.88%	63.91%	37.43%	68.58%	43.05%

5-days ahead predictions and 1-day ahead predictions is slightly higher by 0.33% 3-days ahead predictions.

In 1-day ahead predictions, while there is 8.85% decrease on the profit_accuracy, 13.02% increase on the transaction number ratio. In 3-days ahead predictions, there is 4.4% and 5.15% decreases on the profit_accuracy and transaction number ratio respectively. Finally, while there is 10.84% decrease on the profit_accuracy, 2.37% increase on the transaction number ratio in 5-days ahead predictions.

CHAPTER 6

CONCLUSION

In this thesis, we applied two separate LSTM models to forecast directional movement of EUR/USD currency. These predictions cover periods of 1-day, 3-days and 5-days ahead. Classification is based on three classes which are no_action, $decrease$ and $increase$. No_action means the changes remaining between certain thresholds are negligible and requires no action at all. This enables us to define new performance metric $profit\_accuracy$ that gives us the ratio of number of profitable transactions over the total transactions. We simply define profitable transaction as right prediction of decrease and increase classes. Predicting the right direction of currency pair gives us a chance to take profit from those transactions. This is the main objective of our study and this metric meets our expectation entirely since predicted class no_action have no contribution on the profit/loss of the transaction. The proposed algorithms are tested in EUR/USD currency pair for different periods.

Based on the analysis of the experiments, we present the general outlines of our study:

- Applying labeling algorithm makes classes evenly distributed over the dataset. This prevent us form dealing with class imbalance problem,
- ME_LSTM has the highest performance in terms of transaction number ratio among baseline models (87.52% on average),
- ME_LSTM has the highest profit_accuracy among baseline models (48.60% on average),
- ME_LSTM has performed biased attitude towards no_action and $decrease$ classes,

- TI_LSTM has the worst transaction number ratio among baseline models (70.28% on average),
- TI_LSTM is in the middle in terms of profit_accuracy among baseline models (48.01% on average),
- TI_LSTM has good performance in terms of distribution of predicted classes,
- ME_TI_LSTM is in the middle for transaction number ratio among baseline models (75.44% on average),
- ME_TI_LSTM has the worst profit_accuracy among baseline models (45.89% on average),
- ME_TI_LSTM has performed biased attitude towards no_action and $decrease$ classes (similar to ME_LSTM),
- Hybrid model has the highest performance in terms of profit_accuracy for predictions of all periods significantly (**73.61%** on average which is tempting),
- Hybrid model sacrifices the transaction numbers in a huge amount compared to baseline models (**40.37%** on average),
- The most important finding is that hybrid model is not affected by biased predictions of ME_LSTM and maintains a well distributed predictions over test data by preserving and raising profit_accuracy.
- Hybrid model exhibits a performance of **66.58%** on average on extended dataset (with **43.79%** transaction number ratio on average) which is still a promising result despite of the decrease on the profit_accuracy compared to main dataset.

This approach discloses the fact that combination of macroeconomic factors and technical indicators can be used together, yet separately, to form a hybrid model in order to forecast directional movement of currency pairs in Forex environment.

This study exposes possible future directions that are listed below:

- Extension can be made for other curreny pairs such as EUR/GBP, GBP/USD, USD/CHF, GBP/CHF and EUR/CHF,

- In order to validate the model, a trading simulator can be developed. Such a simulator can be useful to observe real time behavior of our model,
- The set of technical indicators and macroeconmic variables can be augmented such as average true range (ATR), money flow index (MFI), %B indicator or other other stock market indexes, GDP prices of previous year, unemployment rate, etc,
- Classifier ensembles or other learning techniques such as SVM or MLP can be used to learn the directional movement of currency pairs,

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short term memory," *Neural Computation 9*, 1997.

[2] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, 1999.

[3] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *Neural Networks and Learning Systems*, 2017.

[4] "`http://www.metatrader5.com`," Accessed: March 2019.

[5] M. Qiu and Y. Song, "Predicting the direction of stock market index movement using an optimized artificial neural network model," *Plos One*, 2016.

[6] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[7] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, 2014.

[8] L. C. Francis E.H. Tay and, "Application of support vector machines in financial time series forecasting," *The International Journal of Management Science*, 2001.

[9] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications*, 2011.

[10] B. Weng, L. Lu, X. Wang, F. M. Megahed, and W. Martinez, "Predicting short-term stock prices using ensemble methods and online data sources," *Expert Systems With Applications*, 2018.

[11] W. Huanga, Y. Nakamori, and S.-Y. Wang, “Forecasting stock market movement direction with support vector machine,” *Computers & Operations Research*, 2004.

[12] Y. Kara, M. A. Boyacioglu, and Ömer Kaan Baykan, “Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange,” *Expeert System and Applications*, 2010.

[13] S. Galeshchuk and S. Mukherjee, “Deep networks for predicting direction of change in foreign exchange rates,” *Intelligent Systems in Accounting Finance & Management*, 2017.

[14] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using trend deterministic data preparation and machine learning technique,” *Expert Systems with Applications*, 2015.

[15] M. Ballings, D. V. den Poel, N. Hespeels, and R. Gryp, “Evaluating multiple classifiers for stock price direction prediction,” *Expert Systems with Applications*, 2015.

[16] Q. Gao, “Stock market forecating using recurrent neural network,” *MSc Thesis, Faculty of Graduate School, University of Missouri, Columbia, USA*, 2016.

[17] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, “Stock market’s price movement prediction with lstm neural networks,” *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[18] X. Zhong and D. Enke, “Forecasting daily stock market return using dimensionality reduction,” *Expert Systems with Applications*, 2017.

[19] H. Hu, L. Tang, S. Zhang, and H. Wang, “Predicting the direction of stock markets using optimized neural networks with google trends,” *Neurocomputing*, 2018.

[20] K. jae Kim, “Financial time series forecasting using support vector machines,” *Neurocomputing*, 2003.

[21] M. D. Archer, “Getting started in currency trading,” *Wiley*, 2010.

[22] M. Ozturk, “Heuristic based trading system on forex data using technical indicator rules,” *MSc Thesis, Graduate School of Natural and Applied Sciences, METU, Ankara, Turkey*, 2015.

[23] “`https://www.investopedia.com/categories/forex.asp`,” Accessed: April 2019.

[24] M. O. Özorhan, “Forecasting direction of exchange rate fluctuations with two dimensional pattern and currency strength,” *PhD Thesis, Graduate School of Natural and Applied Sciences, METU, Ankara, Turkey*, 2017.

[25] J. J. Murphy, “Technical anaylsis of the financial markets,” *New York Institute of Finance*, 1999.

[26] A. Kritzer, “Forex for beginners: A comprehensive guide to profiting from the global currency markets,” *Apress*, 2012.

[27] A. Graves, “Supervised sequence labelling with recurrent neural networks,” *Studies in Computational Intelligence*, 2012.

[28] F. A. Gers and J. Schmidhuber, “Recurrent nets that time and count,” *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on, Volume: 3*, 2000.

[29] M. L. Alex Graves, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 31, Issue: 5)*, 2009.

[30] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, “Dropout improves recurrent neural networks for handwriting recognition,” *arXiv:1312.4569 [cs.CV], https://arxiv.org/abs/1312.4569*, 2014.

[31] A. Graves, “Generating sequences with recurrent neural networks,” *arXiv:1308.0850 [cs.NE], http://arxiv.org/abs/1308.0850*, 2014.

[32] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *arXiv:1409.2329 [cs.NE], http://arxiv.org/abs/1409.2329*, 2015.

[33] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," *arXiv:1410.8206 [cs.CL], https://arxiv.org/abs/1410.8206*, 2015.

[34] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *The Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

[35] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," *The Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

[36] S. K. Sønderby and O. Winther, "Protein secondary structure prediction with long short term memory networks," *arXiv:1412.7828 [q-bio.QM], https://arxiv.org/abs/1412.7828*, 2015.

[37] E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller, "Multi-resolution linear prediction based features for audio onset detection with bidirectional lstm neural networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[38] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *arXiv:1411.4389 [cs.CV], https://arxiv.org/abs/1411.4389*, 2016.

[39] "`https://stockcharts.com/school/doku.php?id=chart_school:technical_indicators`," Accessed: April 2019.

[40] G. Appel, "Technical analysis: Power tools for active investors," *Financial Times Prentice Hall*, 2005.

[41] R. W. Colby, "The encyclopedia of technical market indicators," *McGraw-Hill*, 2003.

[42] J. W. Wilder, "New concepts in technical trading systems," *Trend Research*, 1978.

[43] J. Bollinger, "Bollinger on bollinger bands," *McGraw-Hill*, 2002.

[44] M. Ozturk, . H. Toroslu, and G. Fidan, "Heuristic based trading system on forex data using technical indicator rules," *Applied Soft Computing*, 2016.

[45] D. R. Lambert, "Commodity channel index: Tool for trading cyclic trends," *Technical Analysis of Stocks & Commodities 1*, 1983.

[46] "`https://www.investopedia.com/terms/i/interestrate.asp`," Accessed: April 2019.

[47] "`https://www.investing.com/currencies/eur-usd-historical-data`," Accessed: November 2018.

[48] "`https://sdw.ecb.europa.eu`," Accessed: November 2018.

[49] "`https://data.bls.gov/`," Accessed: November 2018.

[50] "`https://fred.stlouisfed.org/series/DFF`," Accessed: November 2018.

[51] "`https://finance.yahoo.com/`," Accessed: November 2018.

[52] "`https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html`," Accessed: April 2019.

[53] "`https://keras.io`," Accessed: April 2019.