

Yıldırım, Deniz Can; Toroslu, Ismail Hakkı; Fiore, Ugo

Article

Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators

Financial Innovation

Provided in Cooperation with:

Springer Nature

Suggested Citation: Yıldırım, Deniz Can; Toroslu, Ismail Hakkı; Fiore, Ugo (2021) : Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators, Financial Innovation, ISSN 2199-4730, Springer, Heidelberg, Vol. 7, Iss. 1, pp. 1-36, <https://doi.org/10.1186/s40854-020-00220-2>

This Version is available at:

<https://hdl.handle.net/10419/237234>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.




<https://creativecommons.org/licenses/by/4.0/>

RESEARCH

Open Access



Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators

Deniz Can Yıldırım¹, İsmail Hakkı Toroslu^{1*}  and Ugo Fiore²

*Correspondence:

toroslu@ceng.metu.edu.tr

¹ Department of Computer Engineering, Middle East Technical University, 06800 Ankara, Turkey

Full list of author information is available at the end of the article

Abstract

Forex (foreign exchange) is a special financial market that entails both high risks and high profit opportunities for traders. It is also a very simple market since traders can profit by just predicting the direction of the exchange rate between two currencies. However, incorrect predictions in Forex may cause much higher losses than in other typical financial markets. The direction prediction requirement makes the problem quite different from other typical time-series forecasting problems. In this work, we used a popular deep learning tool called “long short-term memory” (LSTM), which has been shown to be very effective in many time-series forecasting problems, to make direction predictions in Forex. We utilized two different data sets—namely, macroeconomic data and technical indicator data—since in the financial world, fundamental and technical analysis are two main techniques, and they use those two data sets, respectively. Our proposed hybrid model, which combines two separate LSTMs corresponding to these two data sets, was found to be quite successful in experiments using real data.

Keywords: Time series, Forex, Directional movement forecasting, Technical and macroeconomic indicators, LSTM

Introduction

The foreign exchange market, known as Forex or FX, is a financial market where currencies are bought and sold simultaneously. Forex is the world’s largest financial market, with a volume of more than \$5 trillion. It is a decentralized market that operates 24 h a day, except for weekends, which makes it quite different from other financial markets.

The characteristics of Forex show differences compared to other markets. These differences can bring advantages to Forex traders for more profitable trading opportunities. Some of these advantages include no commissions, no middlemen, no fixed lot size, low transaction costs, high liquidity, almost instantaneous transactions, low margins/high leverage, 24-h operations, no insider trading, limited regulation, and online trading opportunities. Two types of techniques are used to predict future values for typical financial time series—fundamental analysis and technical analysis—and both can be

used for Forex. The former uses macroeconomic factors while the latter uses historical data to forecast the future price or the direction of the price.

The main decision in Forex involves forecasting the directional movement between two currencies. Traders can profit from transactions with correct directional prediction and lose with incorrect prediction. Therefore, identifying directional movement is the problem addressed in this study.

We chose the Euro/US dollar (EUR/USD) pair for the analysis since it is the largest traded Forex currency pair in the world, accounting for more than 80% of the total Forex volume.

In recent years, deep learning tools, such as long short-term memory (LSTM), have become popular and have been found to be effective for many time-series forecasting problems. In general, such problems focus on determining the future values of time-series data with high accuracy. However, in direction prediction problems, accuracy cannot be defined as simply the difference between actual and predicted values. Therefore, a novel rule-based decision layer needs to be added after obtaining predictions from LSTMs.

In this work, we propose a hybrid model composed of a macroeconomic LSTM model and a technical LSTM model, named after the types of data they use. We first separately investigated the effects of these data on directional movement. After that, we combined the results to significantly improve prediction accuracy. The macroeconomic LSTM model utilizes several financial factors, including interest rates, Federal Reserve (FED) funds rate, inflation rates, Standard and Poor's (S&P) 500, and Deutscher Aktien Index (DAX) market indexes. Each factor has important effects on the trend of the EUR/USD currency pair. This can be interpreted as a fundamental analysis of price data. The other model is the technical LSTM model, which takes advantage of technical analysis. Technical analysis is based on technical indicators that are mathematical functions used to predict future price action. The feature set in our model uses popular technical indicators such as moving average (MA), moving average convergence divergence (MACD), rate of change (ROC), momentum, relative strength index (RSI), Bollinger bands (BB), and the commodity channel index (CCI).

The contributions of this study are as follows:

- A popular deep learning tool called LSTM, which is frequently used to forecast values in time-series data, is adopted to predict direction in Forex data.
- Both macroeconomic and technical indicators are used as features to make predictions.
- A novel hybrid model is proposed that combines two different models with smart decision rules to increase decision accuracy by eliminating transactions with weaker confidence.
- The proposed model and baseline models are tested using recent real data to demonstrate that the proposed hybrid model outperforms the others.

The rest of this paper is organized as follows. In “[Related work](#)” section, related studies of the financial time-series prediction problem are thoroughly examined. “[Forex preliminaries](#)”–“[Technical indicators](#)” sections provide background information about

Forex, LSTM, and the technical indicators. Then, “The data set” section presents the data set used in the experiments. “LSTM-based hybrid model using macroeconomic and technical indicators” section introduces the proposed algorithm to handle the directional movement prediction problem. Moreover, the preprocessing and postprocessing phases are also explained in detail. “Experiments” section presents the results of the experiments and the classification performances of the proposed model. “Discussion” and “Conclusion” sections discuss the experimental results and provide insight for future research directions.

Related work

Various forecasting methods have been considered in the finance domain, including machine learning approaches (e.g., support vector machines and neural networks) and new methods such as deep learning. Unfortunately, there are not many survey papers on these methods. Cavalcante et al. (2016), Bahrammirzaee (2010), and Saad and Wunsch (1998) have provided overviews of the field. The most recent of these, by Cavalcante et al. (2016), categorized the approaches used in different financial markets. Although that study mainly introduced methods proposed for the stock market, it also discussed applications for foreign exchange markets.

There has been a great deal of work on predicting future values in stock markets using various machine learning methods. We discuss some of them below.

Selvamuthu et al. (2019) used neural networks based on Levenberg–Marquardt, scaled conjugate gradient, and Bayesian regularization for stock market prediction based on tick data and 15-min-interval data for an Indian company.

Patel et al. (2015b) developed a two-stage fusion structure to predict the future values of the stock market index for 1–10, 15, and 30 days using 10 technical indicators. In the first stage, support vector machine regression (SVR) was applied to these inputs, and the results were fed into an artificial neural network (ANN). SVR and random forest (RF) models were used in the second stage. They compared the fusion model with standalone ANN, SVR, and RF models. They reported that the fusion model significantly improved upon the standalone models.

Guresen et al. (2011) explored several ANN models for predicting stock market indexes. These models include multilayer perceptron (MLP), dynamic artificial neural network (DAN2), and hybrid neural networks with generalized autoregressive conditional heteroscedasticity (GARCH). Applying mean-square error (MSE) and mean absolute deviation (MAD), their results showed that MLP performed slightly better than DAN2 and GARCH-MLP while GARCH-DAN2 had the worst results.

Weng et al. (2018) developed a financial expert system using ensemble methods (i.e., neural network regressing ensemble (NNRE), support vector regression ensemble (SVRE), boosted regression tree (BRT), and random forest regression (RFR)) to predict stock prices 1 day ahead. Market prices, technical indicators, financial news, Google Trends, and the number unique visitors to Wikipedia pages were used as inputs. They also investigated the effect of PCA on performance. They reported that ensembles with PCA performed better than those without PCA. They also noted that BRT and RFR were the best while SVRE was the worst in terms of mean absolute percentage error.

Huang et al. (2005) examined forecasting weekly stock market movement direction using SVM. They compared SVM with linear discriminant analysis, quadratic discriminant analysis, and Elman back-propagation neural networks. They also proposed a model that combined SVM with other classifiers. They used not only the NIKKEI 225 index but also macroeconomic variables as features for the model. Their direction calculation was based on the first-order difference natural logarithmic transformation, and the directions were either increasing or decreasing. SVM outperformed the other models with an accuracy of 73% while the combined model was the best, with an accuracy of 75%.

Kara et al. (2011) compared the performance of ANN and SVM for predicting the direction of stock price index movement. Ten technical indicators were used as inputs for the model. They found that ANN, with an accuracy of 75.74%, performed significantly better than SVM, which had an accuracy of 71.52%.

Patel et al. (2015a) compared the performance of four classifiers (ANN, SVM, random forest, and naive Bayes) for stock price index direction using two approaches. In the first approach, they used 10 technical indicator values as inputs with different parameter settings for classifiers. Prediction accuracy fell within the range of 0.7331–0.8359. In the other approach, they represented same 10 technical indicator results as directions (up and down), which were used as inputs for the classifiers. Using this approach, they enhanced accuracy by about 15% for all of the classifiers. Although their experiments concerned short-term prediction, the direction period was not explicitly explained.

Ballings et al. (2015) evaluated ensemble methods (random forest, AdaBoost, and kernel factory) against neural networks, logistic regression, SVM, and k-nearest neighbor for predicting 1 year ahead. They used different stock market domains in their experiments. According to the median area under curve (AUC) scores, random forest showed the best performance, followed by SVM, random forest, and kernel factory.

Hu et al. (2018) introduced an improved sine–cosine algorithm (ISCA) for optimizing the weights and biases of BPNN to predict the directions of open stock prices of the S&P 500 and Dow Jones Industrial Average indices. Using Google Trends data in addition to the opening, high, low, and closing price, as well as trading volume, in their experiments, they obtained an 86.81% hit ratio for the S&P 500 index and an 88.98% hit ratio for the Dow Jones Industrial Average Index.

Gui et al. (2015) investigated SVM for predicting stock price index direction with different parameter settings. That study also compared the result for SVM with BPNN and case-based reasoning models; multiple technical indicators were used as inputs for the models. That study found that SVM outperformed the other models with an accuracy of 57.8313% while the other models had accuracies of 54.7332% and 51.9793%, respectively.

Qiu and Song (2016) developed a genetic algorithm (GA)—based optimized ANN to predict the direction of the next day's price in the stock market index. GA was used to optimize the initial weights and bias of the model. Two types of input sets were generated using several technical indicators of the daily price of the Nikkei 225 index and fed into the model. They obtained accuracies 60.87% for the first set and 81.27% for the second set.

Zhong and Enke (2017) investigated three-dimensional reduction techniques applied to ANN for forecasting the daily direction of the S&P 500 Index ETF (SPY). Principal

component analysis (PCA), fuzzy robust principal component analysis (FRPCA), and kernel-based principal component analysis (KPCA) were used to reduce the number of features. Their experiments indicated that ANN with PCA performed slightly better than the other two techniques.

Zhong and Enke (2019) used deep neural networks and ANNs to forecast the daily return direction of the stock market. They performed experiments on both untransformed and PCA-transformed data sets to validate the model.

In addition to classical machine learning methods, researchers have recently started to use deep learning methods to predict future stock market values. LSTM has emerged as a deep learning tool for application to time-series data, such as financial data.

Zhang et al. (2017) proposed a state-frequency memory recurrent network, which is a modification of LSTM, to forecast stock prices. By decomposing the hidden states of memory cells into multiple frequency components, they could learn the trading patterns of those frequencies. They used state-frequency components to predict future price values through nonlinear regression. They used stock prices from several sectors and performed experiments to make forecasts for 1, 3, and 5 days. They compared the results with LSTM and autoregressive integrated moving average (ARIMA) in terms of mean-square error. They obtained errors of 5.57, 17.00, and 28.90 for the different steps, which outperformed the other models.

Fulfillment et al. (2016) studied stock market forecasting in six different domains using LSTM. He aimed to predict the next 3 h using hourly historical stock data. The model was trained to classify three classes—namely, increasing 0–1%, increasing above 1%, and not increasing (less than 0%). The accuracy results ranged from 49.75 to 59.5%. That study also built a stock trading simulator to test the model on real-world stock trading activity. With that simulator, he managed to make profit in all six stock domains with an average of 6.89%.

Nelson et al. (2017) examined LSTM for predicting 15-min trends in stock prices using technical indicators. They used 175 technical indicators (i.e., external technical analysis library) and the open, close, minimum, maximum, and volume as inputs for the model. They compared their model with a baseline consisting of multilayer perceptron, random forest, and pseudo-random models. The accuracy of LSTM for different stocks ranged from 53 to 55.9%. They concluded that LSTM performed significantly better than the baseline models, according to the Kruskal–Wallis test.

More recently, Fischer and Krauss (2018) applied LSTM to the stock market. They investigated many different aspects of the stock market and found that LSTM was very successful for predicting future prices for that type of time-series data. They also compared LSTM with more traditional machine learning tools to show its superior performance.

Similarly, Di Persio and Honchar (2016) applied LSTM and two other traditional neural network based machine learning tools to future price prediction. They also analyzed ensemble-based solutions by combining results obtained using different tools.

In addition to traditional exchanges, many studies have also investigated Forex. Some studies of Forex based on traditional machine learning tools are discussed below.

Galeshchuk and Mukherjee (2017) investigated the performance of a convolutional neural network (CNN) for predicting the direction of change in Forex. Using the daily

closing rates of EUR/USD, GBP/USD, and USD/JPY, they compared the results of CNN with their baseline models and SVM. While the baseline models and SVM had an accuracy of around 65%, their proposed CNN model had an accuracy of about 75%.

Meanwhile, Kayal (2010) investigated the use of MLP in Forex. That work used basic technical indicators as inputs.

Ghazali et al. (2009) also investigated the use of neural networks for Forex. They proposed a higher-order neural network called a dynamic ridge polynomial neural network (DRPNN). In their experiments, DRPNN performed better than a ridge polynomial neural network (RPNN) and a pi-sigma neural network (PSNN).

To predict exchange rates, Majhi et al. (2009) proposed using new ANNs, referred to as a functional link artificial neural network (FLANN) and a cascaded functional link artificial neural network (CFLANN). They demonstrated that those new networks were more robust and had lower computational costs compared to an MLP trained with back-propagation.

In what is commonly called a *mark-to-market* approach, market prices are increasingly being used to calibrate models to quantify risk in several sectors. The net present value of a financial institution, for example, is an important input for estimating both bankruptcy risk (e.g., Kou et al. 2020) and the likelihood that shocks will propagate throughout the financial system (Kou et al. 2019). In such a context, stock price crashes not only dramatically damage the capital market but also have medium-term adverse effects on the financial sector as a whole (Wen et al. 2019). Credit risk is a major factor in financial shocks. Therefore, a realistic appraisal of solvency needs to be an objective for banks. At the level of the individual borrower, credit scoring is a field in which machine learning methods have been used for a long time (e.g., Shen et al. 2020; Wang et al. 2020).

Deep learning methods such as LSTM are rarely used for Forex. In one recent work, Shen et al. (2015) proposed a modified deep belief network. They were able to show that deep learning approaches outperformed traditional methods.

Even though LSTM is starting to be used in financial markets, using it in Forex for direction forecasting between two currencies, as proposed in the present work, is a novel approach.

Forex preliminaries

Forex has characteristics that are quite different from those of other financial markets (Archer 2010; Ozorhan et al. 2017). To explain Forex, we start by describing how a trade is made. Profit/loss calculations are made using the difference between the final ratio and the initial ratio of the currency pair that has been traded. If the ratio of the currency pair increases and the trader goes long, or the currency pair ratio decreases and the trader goes short, the trader will profit from that transaction when it is closed. Otherwise, the trader not profit. For example, let us assume the EUR/USD ratio was 1.1500 when the trader started a transaction, going long with an initial amount of \$10,000. When the position closes (i.e., the transaction ends) with a ratio of 1.1550, the trader will gain $10000 * (1.1550 - 1.1500) = \50 . When the position closes with a ratio of 1.1450, the trader will lose $10000 * (1.1500 - 1.1450) = \50 . Furthermore, these calculations are based on no leverage. If the trader uses a leverage value such as 10, both the loss and the gain are multiplied by 10.

Detailed definitions of commonly used concepts and terms in Forex can be found in Forex (2018), Archer (2010) and Özorhan (2017). Here, we explain only the most important ones.

Base currency, which is also called the transaction currency, is the first currency in the currency pair while quote currency is the second one in the pair. To illustrate, in the EUR/USD pair, EUR is the base currency, and USD is the quote currency.

Being long (or going long) means buying the base currency or selling the quote currency in the currency pair. Being short (or going short) means selling the base currency or buying the quote currency in the currency pair. Pip is an abbreviation for “percentage of point,” defined as the smallest amount of change occurring in the currency ratio. In general, pip corresponds to the fourth decimal point (i.e., minimum as 0.0001) of that currency. Pipette is the fractional pip, which corresponds to the fifth decimal point (i.e., as 0.00001). In other words, 1 pip equals 10 pipettes.

Leverage corresponds to the use of borrowed money when making transactions. A leverage of 1:100 indicates that if one opens a position with a volume of 1, the actual transaction volume will be 100. After using leverage, one can either gain or lose 100 times the amount of that volume. Margin refers to money borrowed by a trader that is supplied by a broker to make investments using leverage. In this way, one can multiply his/her gains or losses.

Bid price is the price at which the trader can sell the base currency. Ask price is the price at which the trader can buy the base currency. Spread is the difference between the ask and bid prices. A lower spread means the trader can profit from small price changes. Spread value is dependent on market volatility and liquidity. Stop loss is an order to sell a currency when it reaches a specified price. This order is used to prevent larger losses for the trader. Take profit is an order by the trader to close the open position (transaction) for a gain when the price reaches a predefined value. This order guarantees profit for the trader without having to worry about changes in the market price. Market order is an order that is performed instantly at the current price. Swap is a simultaneous buy and sell action for the currency at the same amount at a forward exchange rate. This protects traders from fluctuations in the interest rates of the base and quote currencies. If the base currency has a higher interest rate and the quote currency has a lower interest rate, then a positive swap will occur; in the reverse case, a negative swap will occur.

Fundamental analysis and technical analysis are the two techniques commonly used for predicting future prices in Forex. While the first is based on economic factors, the latter is related to price actions (Archer 2010).

Fundamental analysis focuses on the economic, social, and political factors that can cause prices to move higher, move lower, or stay the same (Archer 2010; Murphy 1999). These factors are also called macroeconomic factors. Economic data reports, interest rates, monetary policy, and international trade/investment flows are some examples (Özorhan et al. 2017).

Technical analysis uses only the price to predict future price movements (Kritzer and Service 2012). This approach studies the effect of price movement. Technical analysis mainly uses open, high, low, close, and volume data to predict market direction or generate sell and buy signals (Archer 2010). It is based on the following three assumptions (Murphy 1999):

- Market action discounts everything.
- Price moves in trends.
- History repeats itself.

Chart analysis and price analysis using technical indicators are the two main approaches in technical analysis. While the former is used to detect patterns in price charts, the latter is used to predict future price actions (Ozorhan et al. 2017).

Long short-term memory (LSTM)

Long short-term memory (LSTM) was proposed by Hochreiter and Schmidhuber (1997). LSTM is a recurrent neural network architecture that was designed to overcome the vanishing gradient problem found in conventional recurrent neural networks (RNNs) (Biehl 2005). Errors between layers tend to vanish or blow up, which causes oscillating weights or unacceptably long convergence times. The initial LSTM structure solves this problem by introducing the constant error carousel (CEC). In this way, the architecture ensures constant error flow between the self-connected units (Hochreiter and Schmidhuber 1997).

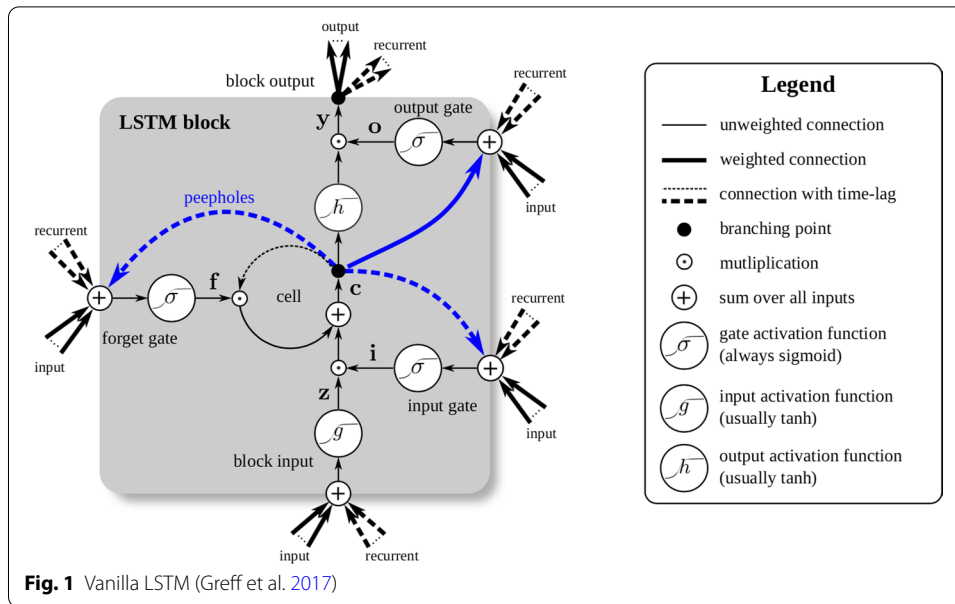
The memory cell of the initial LSTM structure consists of an input gate and an output gate. While the input gate decides which information should be kept or updated in the memory cell, the output gate controls which information should be output. This standard LSTM was extended with the introduction of a new feature called the forget gate (Gers et al. 2000). The forget gate is responsible for resetting a memory state that contains outdated information. Furthermore, peephole connections and full back-propagation through time (BPTT) training are final features that were added to the LSTM architecture (Gers and Schmidhuber 2000; Greff et al. 2017). With these modifications, the architecture was renamed Vanilla LSTM (Greff et al. 2017), as shown in Fig. 1.

LSTM offers an effective and scalable model for learning problems that includes sequential data (Greff et al. 2017). It has been used in many different fields, including handwriting recognition (Graves et al. 2009; Pham et al. 2014) and generation (Graves 2013), language modeling (Zaremba et al. 2014) and translation (Luong et al. 2015), acoustic modeling of speech (Zia and Zahid 2019), speech synthesis (Fan et al. 2014), protein secondary structure prediction (Sønderby and Winther 2014), audio analysis (Marchi et al. 2014), and video data analysis (Donahue et al. 2017; Greff et al. 2017).

Forward pass

One of the two main operations of LSTM, shown in Fig. 1, is called the forward pass. In the forward pass, the calculation moves forward by updating the weights (Greff et al. 2017). The weights of LSTM can be categorized as follows:

- Input weights: $W_z, W_i, W_f, W_o \in \mathbb{R}^{N \times M}$
- Recurrent weights: $R_z, R_i, R_f, R_o \in \mathbb{R}^{N \times N}$
- Peephole weights: $p_i, p_f, p_o \in \mathbb{R}^N$



- Bias weights: $b_z, b_i, b_f, b_o \in \mathbb{R}^N$,

where z is the block input, i is the input gate, f is the forget gate, o is the output gate, N is the number of LSTM blocks, and M is the number of inputs. By introducing x^t as the input vector, y^t as the block output, and c^t as the cell at time t , the formulation of the forward pass in Vanilla LSTM can be defined as below:

$$\bar{z}^t = W_z x^t + R_z y^{t-1} + b_z, \tag{1}$$

$$z^t = g(\bar{z}^t), \tag{2}$$

$$\bar{i}^t = W_i x^t + R_i y^{t-1} + p_i \odot c^{t-1} + b_i, \tag{3}$$

$$i^t = \sigma(\bar{i}^t), \tag{4}$$

$$\bar{f}^t = W_f x^t + R_f y^{t-1} + p_f \odot c^{t-1} + b_f, \tag{5}$$

$$f^t = \sigma(\bar{f}^t), \tag{6}$$

$$c^t = z_t \odot i^t + c^{t-1} \odot f^t, \tag{7}$$

$$\bar{o}^t = W_o x^t + R_o y^{t-1} + p_o \odot c^t + b_o, \tag{8}$$

$$o^t = \sigma(\bar{o}^t), \tag{9}$$

$$y^t = h(c^t) \odot o^t, \tag{10}$$

where σ is the logistic sigmoid function, g and h are hyperbolic tangent functions, and \odot is the point-wise multiplication of the two vectors.

Back-propagation through time

The other main operation is back-propagation. Back-propagation through time (BPTT) is the process of calculating the deltas of LSTM blocks and the gradient of the weights (Greff et al. 2017).

First, the deltas (δ) of LSTM blocks and the inputs are calculated. In the below equations, Δ^t is the vector of the deltas passed down from the above layer, and T is the transposition operator. Calculation of the deltas is performed as follows:

$$\delta y^t = \Delta^t + R_z^T \delta z^{t+1} + R_i^T \delta i^{t+1} + R_f^T \delta f^{t+1} + R_o^T \delta o^{t+1}, \tag{11}$$

$$\delta \bar{o}^t = \delta y^t \odot h(c^t) \odot \sigma'(\bar{o}^t), \tag{12}$$

$$\delta \bar{c}^t = \delta y^t \odot o^t \odot h'(c^t) + p_o \odot \delta \bar{o}^t + p_i \odot \delta \bar{i}^{t+1} + p_f \odot \delta \bar{f}^{t+1} + \delta c^{t+1} \odot f^{t+1}, \tag{13}$$

$$\delta \bar{f}^t = \delta c^t \odot c^{t-1} \odot \sigma'(\bar{f}^t), \tag{14}$$

$$\delta \bar{i}^t = \delta c^t \odot z^t \odot \sigma'(\bar{i}^t), \tag{15}$$

$$\delta \bar{z}^t = \delta c^t \odot i^t \odot g'(\bar{z}^t), \tag{16}$$

$$\delta x^t = W_z^T \delta \bar{z}^t + W_i^T \delta \bar{i}^t + W_f^T \delta \bar{f}^t + W_o^T \delta \bar{o}^t. \tag{17}$$

Then, the calculation of the gradient of the weights is performed. In the below formulas, $*$ can be any of $\{\bar{z}, \bar{i}, \bar{f}, \bar{o}\}$, $< *_{1}, *_{2} >$ corresponds to the outer product of the two vectors, and T is the vector length. The calculations are as follows:

$$\delta W_* = \sum_{t=0}^T < \delta *^t, x^t >, \tag{18}$$

$$\delta R_* = \sum_{t=0}^{T-1} < \delta *^{t+1}, y^t >, \tag{19}$$

$$\delta b_* = \sum_{t=0}^T \delta *^t, \tag{20}$$

$$\delta p_i = \sum_{t=0}^{T-1} c^t \odot \delta \bar{i}^{t+1}, \quad (21)$$

$$\delta p_f = \sum_{t=0}^{T-1} c^t \odot \delta \bar{f}^{t+1}, \quad (22)$$

$$\delta p_o = \sum_{t=0}^{T-1} c^t \odot \delta \bar{o}^t. \quad (23)$$

Using Eqs. 11–23, all weights are updated.

Technical indicators

A technical indicator is a time series that is obtained from mathematical formula(s) applied to another time series, which is typically a price (TIO 2018). These formulas generally use the close, open, high, low, and volume data. Technical indicators can be applied to anything that can be traded in an open market (e.g., stocks, futures, commodities, and Forex). They are empirical assistants that are widely used in practice to identify future price trends and measure volatility (Ozorhan et al. 2017). By analyzing historical data, they can help forecast the future prices.

According to their functionalities, technical indicators can be grouped into three categories: lagging, leading, and volatility. Lagging indicators, also referred to as trend indicators, follow the past price action. MA and MACD are the best examples of lagging indicators. Leading indicators, also known as momentum-based indicators, aim to predict future price trend directions and show rates of change in the price. ROC and RSI are the best-known examples of leading indicators. Volatility-based indicators measure volatility levels in the price. BB is the most widely used volatility-based indicator.

The technical indicators used in this study are described below.

Moving average (MA)

Moving average (MA) is a trend-following (or lagging) indicator that smooths prices by averaging them in a specified period. In this way, MA can help filter out noise. MA can not only identify the trend direction but also determine potential support and resistance levels (TIO 2018).

Moving average convergence divergence (MACD)

Moving average convergence divergence (MACD) is a momentum oscillator 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 (Appel 2005). MACD uses the short-term moving average to identify price changes quickly and the long-term moving average to emphasize trends (Ozorhan et al. 2017).

Rate of change (ROC)

Rate of change (ROC) is a momentum oscillator that defines the velocity of the price. This indicator measures the percentage of the direction by calculating the ratio between the current closing price and the closing price of the specified previous time (Ozorhan et al. 2017).

Momentum

Momentum measures the amount of change in the price during a specified period (Colby 2003). It is a leading indicator that either shows rises and falls in the price or remains stable when the current trend continues. Momentum is calculated based on the differences in prices for a set time interval (Murphy 1999).

Relative strength index (RSI)

The relative strength index (RSI) is a momentum indicator developed by J. Welles Wilder in 1978. RSI is based on the ratio between the average gain and average loss, which is called the relative strength (RS) (Ozorhan et al. 2017; Wilder 1978). RSI is an oscillator, which means its values change between 0 and 100. It determines overbought and oversold levels in the prices.

Bollinger bands (BB)

Bollinger bands (BB) refers to a volatility-based indicator developed by John Bollinger in the 1980s. It has three bands that provide relative definitions of high and low according to the base (Bollinger 2001). While the middle band is the moving average in a specific period, the upper and lower bands are calculated by the standard deviations in the price, which are placed above and below the middle band. The distance between the bands depends on the volatility of the price (Bollinger 2001; Ozturk et al. 2016).

Commodity channel index (CCI)

The commodity channel index (CCI) is a momentum-based indicator developed by Donald Lambert in 1980. CCI is based on the principle that current prices should be examined based on recent past prices, not those in the distant past, to avoid confusing present patterns (Lambert 1983). This indicator can be used to highlight a new trend or warn against extreme conditions. Moreover, CCI identifies overbought and oversold conditions (Özorhan 2017).

The data set

Interest and inflation rates are two fundamental indicators of the strength of an economy. In the case of low interest rates, individuals tend to buy investment tools that strengthen the economy. In the opposite case, the economy becomes fragile. If supply does not meet demand, inflation occurs, and interest rates also increase (IRD 2018).

Germany and the US are two of the world's most powerful economies. In such economies, the stock markets have strong relationships with their currencies. DAX is the German stock index, which has a strong relationship on the price of the EUR while

the S&P 500 is one a US stock index that affects the USD. Central banks' interest rates are also important factors determining the prices of currencies. Therefore, the interest rates determined by the Central Bank of Europe and the Fed directly affect EUR and USD prices, respectively.

In this work, to investigate the effect of macroeconomic factors on the value of the EUR/USD currency pair, we used the factors described in Table 1, as well as the close, open, high, and low values of the EUR/USD pair, which were retrieved from EUR/USD historical data (EUR 2018). The rest of the data were obtained from various online resources, including the ECB Statistical Data Warehouse (ECB 2018; EU 2018; Germany 2018), Bureau of Labor Statistics Data (2018), Federal Reserve Economic Data (EFFR 2018), and Yahoo Finance (DAX 2018).

The data set was created with values from the period January 2013–January 2018. This 5-year period contains 1234 data points in which the markets were open. There were 613 increases and 620 decreases for the EUR/USD ratio during this period. Table 1 presents explanations for each field in the data set. Monthly inflation rates were collected from the websites of central banks, and they were repeated for all days of the corresponding month to fill the fields in our daily records.

LSTM-based hybrid model using macroeconomic and technical indicators

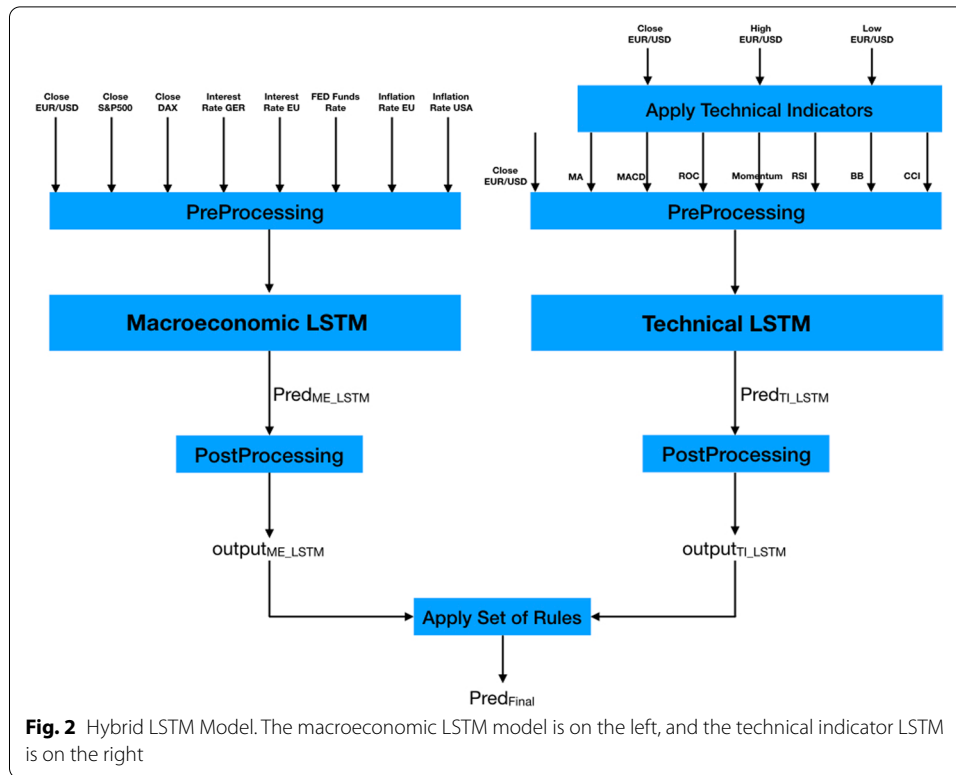
Using LSTM, we constructed a hybrid model to forecast directional movement in the EUR/USD currency pair that uses both macroeconomic and technical indicators. This hybrid model consists of two separate LSTM models that learn different parameter settings for different input sets (Yıldırım and Toroslu 2019). These models are called “macroeconomic LSTM” (ME-LSTM) and “technical LSTM” (TI-LSTM); they are explained below in “[Macroeconomic LSTM model](#)” and “[Technical LSTM model](#)” sections, respectively.

The main structure of the hybrid model, as shown in Fig. 2, can be summarized as follows:

- 1 Preprocess the dataset.
- 2 Train ME-LSTM and postprocess its results.

Table 1 Macroeconomic data and the currency pair used in the data set

Term	Explanation
<i>Close(EURUSD)</i>	Daily close value of euro/dollar currency pair
<i>Open(EURUSD)</i>	Daily open value of euro/dollar currency pair
<i>High(EURUSD)</i>	Daily high value of euro/dollar currency pair
<i>Low(EURUSD)</i>	Daily low value of euro/dollar currency pair
<i>Inflation Rate_{EU}</i>	Monthly inflation rate for the European area
<i>Inflation Rate_{USA}</i>	Monthly inflation rate for the US area
<i>Interest Rate_{GER}</i>	Monthly interest rate in Germany
<i>Interest Rate_{EU}</i>	Monthly interest rate in the European area
<i>FED Funds Rate</i>	Daily Fed funds rates
<i>Close(S&P 500)</i>	Standard and Poor—daily close American stock market index
<i>Close(DAX)</i>	Daily close German stock index



- 3 Train TI-LSTM and postprocess its results.
- 4 Apply different strategies to combine these LSTMs and use their individual results.

Baseline LSTMs

As a baseline, ME-LSTM and TI-LSTM were tested separately. Also, by combining all of the features of these two into a single model, we generated a third baseline model: ME-TI-LSTM.

Macroeconomic LSTM model

This LSTM model (ME_LSTM) was built to investigate the effects of macroeconomic factors on the price movement of the EUR/USD pair. These factors, which are explained in detail in “[The data set](#)” section, are listed below:

- Interest rates of Germany and the EU
- FED funds rate (for the US)
- Inflation rates in the EU and the US
- Close value of the S&P 500 market index
- Close value of the DAX market index

After the preprocessing phase, the ME_LSTM model was trained using all of these macroeconomic factors together with the closing values of the EUR/USD pair.

Technical LSTM model

This LSTM model (TI_LSTM) is formed by using technical indicators to observe their effects on the price movement of the EUR/USD 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 the preprocessing stage, the TI_LSTM model is trained using these seven technical indicators together with the closing values of the EUR/USD pair.

Macroeconomic and technical LSTM model

This LSTM model (ME_TI_LSTM) was formed using all of the macroeconomic and technical indicators taken together to observe the effects of the combined set of indicators. After the preprocessing stage, ME_TI_LSTM was trained using the macroeconomic and technical indicators mentioned above together with the closing values of the EUR/USD currency pair.

Proposed model: hybrid LSTM model

Our proposed model does not combine the features of the two baseline LSTMs into a single model. Instead, we propose a rule-based decision mechanism that acts as a kind of postprocessing; it is used to combine the results of the baselines into a final decision (Yıldırım and Toroslu 2019).

Training classifiers and labeling the data

We trained ME-LSTM, TI-LSTM, and ME-TI-LSTM using the same settings. The data set was split into the training and test sets, with ratios of 80% and 20%, respectively. The training phase was carried out with different numbers of iterations (50, 100, and 150).

Our data points were labeled based on a histogram analysis and the entropy approach. At the end of these operations, we divided the data points into three classes by using a threshold value:

- *Class_inc*: Corresponds to an increase in a price that is more than a threshold value.
- *Class_dec*: Corresponds to a decrease in a price that is more than a threshold value.
- *Class_noact*: Corresponds to a price change that is less than a threshold value.

In addition to the usual classes, *increase* and *decrease*, we introduced a third class *no_action*, which corresponds to the changes remaining in a predefined threshold range that is sufficiently small and thus negligible. Only when a difference between two consecutive data points is greater/less than the threshold will the next data point be labeled as

increase/decrease. Otherwise, we treated the next data point as unaltered. This new class enabled us to eliminate some data points for generating risky trade orders. This helped us improve our results compared to the binary classification results. This approach generates a fewer number of trades but with higher accuracy, as reported in “[Experiments](#)” section.

Histogram analysis and threshold calculation

In addition to the decrease and increase classes, we needed to determine the threshold we could use to generate a third class—namely, a no-action class—corresponding to insignificant changes in the data. Algorithm 1 was used to determine the upper bound of this threshold value. The aim was to prevent exploring all of the possible difference values and narrow the search space. In other words, we assumed that the optimal threshold value should be in the range of [0, threshold_upper_bound] instead of [0, max_of_differences].

The idea of Algorithm 1 is to determine the upper bound of the threshold based on 85% coverage of all differences. To do that, first, histogram analysis was performed on the closing prices of the EUR/USD pair to determine the distributions of price changes occurring in the data during consecutive days.

We placed the EUR/USD ratio differences between consecutive days into 10 bins (as *number_of_bins* value), which range equally between the minimum (which is 0) and maximum difference values. We determined the count of each bin and sorted them in descending order. After that, the counts of the bins were summed until the sum exceeded 85% of the whole count (the data set size). Then, the maximum difference value of the last bin added was used as the upper bound of the threshold value.

As can be seen in Algorithm 1, it has two phases. In the first phase, which simply corresponds to line 2, the whole data set is processed linearly to determine the distributions of the differences, using a simple histogram construction function. The second phase is depicted in detail, corresponding to the rest of the algorithm. To improve the threshold construction operation, an upper bound of the potential threshold was calculated as the value that is larger than 85% of the differences between two consecutive days’ closing

Algorithm 1 Histogram Analysis.

```

1: procedure HISTOGRAM_ANALYSIS(close_diff[ ], number_of_bins)           ▷ close(t) - close(t - k)
2:   bin_counts[ ], bin_max_diff[ ] ← perform_histogram(close_diff[ ], number_of_bins)
3:   ▷ bin_counts, bin_max_diff are reverse sorted arrays according to bin_counts values
4:   temp_sum ← 0
5:   sum_bin_counts ← sum(bin_counts[ ])
6:   i ← 1
7:   while i ≤ number_of_bins do
8:     temp_sum ← temp_sum + bin_counts[i]
9:     if temp_sum/sum_bin_counts > 0.85 then
10:      break
11:    i ← i + 1
12:   threshold_upper_bound ← bin_max_diff[i]
13:   return threshold_upper_bound

```

values.

The threshold value should be determined based on entropy. Entropy is related to the distribution of the data. The following formula defines entropy where p_i corresponds to the probability of the occurrence of class i :

$$Entropy = - \sum p_i * \log p_i. \quad (24)$$

To get balanced distribution, we calculated the entropy of class distribution in an iterative way for each threshold value up until the maximum difference value. However, we precalculated the threshold of the upper bound value and used it instead of the maximum difference value. After limiting the iteration number to the upper bound of the threshold found in the histogram analysis, we aimed to find the final threshold τ , which maximizes entropy. Algorithm 2 shows the details of our approach.

In Algorithm 2, to find the best threshold, potential threshold values are attempted with increments of 0.00001. Dropping the maximum threshold value is thus very important in order to reduce the search space. The main *while* loop is used to try each threshold value between 0 and the *threshold_upper_bound* with increments of 0.00001. For each threshold value, the number of increases (labeled as 2) and decreases (labeled as 1) above the threshold value are both determined, and the rest of the changes are assumed to be *no_change* (labeled 0). Then, the entropy value for this distribution is calculated. At the end of the *while* loop, the distribution that gives the best entropy is determined, and that distribution is used to determine the increase, decrease, and no-change classes.

In our experiments, we observed that in most cases, the threshold upper bound approach significantly reduced the search space (i.e., searching for the threshold value). In a typical case, this improvement corresponds to reducing the search space to around 20% of the original. For example, in one case, the maximum difference value was 0.029, but our approach determined the upper bound of the threshold value to be 0.00652. In

Algorithm 2 Threshold Calculation.

```

1: procedure CALCULATE_THRESHOLD(close_diff[ ]) ▷ close(t) - close(t - k)
2:   threshold_upper_bound ← HISTOGRAM_ANALYSIS(close_diff[ ], 10)
3:   temp_threshold ← 0
4:   best_entropy = -Infinity
5:   while temp_threshold < threshold_upper_bound do
6:     labels ← {0}
7:     indexes_incr ← indexOf(close_diff[ ] > temp_threshold)
8:     indexes_decr ← indexOf(-close_diff[ ] > temp_threshold)
9:     labels{indexes_incr} ← 2
10:    labels{indexes_decr} ← 1
11:    entropy ← calculate_entropy(labels[ ])
12:    if entropy > best_entropy then
13:      best_entropy ← entropy
14:      threshold ← temp_threshold
15:      temp_threshold ← temp_threshold + 0.00001
16:  return threshold

```

this case, the optimum threshold value was found to be 0.0023.

Postprocessing

The purpose of this processing is to determine the final class decision. We combined the predictions of the ME_LSTM and TI_LSTM models with the following set of rules:

- If one model's prediction is *class_noact*, then the final decision will be *class_noact*.
- If both models agree on the labels, we set the final decision as this label.

- If the predictions of the two models are different, we choose for the final decision the one whose prediction has higher probability. If the probability is the same, we choose the prediction of the TI_LSTM model.

This is a type of conservative approach to trading; it reduces the number of trades and favors only high-accuracy predictions.

Performance metric

Measuring the accuracy of the decisions made by these models also requires a new approach. Consider that during the testing phase of one of the LSTMs, our model predicts the class as “increase” (or “decrease”), but according to our three-class classification, it actually corresponds to a “no_act” class. In that case, we check if the actual movement is in the same direction with the prediction; that is, there was an “increase” (or “decrease”) but with less than the threshold value. If that is the case, then the prediction is correct, and we treat this test case as the correct classification.

We introduced a new performance metric to measure the success of our proposed method. We defined profit_accuracy as the accuracy that is related to the number of increases and decreases in the predicted labels. We can interpret this metric such that it gives the ratio of the number of profitable transactions over the total number of transactions, defined using Table 2. In the below formula, the following values are used:

- True_dec: the number of true predictions decreases
- True_inc: the number of true predictions increases
- False_dec_noact: the number of predictions of the no-action class decreases
- False_inc_noact: the number of predictions of the no-action class increases
- False_inc_dec: the number of predictions of the decrease class increases
- False_dec_inc: the number of predictions of the increase class decreases

Note that in the above formula, there is no case corresponding to the “True_inc_noact” and the “True_dec_noact” counts since we converted such decisions into “True_inc” and “True_dec,” respectively, as explained above.

$$\begin{aligned}
 & \textit{ProfitAccuracy} \\
 &= \frac{\textit{True_dec} + \textit{True_inc}}{\textit{False_dec_noact} + \textit{False_inc_noact} + \textit{True_dec} + \textit{False_inc_dec} + \textit{False_dec_inc} + \textit{True_inc}} \tag{25}
 \end{aligned}$$

Table 2 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

Table 3 Data set statistics (training and test sets)

	Threshold	# of no_action	# of decrease	# of increase
One day ahead	0.0023	412 (334–78)	400 (327–73)	402 (310–92)
Three days ahead	0.0040	413 (317–96)	414 (357–57)	385 (295–90)
Five days ahead	0.0055	400 (311–89)	422 (370–52)	388 (287–101)

Experiments

After applying the labeling algorithm, we obtained a balanced distribution of the three classes over the data set. This algorithm calculates different threshold values for each period and forms different sets of class distributions. For predictions of different periods, the thresholds and corresponding number of data points (explicitly via training and test sets) in each class are calculated, as shown in Table 3.

This table shows that the class distributions of the training and test data have slightly different characteristics. While the class *decrease* has a higher ratio in the training set and a lower ratio in the test set, the class *increase* shows opposite behavior. Class *no_action*, meanwhile, is more stable in both sets. This is because a split is made between the training and test sets without shuffling the data sets to preserve the order of the data points.

We collected daily EUR/USD rates for a total of 1214 consecutive days. We used the first 971 days of this data to train our models and the last 243 days to test them. Our models aims to determine if there will be an “increase” or “decrease” in the next day, 3 days ahead, and 5 days ahead of the day of the prediction. If one of these is predicted, a transaction is considered to be started on the test day ending on the day of the prediction (1, 3, or 5 days ahead). Otherwise, no transaction is started. A transaction is successful and the traders profit if the prediction of the direction is correct.

Experiments on long-term real data

For time-series data, LSTM is typically used to forecast the value for the next time point. It can also forecast the values for further time points by replacing the output value with not the next time point value but the value for the chosen number of data points ahead. This way, during the test phase, the model predicts the value for that many time points ahead. However, as expected, the accuracy of the forecast usually diminishes as the distance becomes longer.

Zhang et al. (2017) used a very similar LSTM model for stock price prediction. They defined it as an n-step prediction as follows:

$$\hat{p}_{t+n} = f(p_1, p_2, \dots, p_t).$$

This simply corresponds to mapping the history of prices from p_1 to p_t into n-steps ahead. They performed experiments for 1, 3, and 5 days ahead. In their experiments, the accuracy of the prediction decreased as n became larger.

Our experiments also involved 1-day, 3-day, and 5-day predictions of the directional movement of the EUR/USD currency pair. We used individual LSTM models and the simple combined LSTM as baselines and compared them with our proposed hybrid

Table 4 ME_LSTM model: one-day-ahead result summary

	Profit_accuracy (%)	# of transactions
Iterations=50	46.50	243/243
Iterations=100	55.00	60/243
Iterations=150	48.97	243/243
Iterations=200	52.27	52/243
Average	50.69	149.50/243

model. We also present the number of total transactions made on test data for each experiment. Accuracy results are obtained for transactions that are made.

For each experiment, we performed 50, 100, 150, and 200 iterations in the training phases to properly compare different models. The execution times of the experiments were almost linear with the number of iterations. For our data set, using a typical high-end laptop (MacBook Pro, 2.7 GHz dual-core Intel Core i5 processor, 8 GB memory, 256 GB disk space), the training phase for 200 iterations took more than 7 h.

Forecasting one day ahead

Macroeconomic LSTM model results

As seen in Table 4, this model shows huge variance in the number of transactions. Meanwhile, the profit_accuracy results show small variance, with 50.69% ± 3,72% accuracy on average. Additionally, the average predicted transaction number is 149.50, which corresponds to 61.52% of the test data.

Technical LSTM model results

In these experiments, whose results are shown in Table 5, the profit_accuracy results are also close to each other, with 52.18% ± 1.93% accuracy on average. For this LSTM model,

Table 5 TI_LSTM model: one-day-ahead result summary

	Profit_accuracy	# of transactions
Iterations=50	50.26	189/243
Iterations=100	50.93	161/243
Iterations=150	53.11	177/243
Iterations=200	54.43	94/243
Average	52.18	155.25/243

Table 6 ME_TI_LSTM model: one-day-ahead result summary

	Profit_accuracy	# of transactions
Iterations=50	47.62	231/243
Iterations=100	55.40	139/243
Iterations=150	46.67	195/243
Iterations=200	62.50	64/243
Average	53.05	157.25/243

Table 7 Hybrid model: one-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
50	200	81.63	52/243	82.35	52/243
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
100	200	73.17	46/243	75.00	46/243
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
150	200	100.00	8/243	100.00	8/243
200	50	80.00	20/243	76.47	20/243
200	100	80.77	28/243	80.77	28/243
200	150	84.00	26/243	82.61	26/243
200	200	83.33	28/243	85.71	28/243
Average		77.32	64.75/243	77.76	64.75/243

the average predicted transaction number is 155.25, which corresponds to 63.89% of the test data.

Macroeconomic and technical LSTM model results

The results for this model are shown in Table 6. The profit_accuracy results have higher variance, with 53.05% ± 7.42% accuracy on average. The average predicted transaction number is 157.25, which corresponds to 64.71% of the test data. One major difference of this model is that it is for 200 iterations. For this test case, the accuracy significantly increased, but the number of transactions dropped even more significantly.

Hybrid LSTM model results

Table 7 summarizes the profit_accuracy values and the number of transactions for each case in this model. In some experiments, the number of transactions is quite low. In particular, for 200 iterations, our model generated very few transactions, which corresponds to the “increase” and “decrease” predictions. Basically, the total number of decrease and increase predictions are in the range of [8, 137], with an overall average of 64.75. That value corresponds to a transaction ratio of $64.75/243 = 26.65\%$. Moreover, we obtained an average profit_accuracy in 16 cases of $77.32\% \pm 7.82\%$ and $77.76\% \pm 8.33\%$ for ME_LSTM- and TI_LSTM-based modified hybrid models, respectively, where 7.82 and 8.33 represent standard deviations.

When we analyze the results for one-day-ahead predictions, we observe that although the baseline models made more transactions (89.25 more on average out of 243), our hybrid model predicted more accurately (25,57% better on average).

Table 8 ME_LSTM model: three-days-ahead result summary

	Profit_accuracy (%)	# of transactions
Iterations=50	57.44	242/243
Iterations=100	54.40	182/243
Iterations=150	39.83	236/243
Iterations=200	53.57	38/243
Average	51.31	174.50/243

Table 9 TI_LSTM model: three-days-ahead result summary

	Profit_accuracy (%)	# of transactions
Iterations=50	43.31	157/243
Iterations=100	47.78	180/243
Iterations=150	51.37	146/243
Iterations=200	51.85	103/243
Average	48.58	146.50/243

Table 10 ME_TI_LSTM model: three-days-ahead result summary

	Profit_accuracy (%)	# of transactions
Iterations=50	43.16	234/243
Iterations=100	43.81	226/243
Iterations=150	42.68	164/243
Iterations=200	85.71	10/243
Average	53.84	158.50/243

Forecasting three days ahead

Macroeconomic LSTM model results

Table 8 presents the results of these experiments. According to the results, profit_accuracy had high variance, with $51.31\% \pm 7.83\%$ accuracy on average. Additionally, the average predicted transaction number is 174.50, which corresponds to 71.81% of the test data. One significant observation concerns the huge drop in the number of transactions for 200 iterations without any increase in accuracy.

Technical LSTM model results

As shown in Table 9, in this set of experiments, the profit_accuracy results showed smaller variance, with $48.58\% \pm 3.95\%$ on average. Furthermore, the variance in the number of transactions is also smaller; the average predicted transaction number is 146.50, which corresponds to 60.29% of the test data. There is a drop in the number of transactions for 200 iterations but not as much as with the macroeconomic LSTM.

Table 11 Hybrid model: three-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
50	200	94.74	21/243	95.00	21/243
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
100	200	100.00	2/243	100.00	2/243
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
150	200	95.00	43/243	95.00	43/243
200	50	77.78	9/243	75.00	9/243
200	100	84.62	13/243	84.62	13/243
200	150	100.00	9/243	100.00	9/243
200	200	100.00	14/243	100.00	14/243
Average		78.98	65.13/243	79.23	65.13/243

Table 12 ME_LSTM model: five-days-ahead result summary

	Profit_accuracy (%)	# of transactions
Iterations=50	43.40	235/242
Iterations=100	47.11	242/242
Iterations=150	44.74	228/242
Iterations=200	54.00	120/242
Average	47.31	206.25/242

Macroeconomic and technical LSTM model results

The results for this model are presented in Table 10. The profit_accuracy results are very close to each other, except at 200 iterations, with $53.84\% \pm 21.25\%$ accuracy on average. Additionally, the average predicted transaction number is 158.50, which corresponds to 65.23% of the test data. However, the case with 200 iterations is quite different from the others, with only 10 transactions out of a possible 243 generating a very high profit accuracy.

Hybrid LSTM model results

Table 11 shows the profit_accuracy values and the number of transactions for each case. The total number of “decrease” and “increase” predictions are in the range of [2, 155]. On average, this value is 65.13, which corresponds to a transaction ratio of $65.13/243 = 26.80\%$. Moreover, the average profit_accuracies are $78.98\% \pm 15.02\%$ and $79.23\% \pm 15.06\%$ for the ME_LSTM- and TI_LSTM-based modified hybrid

Table 13 TI_LSTM model: five-days-ahead result summary

	Profit_accuracy (%)	# of transactions
Iterations=50	48.13	187/242
Iterations=100	41.48	176/242
Iterations=150	45.73	164/242
Iterations=200	64.18	79/242
Average	49.88	151.50/242

Table 14 ME_TI_LSTM model: five-days-ahead result summary

	Profit_accuracy (%)	# of transactions
Iterations=50	44.44	81/242
Iterations=100	42.72	206/242
Iterations=150	46.51	172/242
Iterations=200	61.25	96/242
Average	48.73	138.75/242

models, respectively. There are also some very striking cases with 100% accuracy, involving 200 iterations for at least one of the LSTM models. However, all of these cases produced a very small number of transactions.

When we compare the results, similar to the one-day-ahead cases, we observe that the baseline models produced more transactions (more than 94.70 out of 243 on average), but the hybrid model predicted more accurately (27.87% better on average).

Forecasting 5 days ahead

Macroeconomic LSTM model results

The results of these experiments are shown in Table 12. According to the results, the profit_accuracy values have small variance, with $47.31\% \pm 4.71\%$ accuracy on average. Additionally, the average predicted transaction number is 206.25, corresponding to 85.23% of the test data.

Technical LSTM model results

Table 13 shows the results of these experiments. The profit_accuracy results have higher variance in these experiments, especially in the case of 200 iterations, with $49.88\% \pm 9.92\%$ accuracy on average. The average predicted transaction number is 151.50, corresponding to 62.60% of the test data. Again, the case of 200 iterations shows huge differences from the other cases, generating less than half the number of the lowest number of transactions generated by the others.

Macroeconomic and technical LSTM model results

Table 14 shows the results of these experiments. Similar to the technical LSTM model, the profit_accuracy results are close to each other, except at 200 iterations, with an overall average accuracy of $48.73\% \pm 8.49\%$. Meanwhile, the average predicted transaction

Table 15 Hybrid model: five-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
50	200	88.57	36/242	87.88	36/242
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
100	200	100.00	9/242	100.00	9/242
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
150	200	Nan	0/242	Nan	0/242
200	50	92.68	43/242	92.68	43/242
200	100	86.36	44/242	86.05	44/242
200	150	90.00	40/242	87.88	40/242
200	200	86.00	51/242	85.11	51/242
Average		84.08	69.31/242	83.44	69.31/242

Table 16 Extended data set statistics (training and test sets)

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

number is 138.75, corresponding to 57.34% of the test data. However, the case of 200 iterations is not an exception, and there is huge variance among the cases.

Hybrid LSTM model results

Table 15 presents the profit_accuracy values and the number of transactions for each case in these experiments. The total number of “decrease” and “increase” predictions is in the range of [0, 112]. On average, this value is 69.31, corresponding to a transaction ratio of $69.31/242 = 28.64\%$. Moreover, the overall average profit_accuracies are $84.08\% \pm 6.54\%$ and $83.44\% \pm 6.69\%$ for the ME_LSTM- and TI_LSTM-based modified hybrid models, respectively.

From the five-days-ahead prediction experiments, we observe that, similar to the one-day- and three-days-ahead experiments, the baseline models produced more transactions (more than 96.19 on average out of 242), but the hybrid model predicted more accurately (35.12% better on average).

Table 17 Hybrid model (on extended dataset): one-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
50	200	81.82	52/152	81.82	52/152
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
100	200	75.34	84/152	74.65	84/152
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
150	200	84.13	75/152	84.38	75/152
200	50	83.08	71/152	81.36	71/152
200	100	79.31	67/152	79.66	67/152
200	150	79.31	64/152	78.18	64/152
200	200	85.45	62/152	85.19	62/152
Average		70.93	73.19/152	72.19	73.19/152

Experiments using recent real data

To further validate our results, we extended our data set to include a very recent one—namely, EUR/USD rates from January 1, 2018, to April 1, 2019. This extended data set has 1539 data points, which contain 761 increases and 777 decreases overall. Applying our labeling algorithm, we formed a data set with a balanced distribution of three classes. Table 16 presents the statistics of the extended data set.

The extended data set is split into training and test sets, with ratios of 90% and 10%, respectively. Below, we report one-day-, three-days-, and five-days-ahead prediction results for our hybrid model based on the extended data.

Forecasting one day ahead

Table 17 presents the profit_accuracy values and the number of transactions for each case. The total number of “decrease” and “increase” predictions is in the range of [52, 97]. The average the number of predictions is 73.19, corresponding to a transaction ratio of $73.19/152 = 48.15\%$. Moreover, the average profit_accuracies in the 16 cases are $70.93\% \pm 10.60\%$ and $72.19\% \pm 10.14\%$ for the ME_LSTM- and TI_LSTM-based modified hybrid models, respectively.

Forecasting three days ahead

The results of these experiments are shown in Table 18. The total number of generated transactions is in the range of [2, 83]. Some cases with 200 iterations produced a very small number of transactions. The average number of transactions is 39.88, for a transaction ratio of $39.88/152 = 26.24\%$. Also, the average profit_accuracies are $71.76\% \pm$

Table 18 Hybrid model (on extended dataset): three-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
50	200	100.00	2/152	100.00	2/152
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
100	200	86.67	55/152	88.68	55/152
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
150	200	81.25	37/152	82.35	37/152
200	50	88.89	10/152	88.89	10/152
200	100	83.33	6/152	83.33	6/152
200	150	57.14	8/152	62.50	8/152
200	200	62.50	8/152	57.14	8/152
Average		71.76	39.88/152	70.30	39.88/152

Table 19 Hybrid model (on extended dataset): five-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
50	200	75.34	84/152	72.73	84/152
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
100	200	75.00	77/152	71.88	77/152
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
150	200	75.68	84/152	72.73	84/152
200	50	71.64	71/152	72.46	71/152
200	100	66.67	84/152	66.22	84/152
200	150	67.61	81/152	68.06	81/152
200	200	72.06	75/152	69.35	75/152
Average		71.24	71.56/152	68.25	71.56/152

13.77% and 70.30% ± 14.15% for the ME_LSTM- and TI_LSTM-based modified hybrid models respectively.

Table 20 Summary of all experiments conducted on the main data set

	One day ahead		Three days ahead		Five 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

Forecasting 5 days ahead

Table 19 shows the results for the five-days-ahead prediction experiments. Interestingly, the total numbers predictions are much closer to each other in all of the cases compared to the one-day- and three-days-ahead predictions. These numbers are in the range of [59, 84]. On average, the number of transactions is 71.56, corresponding to a transaction ratio of $71.56/152 = 47.08\%$. Moreover, the average profit_accuracy values are $71.24\% \pm 5.40\%$ and $68.25\% \pm 4.95\%$ for the ME_LSTM- and TI_LSTM-based modified hybrid models, respectively.

Discussion

Table 20 summarizes the overall results of the experiments. In the one-day-ahead predictions, the individual LSTM models had a slightly better profit_accuracy than ME_TI_LSTM, which was less than 1%. However, they produced 3.91% fewer transactions than ME_TI_LSTM on average. Moreover, when we combined the predictions of the individual models in our proposed model, it reached a much higher profit_accuracy of 73.09% (22.30% improvement) on average while reducing the number of transactions to 37.96%.

In the three-days-ahead predictions, the individual models had even better profit_accuracy results than ME_TI_LSTM by 5.81% but, again, with fewer transactions on average. In these experiments, there were huge differences in terms of the number of transactions generated by the two different LSTMs. While ME_LSTM produced more than 90% of the transactions, TI_LSTM only generated around 66%. Moreover, our proposed hybrid model showed a much better performance than the other three with a profit_accuracy of 68.31% (a 19.29% average improvement over the others). As in the above case, this higher accuracy was obtained by reducing the number of transactions to 42.57%.

Finally, in the five-days ahead predictions, the profit_accuracy results for individual LSTMs and the ME_TI_LSTM were very close. Similar to the three-days-ahead prediction, ME_LSTM produced a very high number of transactions, with more than 97%, while ME_TI_LSTM had the lowest, with an accuracy of around 63%. Moreover, the hybrid model showed an exceptional accuracy performance of 79.42% (34.33% improvement) by reducing the number of transactions to 32.72%.

Additional results of these experiments can be summarized as follows:

- ME_LSTM: The profit_accuracy of three-days-ahead predictions was slightly better than that of one-day-ahead predictions (by just 0.40%). Also, both were higher

than the five-days-ahead predictions, by 5.48% and 5.08%, respectively. The number of transactions became higher with further forecasting, for 87.52% on average. It is difficult to form a simple interpretation of these results, but, in general, we can say that with macroeconomic indicators, more transactions are generated.

- **TI LSTM:** Profit_accuracy decreased when we extended the prediction period, falling within the range of [45.11–51.43%]. The number of transactions was less in the five-days-ahead predictions than in the one-day and three-day predictions. Compared to the ME_LSTM, these results show that there is no winner between the two individual LSTMs.
- **ME TI LSTM:** The profit_accuracy of the one-day-ahead predictions was the highest, at 49.89%. Additionally, the profit_accuracy of the five-days-ahead prediction was 1.34% higher than the three-days-ahead prediction. The transaction number ratio over the test data varied and was around 75.44% on average. These results also show that a simple combination of two sets of indicators did not produce better results than those obtained individually from the two sets.
- **Hybrid model:** Our proposed model, as expected, generated much higher accuracy results than the other three models. In both one-day- and three-days-ahead cases, the improvement was above 20%, and in the five-days-ahead case, it was even higher, with an improvement of more than 30% compared to the other three LSTMs. Interestingly, the performance of profit_accuracy was the highest in the five-days-ahead predictions. Moreover, in all cases, it generated the smallest number of transactions compared to the other models (40.37% on average).

The main motivation for our hybrid model solution was to avoid the drawbacks of the two different LSTMs (i.e., macroeconomic and technical LSTMs). When the ME_LSTM and TI_LSTM were executed separately using the features of their corresponding data sets (i.e., macroeconomic features and technical indicator features), they generated too many transactions. Some of these transactions were generated with not very good signals and thus had lower accuracy results. When all features were simply appended to each other, in what we call ME_TI_LSTM, the results did not change much.

Although the two individual baseline LSTMs used completely different data sets, their results seemed to be very similar. Actually, their accuracy results can be interpreted as failure since they were around 50%. Even though LSTMs are, in general, quite successful in time-series predictions, even for applications such as stock price prediction, when it comes to predicting price direction, they fail if used directly. That is why there are not many results reported involving using LSTMs for Forex.

Moreover, combining two data sets into one seemed to improve accuracy only slightly. For that reason, we developed a hybrid model that takes the results of two individual LSTMs separately and merges them using smart decision logic. In real data, fluctuations in the EUR/USD ratio are usually very small. That is why incorrect directional predictions made by LSTMs correspond to a very small amount of errors. This causes LSTMs to produce models making many such predictions with incorrect directions.

In our hybrid model, weak transaction decisions are avoided by combining the decisions of two LSTMs with a simple set of rules that also take the no-action decision

Table 21 Performance comparison of hybrid model

	One day ahead		Three days ahead		Five days ahead	
	Profit_ accuracy (%)	# of transactions (%)	Profit_ accuracy (%)	# of transactions (%)	Profit_ accuracy (%)	# of transactions (%)
Main data set	73.09	37.86	68.31	42.57	79.42	40.68
Extended data set	64.24	50.88	63.91	37.43	68.58	43.05

into consideration. This extension significantly reduced the number of transactions, by mostly preventing risky ones. As can be seen in Table 20, which summarizes all of the results, the new approach predicted fewer transactions than the other models. Moreover, the accuracy of the proposed transactions of the hybrid approach is much higher than that of the other models.

Comparing the performances of the hybrid model on the main data set and the extended data set, we see some decreases in the profit_accuracy results and some changes in the number of transactions. We present this comparison in Table 21. From these results, we can say the hybrid model’s behavior on the extended data set was very similar to that obtained using the main data set. In other words, the best performance occurred for five-days-ahead predictions, and one-day-ahead predictions is slightly better than three-days-ahead predictions, by 0.33%. Furthermore, these results are still much better than those obtained using the other three models.

We can also conclude that as the number of transactions increased, it reduced the accuracy of the model. This was an expected result, and it was observed in all of the experiments. Depending on the data set, the number of transactions generated by our model could vary. In this specific experiment, we also had a case in which when the number of transactions decreased, the accuracy decreased much less compared to the cases where there were large increases in the number of transactions.

In most financial markets, accurate predictions above 50% technically generate profits. Considering other costs and risks, we can conclude that more than 60% prediction accuracy is a very successful result, and we showed that our hybrid model always had an accuracy of greater than 60%.

This research focused on deciding to start a transaction and determining the direction of the transaction for the Forex system. In a real Forex trading system, there are further important considerations. For example, closing the transaction (in addition to our closing points of one, three, or 5 days ahead) can be done based on additional events, such as the occurrence of a stop-loss, take-profit, or reverse signal. Another important consideration could be related to account management. The amount of the account to be invested at each transaction could vary. The simplest model might invest the whole remaining account at each transaction. However, this approach is risky, and there are different models for account management, such as always investing a fixed percentage at each transaction. Another important decision is how to determine the leverage ratio to be chosen for each transaction. Simple models use fixed ratios for all transactions. Moreover, the leverage ratio can be determined using the strength of model’s decision.

Conclusion

This study applied two separate LSTM models to forecast the directional movement of the EUR/USD currency pair. Our predictions included periods of one day, three days, and 5 days ahead. We designed a classifier to determine the direction of the EUR/USD pair. In our proposed model, there are three classes: *no_action*, *decrease*, and *increase*. *No_action* means that if the changes between two time points are below a predefined threshold, they are negligible and require no action. This enabled us to introduce a new performance metric, *profit_accuracy*, which gives us the ratio of the number of profitable transactions over the total number of transactions. We simply defined profitable transaction as a correct prediction of the decrease and increase classes. Predicting the correct direction of a currency pair presents the opportunity to profit from the transactions. This was the main objective of our study. This metric met our expectation completely since the predicted class *no_action* had no contribution to the profit/loss of the transaction.

We used a balanced data set with almost the same number of increases and decreases. Thus, our results were not biased. Two baseline models were implemented, using only macroeconomic or technical indicator data. We observed that, compared to TI_LSTM, ME_LSTM had a slightly better performance in terms of both *profit_accuracy* and the number of transactions generated. However, the difference was very small and insignificant. Furthermore, combining all of the features into a single LSTM, called ME_TI_LSTM, did not significantly increase accuracy.

Meanwhile, our proposed hybrid model had the best performance in terms of *profit_accuracy* for predictions in all periods (73.61% on average). It reduced the number of transactions compared to the baseline models (40.37% on average). The increase in accuracy can be attributed to dropping risky transactions.

The proposed hybrid model was also tested using a recent data set. The results of the experiments were in line with the other experiments, showing only a small decrease in *profit_accuracy*.

The main contributions of this work can be summarized as follows:

- Macroeconomic and technical indicators can both be used to train LSTMs, separately or together, to predict the directional movement of currency pairs in Forex. We showed that rather than combining these parameters into a single LSTM, processing them separately with different LSTMs and combining their results using smart decision logic improved prediction accuracy significantly.
- Rather than trying to determine whether the currency pair rate will increase or decrease, a third class was introduced—a no-change class—corresponding to small changes between the prices of two consecutive days. This, too, improved the accuracy of direction prediction. We described a novel way to determine the most appropriate threshold value for defining the no-change class.
- LSTMs can be trained to determine not only the next day's value but also the values for k-days ahead. We used this feature to predict three days and 5 days ahead, with some decreases in accuracy values.
- Typically, the accuracy of LSTMs can be improved by increasing the number of iterations during training. We experimented with various iterations to determine

their effects on accuracy values. The results showed that more iterations increased accuracy while decreasing the number of transactions (i.e., potential profits and risks are simultaneously reduced).

In future research, our work could be extended to other currency pairs, such as EUR/GBP, GBP/USD, USD/CHF, GBP/CHF, and EUR/CHF. Additionally, a trading simulator could be developed to further validate the model. Such a simulator could be useful for observing the real-time behavior of our model. However, for such a simulator to be meaningful, several issues related to real trading (e.g., closing the account, account management, leverage ratio decision) must be carefully investigated.

Acknowledgements

Not applicable.

Authors' contributions

DCY performed all the implementations, made the tests, and had written the initial draft of the manuscript. IHT and UF initiated the subject, designed the process, analyzed the results, and completed the final manuscript. All authors read and approved the final manuscript.

Funding

This research received no external funding.

Availability of data and materials

All of the data used in the experiments are publicly available (USD/EUR rates, interest rates, US and German stock market indexes, etc.).

Competing interests

The authors declare that they have no competing interests.

Author details

¹ Department of Computer Engineering, Middle East Technical University, 06800 Ankara, Turkey. ² Department of Management Studies and Quantitative Methods, Parthenope University, 80132 Napoli, NA, Italy.

Appendix

Moving average (MA)

The formulation of MA is given as follows:

$$SMA = \frac{SUM(Close, N)}{N}. \quad (26)$$

In Eq. 26, SMA is the simple moving average, Close is the closing price of the currency pair, N is the period, and SUM(Close, N) is the sum of closing prices in period N.

Moving average convergence divergence (MACD)

The formulation of MACD is given as follows:

$$MACDLine = EMA(Close, N1) - EMA(Close, N2), \quad (27)$$

$$EMA = Close - EMA(previous) * C + EMA(previous), \quad (28)$$

where

$$C = \frac{2}{N + 1}. \quad (29)$$

In Eq. 27, MACDLine is the MACD line of the price; Close is the close price; EMA and EMA(previous) are the current and previous period's EMA values (EMA is the

exponential moving average), respectively; N_1 and N_2 are the short- and long-term moving average periods ($N_1 = 12$ and $N_2 = 26$ are in default), N is the total number of periods in a simple moving average to be roughly approximated by the EMA; and C is the exponential smoothing constant.

Rate of change (ROC)

The formulation of ROC is given as follows:

$$ROC = \frac{Close - Close(previous, N)}{Close(previous, N)} * 100. \quad (30)$$

In Eq. 30, ROC is the rate-of-change value, N is the period, and $Close$ and $Close(previous, N)$ are the closing price and the closing price N periods ago, respectively.

Momentum

The formulation of momentum is given as follows:

$$Momentum = Close - Close(previous, N). \quad (31)$$

In Eq. 31, momentum is the momentum value of the currency pair. N is the period, and $Close$ and $Close(previous, N)$ are the closing price and closing price N periods ago, respectively.

Relative strength Index (RSI)

The formulation of RSI is given as follows:

$$AverageGain = \frac{AverageGain(previous) * (N - 1) + CurrentGain}{N}, \quad (32)$$

$$AverageLoss = \frac{AverageLoss(previous) * (N - 1) + CurrentLoss}{N}, \quad (33)$$

$$RS = \frac{AverageGain}{AverageLoss}, \quad (34)$$

$$RSI = 100 - \frac{100}{1 + RS}. \quad (35)$$

In Eq. 35, RS and RSI are the relative strength and relative strength index values, respectively. $CurrentGain$ and $CurrentLoss$ are the positive and negative absolute difference values between the current and previous period's closing price, respectively. $AverageGain(Previous)$, $AverageLoss(Previous)$, $AverageGain$, and $AverageLoss$ are the previous period's average gain and loss and the current average gain and loss in N periods, respectively.

Bollinger bands (BB)

The formulation of BB is given as follows:

$$\text{MiddleBand} = \text{SMA}(\text{Close}, 20), \quad (36)$$

$$\text{UpperBand} = \text{MiddleBand} + \text{SD}(\text{Close}, 20) * 2, \quad (37)$$

$$\text{LowerBand} = \text{MiddleBand} - \text{SD}(\text{Close}, 20) * 2. \quad (38)$$

In Eqs. 36–38, MiddleBand, UpperBand, and LowerBand are the Bollinger bands of the price. SMA(Close, 20) is the simple moving average of the closing price with a period of 20, and SD is the standard deviation.

Commodity channel index (CCI)

The formulation of CCI is given as follows:

$$\text{CCI} = \frac{\text{TypicalPrice} - \text{SMA}(\text{TypicalPrice}, N)}{\text{MeanDeviation} * L}, \quad (39)$$

where

$$\text{TypicalPrice} = \frac{\text{High} + \text{Low} + \text{Close}}{3}, \quad (40)$$

$$\text{MeanDeviation} = \frac{\sum |\text{TypicalPrice} - \text{SMA}(\text{TypicalPrice}, N)|}{N}. \quad (41)$$

In Eq. 39, CCI is the commodity channel index value of the price. Typical price is the typical price of the currency pair. N is the period, SMA is the simple moving average, MeanDeviation is the mean deviation, and L is the Lambert coefficient, equal to 0.015.

Received: 9 October 2019 Accepted: 11 December 2020

Published online: 04 January 2021

References

- Appel G (2005) Technical analysis: power tools for active investors. Financial Times Prentice Hall, p 241. <http://www.ncbi.nlm.nih.gov/pubmed/15003161>
- Archer MD (2010) Getting started in currency trading: winning in today's Forex market. Wiley, London, p 333
- Bahrammirzaee A (2010) A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Comput Appl* 19:1165–1195
- Ballings M, Van Den Poel D, Hespeels N, Gryp R (2015) Evaluating multiple classifiers for stock price direction prediction. *Expert Syst Appl* 42:7046–7056
- Biehl M (2005) Supervised sequence labelling with recurrent neural neural networks. *Neural Netw* 1999:1
- Bollinger J (2001) Bollinger on bollinger bands. McGraw-Hill, London. <https://goo.gl/0ro0fB>
- Bureau of Labor Statistics Data (November 2018). https://data.bls.gov/timeseries/CUUR0000SA0L1E?output_view=pct_12mths. Accessed: Nov 2018
- Cavalcante RC, Brasileiro RC, Souza VLF, Nobrega J, Oliveira ALI (2016) Computational intelligence and financial markets: a survey and future directions. *Expert Syst Appl* 55:194–211
- Colby RW (2003) The encyclopedia of technical market indicators, p 833
- DAX PERFORMANCE INDEX-Yahoo Finance (November 2018). <https://finance.yahoo.com/quote/%5EGDAXI/history?period1=1483218000&period2=1514754000&interval=1d&filter=history&frequency=1d&guccounter=1>. Accessed: Nov 2018
- Di Persio L, Honchar O (2016) Artificial neural networks architectures for stock price prediction: comparisons and applications. *Int J Circuits Syst Sig Process* 10:403–413
- Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, Darrell T (2017) Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans Pattern Anal Mach Intell* 39:677–691
- ECB Statistical Data Warehouse (November 2018). <https://sdw.ecb.europa.eu/>. Accessed: Nov 2018

- EU 25 (November 2018) (fixed composition) as of 1 May 2004, Long-term interest rate for convergence purposes—Unspecified rate type, Debt security issued, 10 years maturity, New business coverage, denominated in All currencies combined—Unspecified counterpart sector—Quick View—ECB Statistical Data Warehouse. http://sdw.ecb.europa.eu/quickview.do?SERIES_KEY=229.IRS.M.D3.L.L40.CI.0000.Z01.N.Z. Accessed: Nov 2018
- EUR USD Historical Data-Investing.com (November 2018). <https://www.investing.com/currencies/eur-usd-historical-data>. Accessed: Nov 2018
- Effective Federal Funds Rate (DFF) (November 2018). <https://fred.stlouisfed.org/series/DFF>. Accessed: Nov 2018
- Fan Y, Qian Y, Xie F, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Proceedings of the annual conference of the international speech communication association. Interspeech, pp 1964–1968
- Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res* 270:654–669
- Forex & Currencies Trading: An Introduction (November 2018). <https://www.investopedia.com/forex-and-currencies-trading-4689676>. Accessed: Nov 2018
- Fulfillment IP, Gao Q, Supervisor T (2016) Stock market forecasting using recurrent neural network: a thesis presented to. MSc thesis, Faculty of Graduate School, University of Missouri, Columbia, USA
- Galeshchuk S, Mukherjee S (2017) Deep networks for predicting direction of change in foreign exchange rates. *Intell Syst Account Finance Manag* 24:100–110
- Germany (November 2018), Long-term interest rate for convergence purposes—Unspecified rate type, Debt security issued, 10 years maturity, New business coverage, denominated in Euro—Unspecified counterpart sector—Quick View—ECB Statistical Data Warehouse. http://sdw.ecb.europa.eu/quickview.do?SERIES_KEY=229.IRS.M.DE.L.L40.CI.0000.EUR.N.Z. Accessed: Nov 2018
- Gers FA, Schmidhuber J (2000) Recurrent nets that time and count. *Proc Int Joint Conf Neural Netw* 3:189–194
- Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. *Neural Comput* 12:2451–2471
- Ghazali R, JaafarHussain A, MohdNavi N, Mohamad B (2009) Non-stationary and stationary prediction of financial time series using dynamic ridge polynomial neural network. *Neurocomputing* 72:2359–2367
- Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans Pattern Anal Mach Intell* 31:855–868
- Graves A (2013) Generating sequences with recurrent neural networks. <http://arxiv.org/abs/1308.0850>
- Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schmidhuber J (2017) LSTM: a search space odyssey. *IEEE Trans Neural Netw Learn Syst* 28:2222–2232
- Gui B, Wei X, Shen Q, Qi J, Guo L (2015) Financial time series forecasting using support vector machine. In: Proceedings—2014 10th international conference on computational intelligence and security, CIS 2014, pp 39–43
- Guresen E, Kayakutlu G, Daim TU (2011) Using artificial neural network models in stock market index prediction. *Expert Syst Appl* 38:10389–10397
- Hochreiter S, Schmidhuber J (1997) Long short term memory. *Neural Comput* 9:1735–1780
- Hu H, Tang L, Zhang S, Wang H (2018) Predicting the direction of stock markets using optimized neural networks with Google Trends. *Neurocomputing* 285:188–195
- Huang W, Nakamori Y, Wang SY (2005) Forecasting stock market movement direction with support vector machine. *Comput Oper Res* 32:2513–2522
- Interest Rate Definition (November 2018). <https://www.investopedia.com/terms/i/interstrate.asp>. Accessed: Nov 2018
- Kara Y, Acar Boyacioglu M, Baykan ÖK (2011) Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange. *Expert Syst Appl* 38:5311–5319
- Kayal A (2010) A neural networks filtering mechanism for foreign exchange trading signals. In: IEEE international conference on intelligent computing and intelligent systems, pp 159–167
- Kou G, Chao X, Peng Y, Alsaadi FE, Herrera-Viedma E (2019) Machine learning methods for systemic risk analysis in financial sectors. *Technol Econ Dev Econ* 25:716–742
- Kou G, Xu Y, Peng Y, Shen F, Chen Y, Chang K, Kou S (2020) Bankruptcy prediction for SMES using transactional data and two-stage multiobjective feature selection. *Decis Support Syst* 2020:1–39
- Kritzer A, Service SO (2012) *Forex for beginners* [electronic resource]: a comprehensive guide to profiting from the global currency markets/by Adam Kritzer. Apress
- Lambert DR (1983) Commodity channel index: tool for trading. *Tech Anal Stocks Commod* 1:1–5
- Luong MT, Sutskever I, Le QV, Vinyals O, Zaremba W (2015) Addressing the rare word problem in neural machine translation. In: ACL-IJCNLP 2015-53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the Asian Federation of natural language processing, proceedings of the conference, vol 1, pp 11–19
- Majhi R, Panda G, Sahoo G (2009) Efficient prediction of exchange rates with low complexity artificial neural network models. *Expert Syst Appl* 36:181–189
- Marchi E, Ferroni G, Eyben F, Gabrielli L, Squartini S, Schuller B (2014) Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks. *ICASSP*. In: IEEE international conference on acoustics, speech and signal processing—proceedings, pp 2164–2168
- Murphy JJ (1999) *Technical analysis of the financial markets*. TA - Book, p 239
- Nelson DM, Pereira AC, De Oliveira RA (2017) Stock market's price movement prediction with LSTM neural networks. In: Proceedings of the international joint conference on neural networks 2017-May, pp 1419–1426
- Özorhan MO (2017) 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
- Ozorhan MO, Toroslu IH, Sehitoglu OT (2017) A strength-biased prediction model for forecasting exchange rates using support vector machines and genetic algorithms. *Soft Comput* 21:6653–6671

- Ozturk M, Toroslu IH, Fidan G (2016) Heuristic based trading system on Forex data using technical indicator rules. *Appl Soft Comput J* 43:170–186
- Patel J, Shah S, Thakkar P, Kotecha K (2015a) Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Syst Appl* 42:259–268
- Patel J, Shah S, Thakkar P, Kotecha K (2015b) Predicting stock market index using fusion of machine learning techniques. *Expert Syst Appl* 42:2162–2172
- Pham V, Bluche T, Kermorvant C, Louradour J (2014) Dropout improves recurrent neural networks for handwriting recognition. In: *Proceedings of international conference on frontiers in handwriting recognition, ICFHR 2014-December*, pp 285–290
- Qiu M, Song Y (2016) Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS ONE* 2016:11
- Saad EW, Wunsch PDVD (1998) Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Trans Neural Netw* 9:1456–1470
- Selvamuthu D, Kumar V, Mishra A (2019) Indian stock market prediction using artificial neural networks on tick data. *Financ Innov* 2019:5
- Shen F, Chao J, Zhao J (2015) Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing* 167:243–253
- Shen F, Zhao X, Kou G (2020) Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory. *Decis Support Syst* 137:113366
- Sønderby SK, Winther O (2014) Protein secondary structure prediction with long short term memory networks. [arXiv:1412.7828](https://arxiv.org/abs/1412.7828) [q-bio.QM]
- Technical Indicators and Overlays (ChartSchool) (November 2018). https://school.stockcharts.com/doku.php?id=technical_indicators. Accessed: Nov 2018
- Wang H, Kou G, Peng Y (2020) Multi-class misclassification cost matrix for credit ratings in peer-to-peer lending. *J Oper Res Soc* 2020:1–12
- Wen F, Xu L, Ouyang G, Kou G (2019) Retail investor attention and stock price crash risk: evidence from China. *Int Rev Financ Anal* 65:101376
- Weng B, Lu L, Wang X, Megahed FM, Martinez W (2018) Predicting short-term stock prices using ensemble methods and online data sources. *Expert Syst Appl* 112:258–273
- Wilder J (1978) New concepts in technical trading systems. *New Concepts Tech Trad Syst* 24:63–69
- Yıldırım DC, Toroslu IH (2019) Forecasting directional movement of Forex data using LSTM with technical and macro-economic indicators. MSc thesis, Graduate School of Natural and Applied Sciences, METU, Ankara, Turkey
- Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. [arXiv:1409.2329](https://arxiv.org/abs/1409.2329) [cs.NE]
- Zhang L, Aggarwal C, Qi GJ (2017) Stock price prediction via discovering multi-frequency trading patterns. In: *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining part, vol F129685*, pp 2141–2149
- Zhong X, Enke D (2017) Forecasting daily stock market return using dimensionality reduction. *Expert Syst Appl* 67:126–139
- Zhong X, Enke D (2019) Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financ Innov* 2019:5
- Zia T, Zahid U (2019) Long short-term memory recurrent neural network architectures for Urdu acoustic modeling. *Int J Speech Technol* 22:21–30

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
