International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-021-00279-9

REGULAR PAPER

A novel ensemble deep learning model for stock prediction based on stock prices and news

Yang Li[1] · Yi Pan[1]

Received: 17 September 2020 / Accepted: 6 August 2021

Abstract
In recent years, machine learning and deep learning have become popular methods for financial data analysis, including financial textual data, numerical data, and graphical data. One of the most popular and complex deep learning in finance topics is future stock prediction. The difficulty that causes the future stock forecast is that there are too many different factors that affect the amplitude and frequency of the rise and fall of stocks at the same time. Some of the company-specific factors that can affect the share price like news releases on earnings and profits, future estimated earnings, the announcement of dividends, introduction of a new product or a product recall, secure a new large contract, employee layoffs, a major change of management, anticipated takeover or merger, and accounting errors or scandals. Furthermore, these factors are only company factors, and other factors affect the future trend of stocks, such as industry performance, investor sentiment, and economic factors. This paper proposes a novel deep learning approach to predict future stock movement. The model employs a blending ensemble learning method to combine two recurrent neural networks, followed by a fully connected neural network. In our research, we use the S&P 500 Index as our test case. Our experiments show that our blending ensemble deep learning model outperforms the best existing prediction model substantially using the same dataset, reducing the mean-squared error from 438.94 to 186.32, a 57.55% reduction, increasing precision rate by 40%, recall by 50%, $F1$-score by 44.78%, and movement direction accuracy by 33.34%, respectively. The purpose of this work is to explain our design philosophy and show that ensemble deep learning technologies can truly predict future stock price trends more effectively and can better assist investors in making the right investment decision than other traditional methods.

Keywords Stock prediction · Deep learning · Machine learning · Ensemble learning · Statistical finance

1 Introduction

Many factors may affect stock prices in various ways. The stock prices change by market forces, which means the stock price changes react to supply and demand in the stock market. If more people want to buy a stock (demand) than sell it (supply), then the price moves up. Similarly, if more people want to sell a stock than buy it, there would be greater supply than demand, and the price would fall. Stock supply and demand are affected by many things. Supply factors include company share issues (e.g., releases new shares to the public), share buybacks (e.g., a company buys back its own shares from investors to reduce supply) and sellers (e.g., the investors responsible for pushing shares back into the market, increasing the supply). Demand factors include company news (e.g., a new product launch, missed targets, good performance), economic factors (e.g., interest rate changes), industry trends (e.g., a booming industry), market sentiment (could be psychological and subjective) and unexpected events (e.g., natural disasters or the death of a government leader). Normally, we can get these supply and demand factors from the financial news, companies' newsletters or their annual reports. For instance, when Apple announces a new product, many people would like to purchase it, and its performance usually would be better soon. Thus, more people are interested in the Apple stock, and then, the Apple stock demand increases, which will lead to a rise in the Apple stock price. On the other hand, when COVID-19 spreads around the world, many airlines cut their flights, and it is

✉ Yang Li
yangli.atlanta@gmail.com

Yi Pan
yipan@gsu.edu

[1] Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

Published online: 17 September 2021

expected their performance would be bad in a short term. Thus, more people want to sell airline stocks; then, the airline stock supply will rise, and their price will go down. If the price goes up, the quantity demanded goes down (but demand itself stays the same). If the price decreases, quantity demanded increases. This is the Law of Demand. If the quantity demanded decreases, the stock price probably would fall. Also people's sentiment or belief plays a role in determining a stock price. Political situations or international affairs may also affect stock prices. Hence, this is a complicated process among the stock supply, demand, and prices. However, there are a few primary factors that affect the stock supply and demand like company news, company performance, industry performance, investor sentiment (e.g., whether in bull or bear market), and other major economic factors described in [1]. If we focus on the major factors, and trace back the historical stock prices, we may be able to predict future stock prices quite accurately. People usually have a short memory about stock factors. Hence, determining a suitable historical window size is important to correctly predict stock prices. If the window size is too large compared with human memory, many factors or news are forgotten by investors and obsolete already and the prediction will not be good. On the other hand, if the window is too short compared with human memory, many news or sentiments outside the window are still remain in people's brain, the prediction will also be bad. Hence, a wrong historical window size is detrimental to our successful stock price predictions.

Stock price prediction is a series of continuous predictions since the stock price is constantly changing to react to timely news and announcements. Therefore, it is very challenging for computer scientists to use artificial intelligence to predict future stock movements because it is hard for a computer to receive the latest information and respond immediately. Computer scientists are currently not particularly successful in stock price prediction for several reasons. Most of the previous works [2,3] often used either textual data like news, twitter, or blogs or numerical data like stock price information instead of using both textual information and statistical information [4]. Since the stock price is related to many factors, only considering one or two factors is unable to provide enough information to forecast the stock price trend. Including as much relevant and useful information as possible will guarantee a better prediction.

Furthermore, previous works [2,3] only use the target company's information on the training model without considering that the target company's competitors or the information of companies in related industries. These types of information will also affect the target company's stock movement. Therefore, the result is not very satisfactory and persuasive because the information provided is insufficient. Moreover, some of the previous works, which used the textual information, did not consider time series. However, the timeline is a significant factor for stock price prediction. The major contributions of this paper are: First, we employ sentimental analysis to get sentimental scores from different news agencies. Second, we use two complementary modern deep learning architectures to obtain predictions based on both scores and historical data. Third, we utilize a fancy ensemble neural network to fuse the decisions to further improve prediction results. The major steps are as follows. This paper proposes to use sentiment analysis to extract useful information from multiple textual data sources and a blending ensemble deep learning model to predict future stock movement. The blending ensemble model contains two levels. The first level contains two recurrent neural networks, one long-short term memory network (LSTM) and one gated recurrent units network (GRU), followed by a fully connected neural network as the second level model. The RNNs, LSTM, and GRU models can effectively capture the time series events in the input data, and the fully connected neural network is used to ensemble several individual prediction results to further improve the prediction accuracy.

2 Related work

There are many related works in the stock prediction domain. However, five previous works have a significant impact on this research. In 2017, Nelson [5] proposed to use LSTM networks with some technical analysis indicators to predict stock price compare with some baseline models like support vector machines (SVM), random forest (RF), and multi-layer perceptron (MLP). Their results have shown that LSTM has considerable improvement in terms of accuracy and offers fewer risks. One of the significant challenges of using artificial intelligence to predict the stock trend with news is the weight of news events in it and the extent to which each news event affects the trend of stock. Hu [6] proposed using a hybrid attention network (HAN) with three principles, including sequential context-dependency, diverse influence, and effective and efficient learning to reduce noises for low-quality information and improve the impact on stocks for high-quality events. Last year, Li [7] proposed a novel approach to use differential privacy to robust the LSTM model for stock prediction. The experimental results have shown that using differential privacy can robust the LSTM model and improve prediction results. The differential privacy inspired LSTM (DP-LSTM) approach inspired us to attempt to use a different approach to predict stock movements. In the paper "Deep Learning for Stock Prediction Using Numerical and Textual Information" [4] is also discussed using textual and numerical data to predict future stock price. The authors stated that converting newspaper article' titles into distributed representations via paragraph vector and modeling the temporal effects of the past events

on opening prices about multiple companies with LSTM could increase stock price prediction accuracy. This previous work also suggests using numerical data and textual data as primary sources to predict future stock prices with LSTM. Leonardo Pinheiro and Mark Dras showed that they explored RNNs with character-level language models using pre-training for intraday and interday stock market forecasting. Their technique is competitive with other state-of-the-art approaches [8]. Our architecture leverages their successful experiences and creates a new model to perform a better prediction.

3 Preliminaries

3.1 Data source

The data used in this research were obtained from the paper "DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News" [7]. We taxonomize the data into two categories: news and stock data; the news data are obtained from CNBC.com, Reuters.com, WSJ.com, Fortune.com, and dates range from December 2017 up to the end of June 2018. CNBC is the world leader in business news and has a real-time financial market coverage. Reuters is an international news organization founded in October 1851; it is one of the industry leaders for online information for tax, accounting, and finance news. The Wall Street Journal (WSJ) is one of the largest American business-focused news organization based in New York City. Fortune is an American multinational business magazine. We consider these four financial news data because these are the four most prominent business news organizations and hence the quality of the news there is exceptional. For the news data, only news articles from the financial domain were considered. Moreover, Ding et. al. [9] advised that news titles can provide adequate information to represent news articles and are more helpful for prediction compared to the article's contents. Besides, the article's content might add extra noises to the model that might cause the model to have poor performance, and it is also hard to accurately summarize the article's content using natural language processing (NLP). Hence, we only use the title of the news to extract sentiment scores. The stock data are the S&P 500 Index with the same date range as the news data. The S&P 500 Index is a stock market index that measures the stock performance of the 500 largest publicly traded companies in the USA. The S&P 500 is one of the best representations of the US stock market. Since our experiment uses news data and stock data to predict future stock market movement and prices, we will only use adjusted closing stock price as the target value. Adjusted closing price amends a stock's closing price to accurately reflect that stock's value after accounting for any corporate actions. It is considered to be the true price of that stock and is often used when examining historical returns or performing a detailed analysis of historical returns.

3.2 Data pre-processing

The news data are pre-processed with Aware Dictionary and Sentiment Reasoner (VADER) to generate sentiment scores. VADER is a lexicon and rule-based model for general sentiment analysis [10]. According to Kirlic's research, there is almost no difference between VADER and human decision making [11]. In addition, VADER not only provides positive and negative scores, but also gives us how positive or negative the sentiment is. Many python libraries include a pre-trained VADER model that makes it very convenient and efficient for us to use. After pre-processing the news data, the VADER will give a compound score; the compound score is a metric that calculates the sum of all the lexicon ratings, which has a normalized value between -1, which represents most extreme negative, and $+1$, which represents most extreme positive [7]. Of course, 0 means neutral news. For example, if the news title is "The Price of the U.S. Dollar is Rising" and its compound score is 0.64, it means this news title is positive, and the positivity weight is about 0.64. The opposite example would be "The Price of the U.S. Dollar is Falling" and if its compound score is -0.56, it means this news title is negative, and the negativity weight is about -0.56.

During the pre-processing, all the null data are removed from the dataset, and all news data and stock data are combined together. Since the stock market only opens and closes during the weekdays, therefore weekends are not included in the dataset. Our dataset contains 121 trading days and a total of six columns; the first column contains the date corresponding to the news data and stock data in the next 5 columns. The news data contain the WSJ news compound score, the Reuters news compound score, the CBNC news compound score, the Fortune news compound score, and the stock data are the adjusted closing prices.

As shown in Fig. 1, we have split the data into three parts: training data, validation data, and test data. The training data are those obtained from 12/07/2017 to 04/09/2018; the validation data are those data between 04/10/2018 and 05/04/2018; the test data are from 05/07/2018 to 06/01/2018. The training dataset is used to train the level 0 sub-models, the validation dataset is used to prepare the level 1 model, and the test dataset is used to evaluate our prediction performance. The details of the model architecture will be discussed later in Sect. 4.4.

Since we are dealing with time series forecasting, a rolling window with fixed-size 10 is used to provide different time steps. As shown in Fig. 2, we are using the past 10 days' financial news from multiple sources and stock prices to predict the next day's stock price. The rolling window data are

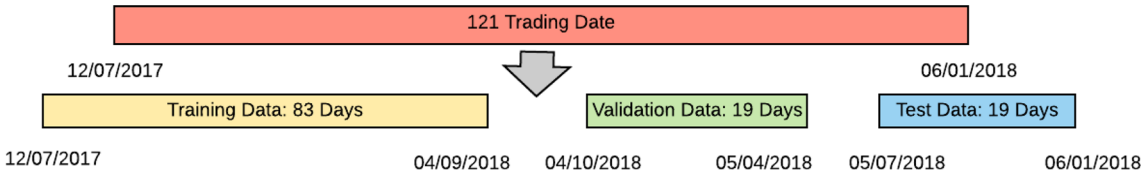

Fig. 1 Schematic diagram of the split dataset

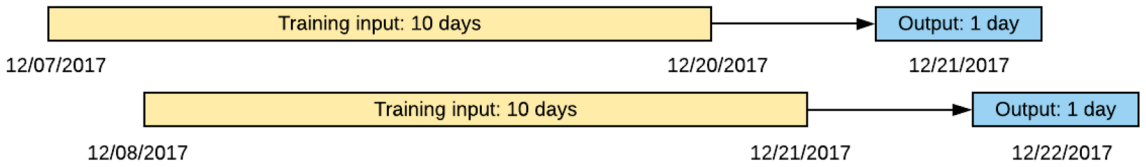

Fig. 2 Schematic diagram of the time step window

historical stock prices, and then, we shift the rolling window by one day and add the next actual stock price to the last date of the rolling window to predict the next day's stock price and so forth. According to the length of the window, the training data are divided into 83-time steps. The validation data are divided into 9-time steps, and the test data are divided into 9 time steps.

Normalization is a rescaling of the data from the original range; all scaled values are within the range between 0 and 1. Since the compound scores are numbers between −1 and 1, to avoid overfitting and improve accuracy, we rescaled the adjusted closing stock prices between 0 and 1.

4 Methodology

Stock price prediction is classified in the time-series category due to its unique characteristic, which means stock price prediction is a continual prediction following the direction of time. The most common techniques used for stock forecasting are statistics algorithms and economics models. However, the results coming out from there are not satisfactory because statistical algorithms and economics models cannot capture the stock movement's patterns. In artificial intelligence, the core techniques are pattern recognition using arithmetic calculations and sequences. RNNs utilize their internal memory to process variable-length sequences of input data, which makes them well suited for time series forecasting [12]. In particular, LSTM and GRU are the first choices because they have used future stock predictions successfully before. However, previous research has found that for a single LSTM network or GRU network, unless scrupulous parameter optimization is performed, data trained with a specific dataset are likely to perform poorly on completely different time-series datasets. After extensive research and experiments, we have found that stack or combine multiple RNNs will provide a more accurate prediction compared to a single LSTM network [13]. Therefore, we decide to deploy a blending ensemble learning model that combines LSTM and GRU to accomplish this difficult task. The main differences between LSTM and the GRU are the exposure of memory content inside the unit and how new information is processed by each unit. For the LSTM unit, the amount of memory content that is seen controlled by the output gate (not all of the content are exposed to other units; the output gate decides what information will be used in the next LSTM unit). The GRU unit exposes its full content to other units without any control. When LSTM receives new content, these new contents will be transported to the forget gate because the forget gate decides what information will be throw away or be kept before the computation process. However, the GRU does not have the forget gate; instead, the GRU utilizes the update gate to control the previous unit's information flow when computing the new candidate activation [14]. Even though these two models are similar, the way they process data and computation process steps are different. These differences might have an impact on weights when dealing with stock and news data. We have found that sometimes the LSTM prediction is more close to the actual stock price during our experiment, while other times the GRU prediction performs better. As we know, some news contents may affect stock more and longer than usual, and other contents may affect stock in a short time. Due to its controlled exposure of the memory content in LSTM, it can filter out a lot of news contents. On the other hand, GRU can outperform LSTM both in terms of convergence in CPU time and in terms of parameter update and generalization [14]. Thus, both models have their strengths and weaknesses, and in our design we hope to use

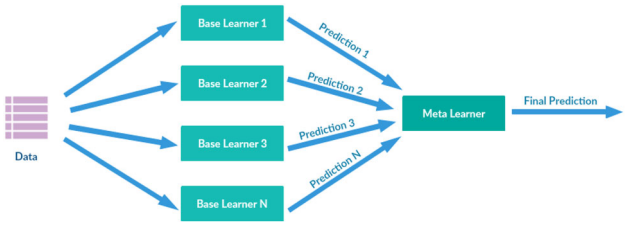

Fig. 3 *Source*:https://www.kdnuggets.com/2019/01/ensemble-learning-5-main-approaches.html

both models with different parameters to complement each other in order to achieve the best prediction results.

4.1 Ensemble learning techniques

The ensemble learning method combines decisions from multiple sub-models to a new model and then to make the final output to improve the prediction accuracy or the overall performance (see Fig. 3). There are many different ensemble learning models: Max Voting, Averaging, Weighted Average, Stacking, Blending, Bagging, Boosting, Adaptive Boosting (AdaBoost), Gradient Boosting Machine (GBM), eXtreme Gradient Boosting (XGB), etc. [15]. Different ensemble models have different characteristics and can be used to solve different problems in various domains. A simple example to describe the ensemble learning method is that compared with an individual's decision, a diverse group of people are more likely to make a better decision. The same principle applies to machine learning and deep learning models; a different set of models are more likely to perform a better comparison to a single model [16] since each model has their own strength and they can complement each other to overcome their individual shortcomings.

4.2 Long short-term memory neural networks

Humans do not start their thinking from scratch every second. They understand each word based on their understanding of previous words. Hence, memory is important in recognition, and traditional neural networks do not have this memory capability. The long short-term memory (LSTM) is a special kind of recurrent neural network originally proposed by Hochreiter and Schmidhuber in 1997 [17]. LSTM contains some memory cells and can solve many time series tasks unsolvable by feed-forward networks or other machine learning models [18]. LSTM is very suitable and particularly successful in the time-series domain because LSTM can store important past information into the cell state and forget the unimportant information. LSTM has three gates that complete these complex tasks by adjusting input and stored information in the cell state. The first gate is the forget gate layer, which decides what information the unit will eliminate from the cell state. The forget gate equation is defined as:

$$f_t = \sigma(W_{\mathrm{f}} \cdot [h_{t-1}, x_t] + b_{\mathrm{f}}) \tag{1}$$

where f_t represents the forget gate at the time step t; σ represents a sigmoid function; W_{f} represents the weights for the forget neurons; h_{t-1} represents the output of the previous cell state at time step $t-1$; x_t represents the input value at current time step; b_{f} represents the biases for the forget gates.

There are two steps when LSTM decides what new information the unit will store in the cell state. The first step is the input gate layer, which determines which values will be updated. The second step is the tanh layer, which generates a new value-added to the present cell state. The equations are defined as:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

where i_t represents the input gate at the time step t; W_i represents the weights for the input neurons; $\widetilde{C}_t$ represents the candidate for the cell state at time step t; b_i and b_C represents the bias for the respective gates. The last gate is the output gate layer, which determines what information will be output. The output gate equation is defined as:

$$o_t = \sigma(W_{\mathrm{o}} \cdot [h_{t-1}, x_t] + b_{\mathrm{o}}) \tag{4}$$

where o_t represents the output gate at the time step t; W_{o} represents the weights for the output neurons; and b_{o} represents the biases for the output gates [17].

We implemented an LSTM which uses the past 10 days as the time window, and input data include adjusted closing stock price, four news sentiment compound scores to predict the next day's adjusted closing stock price. The details of the LSTM structures will be discussed in Sect. 4.4.

4.3 Gated recurrent unit neural networks

A gated recurrent unit (GRU) was deployed and proposed by Cho et al. in 2014 [19] to solve the vanishing gradient problem of the traditional RNN by using an update gate and a reset gate. GRU is also a special kind of recurrent neural network that is very similar to LSTM; GRU has gating units that regulate the flow of information inside the unit. However, GRU does not have a separate memory cell, and that is one of the main differences between GRU and LSTM. The performance of the LSTM and GRU is equally matched in different test environments [14]. However, GRU is computationally more efficient because GRU does not have to use a memory unit. Besides, GRU is more suitable and performs better when dealing with small datasets. The GRU's update gate decides how much of the past information needs

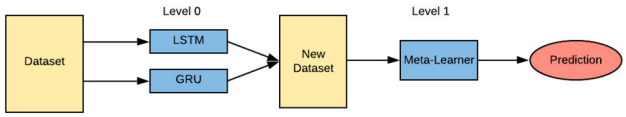

Fig. 4 Architecture overview

to update before passing to the next step. The update gate equation is defined as:

$$z_t = \sigma(W^z x_t + U^z h_{t-1}) \tag{5}$$

where z_t represents the update gate at time step t; W^z represents the weights for the update gate; x_t represents the input at time step t; h_{t-1} represents the holding values for the previous $t-1$ units; and U^z represents the weights for the h_{t-1}.

The second principal component of the GRU is the reset gate. The reset gate decides how much of the past information needs to forget. The reset gate equation is defined as:

$$r_t = \sigma(W^r x_t + U^r h_{t-1}) \tag{6}$$

where r_t represents the reset gate at time step t; W^r represents the weights for the reset gate; GRU can store and filter the information by utilizing the update gates and reset gates. The update and reset gates effectively eliminate the RNN vanishing gradient problem; they store relevant information in the memory cell and pass the values down to the next time steps of the network.

4.4 Architecture overview

The concept of ensemble learning is to use different types of machine learning and deep learning models combined to make predictions or classifications. We deploy an ensemble model called the blending ensemble model; the overview of the architecture is shown in Fig. 4.

We have tested many different configuration combinations during our experiments. Based on our empirical experiences, we choose our parameters such as epoch size, number of neurons, and number of layers. The blending ensemble model has two levels: the first level contains two RNNs; sub-model 1 is the LSTM, and sub-model 2 is the GRU model. We already divided the dataset into three parts: training data (from 12/07/2017 to 04/09/2018), validation data (from 04/10/2018 to 05/04/2018), and test data (from 05/07/2018 to 06/01/2018). Each dataset is essential to train the blending ensemble model. The training data are used to train level 1 sub-models: the LSTM model and the GRU model. After the first-phase training, we use the trained level 1 models to make predictions on the validation data, which is basically the level 2 model's training data. And the test data are used to make the final prediction and accuracy calculation.

First, we use the training data to train the sub-model 1: LSTM model. This LSTM model has only four layers, and each layer contains 50 neurons. We add 0.2 drops out for each hidden layer and train the model with 100 epochs. After we train the LSTM model, we input the validation dataset into the LSTM to make the first prediction with the validation dataset. The first prediction that the LSTM made using the validation is called LSTM validation predictions.

Secondly, we train the sub-model 2: the GRU model. The GRU model we build also contains four layers, and each layer has 50 neurons. We also add 0.2 drops out for each hidden layer and train the model with 100 epochs. The training process for the GRU model is the same as the LSTM. We input the training dataset into the GRU, and after we obtain a trained GRU model, we will input the validation dataset to make the GRU validation predictions.

Once the LSTM validation prediction and GRU validation prediction are obtained, we combine them into a new training dataset in the form of $p \times m$ (p represents number of predictions and m represents number of models). This new training data will pass to the second level to train the meta-learner. The meta-learner is also called the second-level model. The meta-learner is a fully connected neural network with three layers; the activation function for this model is the rectified linear unit (ReLu). After the meta-learner is trained, the test dataset will be input into the sub-models again to produce intermediate test data for the meta-learner. Afterward, the meta-learner will use the intermediate test predictions from the sub-models to make the final predictions.

5 Evaluation metrics

During the experiments, we mainly use four different evaluation metrics: mean square error (MSE), confusion matrix, mean prediction accuracy (MPA) and movement direction accuracy (MDA) to evaluate the blending ensemble model. The MSE is a risk function that measures the average squared difference between the predicted values and the actual value. The MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widetilde{y}_i)^2 \tag{7}$$

where n is the number of predictions, y_i is the vector of observed values of the variable being predicted, with $\widetilde{y}_i$ being the predicted values.

The confusion matrix is usually used in statistical classification, also known as the error matrix. As shown in Fig. 5, the confusion matrix is a unique table layout used to visualize the performance of an algorithm, classification scheme, or prediction model on a set of data where the actual values are known. The confusion matrix has different equations

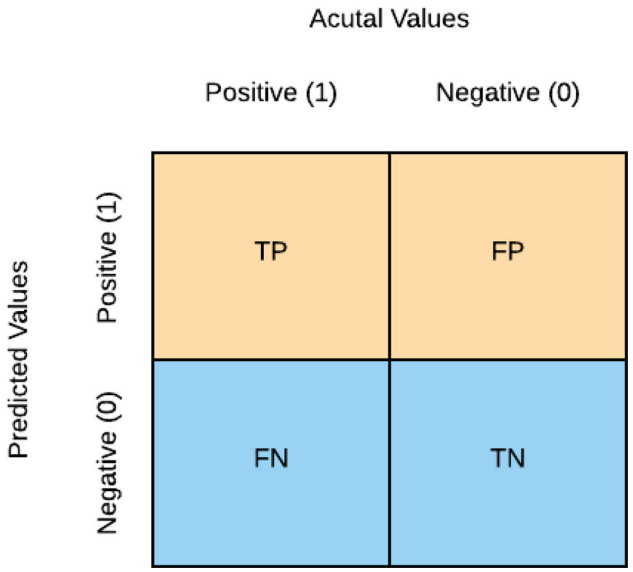

Fig. 5 Confusion table

to measure the performance of the model. In this paper, we will use the precision, recall, and $F1$-score to evaluate the blending ensemble model. Precision indicates how precise the model is out of the positive predictions by calculating how many predicted positive is actually positive. The precision is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

where TP represents the observation is positive, and the prediction is also positive. FP represents the observation is positive, and the prediction is negative.

The recall is defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

where FN represents the observation is positive, but the prediction is negative. Recall calculates the ratio of the actual positives the model captures through labeling it as positive. The recall score indicates what percentage of the class is correctly recognized.

The $F1$-score is the combination of the recall and precision. $F1$-score uses the harmonic mean of precision and recall to compute the accuracy of the model where the score reaches its best value at 1 and worst at 0.

$F1$-score evaluation equation:

$$\begin{aligned} F1\text{-score} &= \left(\frac{\text{Recall}^{-1} + \text{Precision}^{-1}}{2}\right)^{-1} \\ &= 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \end{aligned} \tag{10}$$

It penalizes the extreme values, which make it a better evaluation metric when dealing with imbalanced datasets. It also gives a better measure of the incorrectly classified cases than other metrics.

The next evaluation metric is the mean prediction accuracy (MPA) which is defined as:

$$\text{MPA}_t = 1 - \frac{l}{L} \sum_{l=1}^{L} \frac{\left| X_{t,l} - \hat{X}_{t,l} \right|}{X_{t,l}} \tag{11}$$

where X represents the actual stock price, $\hat{X}$ represents the predicted stock price, l represents the l-th stock, and t represents the day [7].

Finally, the movement direction accuracy (MDA) evaluates the percentage of correct prediction of stock movement directions (positive or negative) during the whole process. Movement direction accuracy (MDA) equation is define as:

$$\text{MDA} = \frac{\text{Number of Correct Movement Predictions}}{\text{Total Number of Movement Predictions}}. \tag{12}$$

6 Experimental results

During the experiments, we use the predicted stock price to compare with the actual stock price and calculate the MSE and MPA values. We then use the predicted values to calculate the price fluctuation of the stock on the forecast day; if the predicted stock price increases, the output is 1, and if the predicted stock price decreases, the output is 0. These results will be used for the confusion matrix to calculate various measurements defined above. We first compare the blending ensemble model with the LSTM model, which is the previous work widely used in the stock prediction domain, and then, we compare with another previous work called the DP-LSTM model. Based on the reported results in [7], the MSE is 198.75. However, we found an error in their code and after running the correct code, the actual MSE is 330.97. In the following comparisons, we will use the values from our experiments with the correct code. To further evaluate the blending ensemble model, we also deploy a GRU model that is very similar to the sub-model 2 that we were building as a test model. In addition, we also recorded the averaging ensemble model and weighted average ensemble model prediction results to contrast with the blending ensemble model.

As shown in Table 1, the blending ensemble model outperforms all other models in every category. The blending ensemble model has significant improvement in the MSE, precision, recall, and $F1$-score evaluation categories. The MPA results are very similar; the blending ensemble model increased by 0.36% compared with LSTM, increased 0.17% compared with DP-LSTM and increased by 0.08% compared

Table 1 Evaluation results

Evaluation metrics	LSTM	DP-LSTM	GRU	Averaging ensemble	Weighted average ensemble	Blending ensemble
MSE	438.94	330.97	249.34	231.16	229.52	**186.32**
MPA	99.29%	99.48%	99.57%	99.57%	99.57%	**99.65%**
Precision	25%	20%	40%	25%	40%	**60%**
Recall	25%	25%	50%	25%	50%	**75%**
$F1$-score	25%	22.22%	44.44%	25%	44.44%	**66.67%**
MDA	33.33%	22.22%	44.44%	33.33%	33.33%	**66.67%**

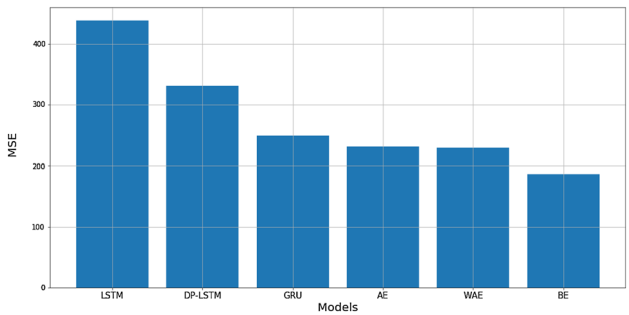

Fig. 6 MSE comparison: AE represents averaging ensemble, WAE represents weighted average ensemble, and BE stands for blending ensemble mode

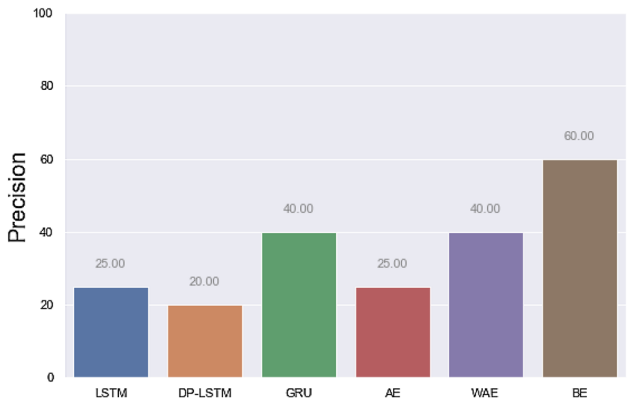

Fig. 7 Precision comparison

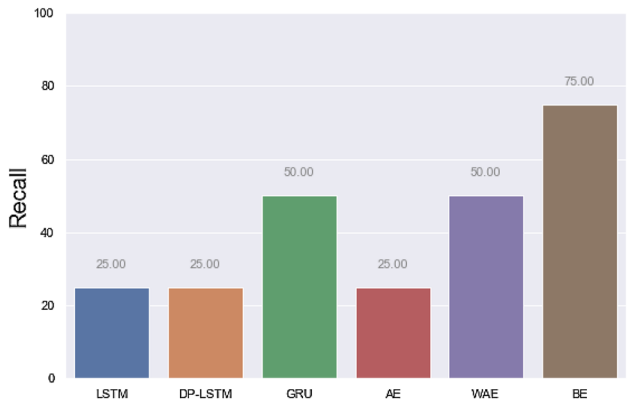

Fig. 8 Recall comparison

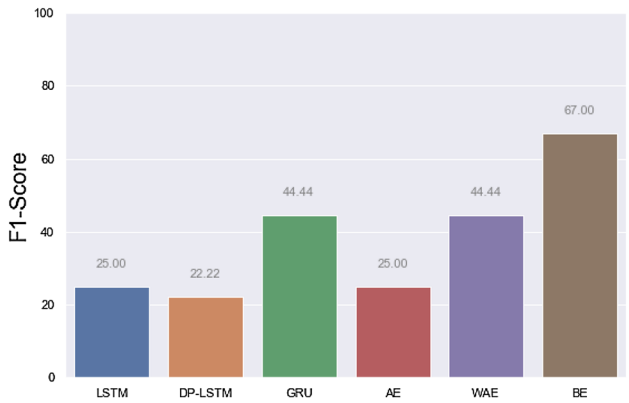

Fig. 9 $F1$-score comparison

with GRU, averaging, weighted average ensemble models. As the results have been shown in the above table, the blending ensemble model improves the results in every evaluation metric compared with previous work and the test models.

As shown in Fig. 6, the blending ensemble model reduces MSE from 438.94 to 186.32, up to 57.55% reduction compared with baseline LSTM. When compared with the DP-LSTM model, the MSE reduces about 43.7% and reduces MSE by 25.27% compared with the test GRU model. The averaging ensemble and weighted average ensemble MSE results are very close. The blending ensemble model reduced the MSE around about 20% compared with the averaging ensemble and weighted average ensemble model.

In terms of precision (see Fig. 7), the blending ensemble model increased the accuracy percentage of at least 20% compared with GRU and weighted average ensemble model and up to 40% when compared with the DP-LSTM model.

In terms of recall (see Fig. 8), the blending ensemble model increases the accuracy by 50% compared with the LSTM, DP-LSTM and averaging ensemble model, and increases accuracy percentage by 25% compared with the test GRU and weighted average ensemble model.

In $F1$-score comparison (see Fig. 9), the blending ensemble model also significantly improved accuracy percentage. The blending ensemble model increased the accuracy per-

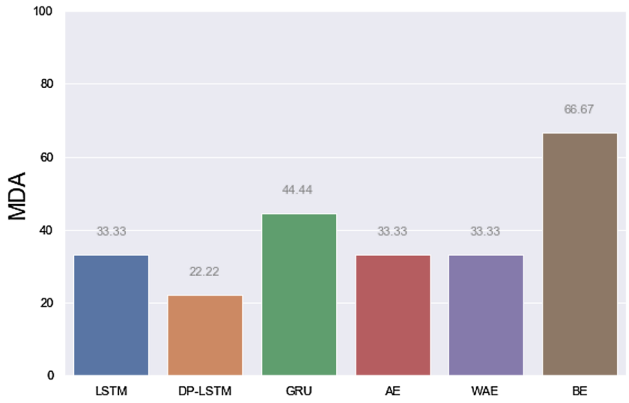

Fig. 10 MDA comparison

centage by 44.78% compared with DP-LSTM, 42% compared with LSTM and averaging ensemble model, and 22.56% compared with GRU and weighted average ensemble model.

Moreover, we also use the stock price fluctuation (positive or negative movement directions) to calculate the movement direction accuracy MDA of all the models in predicting future stock movement directions (see Fig. 10).

On the test dataset predictions, the LSTM, averaging ensemble, and weighted average ensemble model movement accuracy are around 33.33%; DP-LSTM prediction average movement accuracy is around 22.22%. The GRU model prediction average movement accuracy is around 44.44%, and the blending ensemble model achieves approximately 66.67% movement accuracy.

In Fig. 11, the plot showed the entire data included training data, validation data, and testing data. As we can see, the stock price is very unsteady, and the stock's float is very large. However, we can see the prediction results of the blending ensemble model are more closer to the actual stock, and the pattern of the prediction line is more identical to the actual stock. Our experimental results clearly give us the insight that ensemble deep learning with more different data sources handled by different neural network architectures can predict results more accurately in general.

7 Conclusion

This paper proposes a novel deep learning architecture to predict stock prices by combining multiple recurrent neural networks to form a blending ensemble learning model. The blending ensemble model captures the changes in time series influence on the stock price. To evaluate the effectiveness of the method, we have conducted experiments on different recurrent neural network models and set up the same environment and configures to compare with the blending ensemble model. Several evaluation metrics like MSE, MPA, precision, recall, $F1$-score and MDA are used to evaluate the performance of the blending ensemble model. Our experimental results show clearly that the blending ensemble model outperforms all previous methods in every category. According to one of the global tech news organizations such as Nikola NEWS, if a machine learning or deep learning model can reach 60% accuracy on predicting stock movement directions, it can deliver solid returns [20]. Our model can reach up to 67% in MDA on the hold-out test dataset, which strongly suggests that using deep learning models to analyze and predict future stock price or movement direction can be a valuable source to stock investment companies and individual investors. For other performance metrics, the blending ensemble model achieves up to 57.55% improvement in MSE; in predicting the future direction of the stock market, the blending ensemble model improves precision accuracy by 40%, recall accuracy percentage by 50%, and $F1$-score by 44.78%, and movement direction accuracy by 33.34%, compared with the best results in the literature. These results demonstrate that our novel blend multiple different recurrent neural networks can significantly improve the previous best model's robustness and prediction results. The proposed model is different from the existing model in that most of the existing models use a single neural network to predict the stock price. However, the stock price is very volatile and has many factors that can affect the stock price. It is difficult for a single model to predict future price trends accurately because a single model often cannot learn all characteristics of the data. The significant contribution of this paper is the proposed model, which uses multiple models to complement each other, and different models can learn different characters of the data, which can also reduce noise. Moreover, our findings open doors for more sophisticated ensemble deep learning models and using more complex data sources for stock prediction in the future. It is our hope that these new models will truly better assist stock investors in making the correct decision in a real-world situation.

8 Future research

We believe that there is a lot of improvement space over the current blending ensemble model and input data sources. We may use many different ways to improve our current work. Below are possible future research directions.

1. There is a good chance that the current results could be improved by fine-tuning the hyperparameters, increasing the size of the training dataset, and considering other data sources such as the 10-K annual report.
2. Understanding the mechanisms of our prediction can provide more insights into our prediction results. We plan to

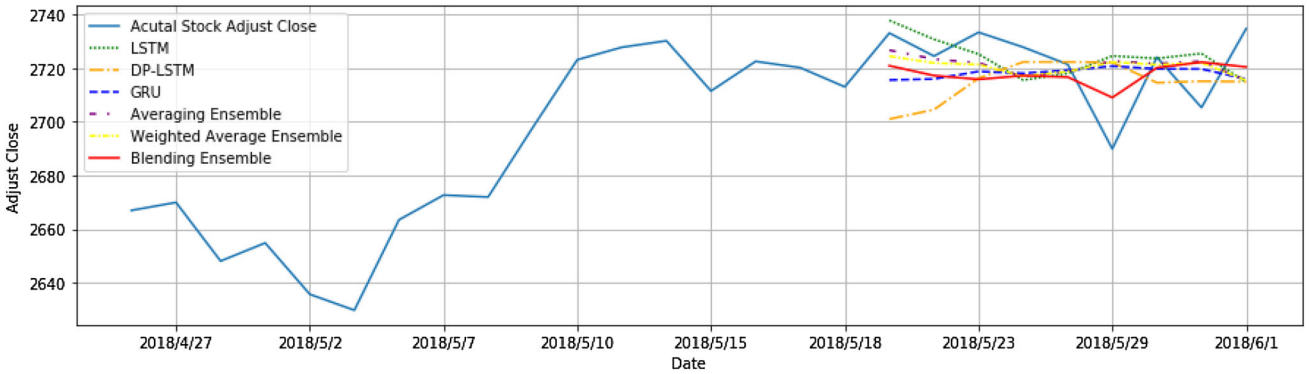

Fig. 11 Prediction result of the blending ensemble learning model compared with LSTM, DP-LSTM, GRU test model, averaging ensemble, and weighted average ensemble model

understand our prediction better through rule generation [21] and use other machine learning technologies such as Clustering SVM [22] and Clustering Deep Learning [23] to improve our results.

3. We also plan to expand the current model by adding LSTM with attention and possibly combining more models to mining different data sources to achieve better prediction results. Especially in level 1, we may employ more parallel models to complement each other.
4. We would like to deploy a reinforcement learning model as the second-level model to explore the area of stock market prediction further. Reinforcement learning is believed to get an optimal policy for a specific problem so that the reward or profit obtained under this strategy could be a better choice. The policy is actually a series of actions that are basically sequential data [24].
5. Also, fuzzy logic systems have been used in many applications, such as wireless network routing [25]. We will introduce fuzzy deep learning into our learning and prediction models in the future [26] since many news items are fuzzy in terms of their positive or negative impacts.
6. We may also dynamically change the historical window size based on the type of news. Some news has long-lasting impact such as housing costs, while others live a very short life such as a sudden disaster. We even can have different window sizes for different news types. Studying which news type has a long-term effect and how long it lasts is probably more a psychological study than a computer science problem. But combining these discoveries in psychology, economy and political science may help our prediction do a better job.

As artificial intelligence becomes more powerful, computer scientists are also constantly developing new models to analyze and predict the stock market, hoping to provide more reliable and more precise stock information to investors. Although our study is preliminary, it is a good start for more interdisciplinary research in this exciting area.

Acknowledgements The authors would like to thank Yinchuan Li for providing the datasets for this research and for his generous support and Sean Cao for his helpful comments.

References

1. Islam, M.D., Salam, M., Hasan, M.D.: Factors affecting the stock price movement: a case study on Dhaka Stock Exchange. Int. J. Bus. Manag. **10**, 253 (2005). https://doi.org/10.5539/ijbm.v10n10p253
2. Choi, H.: Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model. arXiv:1808.01560 (2018)
3. Kordonis, J., Symeonidis, S., Arampatzis, A.: Stock price forecasting via sentiment analysis on twitter. In: PCI '16 (2016). https://doi.org/10.1145/3003733.3003787
4. Akita, R., Yoshihara, A., Matsubara, T., Uehara, K.: Deep learning for stock prediction using numerical and textual information. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–6 (2016). https://doi.org/10.1109/ICIS.2016.7550882
5. Nelson, D., Pereira, A., Oliveira, R.: Stock market's price movement prediction with LSTM neural networks. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1419–1426 (2017). https://doi.org/10.1109/IJCNN.2017.7966019
6. Hu, Z., Liu, W., Bian, J., Liu, X., Liu, T.: Listening to chaotic whispers: a deep learning framework for news-oriented stock trend prediction. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM (2018). https://doi.org/10.1145/3159652.3159690
7. Li, X., Li, Y., Yang, H., Yang, L., Liu, X.: DP-LSTM: differential privacy-inspired LSTM for stock prediction using financial news. arXiv:1912.10806 (2019)
8. Pinheiro, L., Dras, M.: Stock market prediction with deep learning: a character-based neural language model for event-based trading. In: ALTA (2017)
9. Ding, X., Zhang, Y., Liu, T., Duan, J.: Using structured events to predict stock price movement: an empirical investigation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1415–1425. Association for Computational Linguistics (2014). https://doi.org/10.3115/v1/D14-1148

10. Hutto, C., Gilbert, E.: Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: ICWSM (2014)
11. Kirlić, A., Orhan, Z.: Measuring human and Vader performance on sentiment analysis. Invent. J. Res. Technol. Eng. Manag. (IJRTEM) **1**, 42–46 (2017)
12. Connor, J., Martin, D., Atlas, L.: Recurrent neural networks and robust time series prediction. IEEE Trans. Neural Netw. **5**(2), 240–254 (1994). https://doi.org/10.1109/72.279188
13. Krstanovic, S., Paulheim, H.: Ensembles of recurrent neural networks for robust time series forecasting. In: SGAI Conference (2017). https://doi.org/10.1007/978-3-319-71078-5_3
14. Chung, J.,Çaglar, G., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555 (2014)
15. Dietterich, T.: Ensemble methods in machine learning. In: Multiple Classifier Systems, pp. 1–15. Springer, Berlin (2000). https://doi.org/10.1007/3-540-45014-9_1
16. Dietterich, T.: Ensemble Learning, The Handbook of Brain Theory and Neural Networks, 2nd edn. The MIT Press, Cambridge (2002)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
18. Gers, F., Eck, D., Schmidhuber, J.: Applying LSTM to time series predictable through time-window approaches. Neural Nets WIRN Vietri-01, pp. 193–200. Springer, London (2002). https://doi.org/10.1007/3-540-44668-0_93
19. Cho, K., Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder–decoder approaches. https://doi.org/10.3115/v1/W14-4012. arXiv:1409.1259 (2014)
20. Adusumilli, R.: Predicting stock prices using a keras LSTM model. NikolaNews (2019)
21. He, J., Hu, H., Harrison, R., Tai, P., Pan, Y.: Rule generation for protein secondary structure prediction with support vector machines and decision tree. IEEE Trans. NanoBiosci. **5**, 46–53 (2006). https://doi.org/10.1109/TNB.2005.864021
22. Zhong, W., He, J., Harrison, R., Tai, P., Pan, Y.: Clustering support vector machines for protein local structure prediction. Expert Syst. Appl. **32**, 518–526 (2007). https://doi.org/10.1016/j.eswa.2005.12.011
23. Zhong, W., Gu, F.: Predicting local protein 3D structures using clustering deep recurrent neural network. IEEE/ACM Trans. Comput. Biol. Bioinform. (2020). https://doi.org/10.1109/tcbb.2020.3005972
24. Sutton, R., Barto, A.: Reinforcement learning: an introduction. IEEE Trans. Neural Netw. **16**, 285–286 (2005). https://doi.org/10.1109/TNN.1998.712192
25. Liu, H., Li, J., Zhang, Y., Pan, Y.: An adaptive genetic fuzzy multi-path routing protocol for wireless ad-hoc networks. In: Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-assembling Wireless Network, pp. 468–475 (2005). https://doi.org/10.1109/SNPD-SAWN.2005.12
26. Mudiyanselage, T., Xiao, X., Zhang, Y., Pan, Y.: Deep fuzzy neural networks for biomarker selection for accurate cancer detection. IEEE Trans. Fuzzy Syst. (2019). https://doi.org/10.1109/TFUZZ.2019.2958295