2017 International Conference on Tools with Artificial Intelligence

Time-weighted LSTM Model with Redefined Labeling for Stock Trend Prediction

Zhiyong Zhao, Ruonan Rao, Shaoxiong Tu
Shanghai Jiao Tong University
Shanghai, China
Email: {zhaozhiyong, rnrao, tushaoxiong}@sjtu.edu.cn

Jun Shi
Shanghai Rongshi Investment Management Co. Ltd.
Shanghai, China
Email: {shijun}@rsfortune.com

***Abstract*—Various techniques have been applied to predict stock market trends. However, the results are not quite satisfactory due to stock market's complexity. Many approaches either lack a clear and reasonable definition of *trend* or neglect the uniqueness of *time* attribute in stock data, treating them like other attributes, and use one-size-fits-all models to solve such a typical time-series problem. In this paper, we attempted to exploit the time attribute of stock data to improve prediction accuracy. Firstly, instead of treating data indiscriminately, we used time weight function to carefully assign *weights* to data according to their temporal nearness towards the data to be predicted. Secondly, the stock trend definitions were formally given by referencing financial theories and best practices. Lastly, Long Short-Term Memory (LSTM) network was customized to discover the underlying temporal dependencies in data. The trials of different time-weighted functions showed that the relation between the importance of data and their time-series is not constant. Instead, it falls within linear and quadratic, roughly a quasilinear function. Equipped with the time-weighted function, LSTM outperformed other models and can be generalized to other stock indexes. In the test with CSI 300 index, we achieved 83.91% in accuracy when fed with the redefined trends.**

I. Introduction

Stock market is a dynamic, non-linear, and high-dimensional system that involves in various factors such as economy, policies, psychology, etc. which makes it difficult to predict. Some theories, for example, Efficient Market Theory(EMH) [1] states that assets, in a information free-flowing market, always trade at their fair value and reflect all available information which means the stock price are only affected by new information. Therefore, it seems impossible to outperform the market by stock selection or market timing.

The critics, however, claim the market is less efficient on average stock [2]. Furthermore, researchers have found that although it is difficult to predict the movement of stock in the short-term, there is still predictability over the long run [3]. Also, if the market does move randomly all along, there should be no difference in profiting between experienced traders and the novice which is contrary to the reality. Inspired by this, lots of technicians and financial experts have sought their way in discovering the long-term pattern of stock index trends.

In general, there are two main methodology in predicting stock movements: fundamental analysis and technical analysis. Fundamental analysis aims to assess the intrinsic value of the stock by conducting a thorough study about the overall factors that affect the company's stock, such as assets, liabilities, GDP, etc. [4]. It claims that the market may incorrectly price a stock in the short run but will eventually correct the mistake to its real value. The supporters usually purchase the wrongly priced stock and then wait for repricing, profiting from the difference between the real value and market price. Technical analysis is the methodology that primarily studies past market data to forecast the price trends [5]. Users of the methodology believe that all available information are reflected in prices, by studying the pattern of the past, they could predict the movements in future. Approximately 90% of major stock traders [6] are using this method because they tend to capture the trends.

Fundamental analysis requires more expertise and sophisticated insights in securities evaluation which is beyond most ordinary technicians. Therefore, they resort to technical analysis. According to the discussion above, our work mainly focus on the relatively long-term trends on average stock, i.e. stock indexes. In reality, plenty of researchers have strived for forecasting the trends of stock indexes with a wealth of sophisticated means. Some of them adopt news-driven theory, mining the news for trader's sentiment regarding stock [7] while others forecast their results principally relying on the efficient factors selection. The results, however, are not satisfactory.

Among those models, the characteristic of time in stock data was underemphasized. Hence, in this paper, we concentrate on exploiting the time attribute of stock data to improve the performance of prediction. Firstly, we assume the importance of data at different time point varies such that different weights should be assigned to them. In practice, most recent data play more important roles in the near future movements. In another word, weights of data increase along time. Secondly, since the problem is about trends prediction, it is rather vital to define trends both reasonably in financial theory and practically in model implementation. Dow Theory and Wave Theory are referenced in the trend modeling and will be detailed illustrated in section.III. Lastly, particular approaches should be utilized to handle such time-series data.

In a nutshell, the contributions of this work are as follows:

1) discovered the non-linear relation between the importance and time point of stock data;

2375-0197/17/31.00 ©2017 IEEE
DOI 10.1109/ICTAI.2017.00184

2) proposed an approach to model trends inspired by financial theories;
3) achieved relatively high prediction accuracy by utilizing time-weighted LSTM with redefined trends.

II. Theoretical foundation and Related Work

A. Dow Theory and Wave Theory

Dow Theory is the cornerstone of technical analysis. It has strong ability to capture the longer-term trends which is preferred by traders who tend to invest in the long haul. An important principle in the theory is that the averages discount everything which means the markets reflect all possible factors that affects overall supply and demand, even though some emergent events occur, their effects will be assimilated into the price almost instantaneously. Besides, it claims the market moves in trends which has three parts, primary, intermediate and minor. Each small trend is part of the big trend and share some similar characteristics. Once the trends formed, they are more likely to continue.

Wave Theory [8] is often seen as the extension and improvement of Dow Theory. It was originally applied to the stock index where the market roughly follows a rhythm of a five waves advance followed by a three waves decline. The eight waves, including five up and three down, form the smallest unit for trends. The main trends are pushed by the former waves while the latter do the adjustment. The length and height of each wave may vary, but the basic form still remains.

In summary, Dow Theory provides supports for studying historical data and classifying trends while Wave Theory gives specific guidance on trends identifying. Both of them are important for technical analysis and their insights will be incorporated in our modeling.

B. Machine Learning in Stock Prediction

As machine learning flourishes in a wealth of areas recently, many researchers have been attempting to use learning algorithms in solving stock prediction problem. George S. Atsalakiset. al. [9] surveyed over 100 related published articles that focus on neural and neuro-fuzzy techniques in forecasting stock markets. The survey showed that the soft computing techniques are viable candidates to capture stock market nonlinear relations with not necessarily prior knowledge of input data. About 30% of the publication used closing price as the input data in stock or index forecasting. In terms of methodology, roughly 60% of the surveyed articles use feedforward neuron network(FNN) or recurrent neuron network(RNN) which outperform conventional models in most cases. J. G. Agrawal, al. [6] showed state-of-the-art in stock prediction techniques in which fundamental analysis and technical analysis were thoroughly compared. It suggested incorporating domain knowledge into the system might help in achieving better performance.

Xiao Ding al. [10] also in favor of that the stock market are event-driven. With events extracted from news, their deep convolutional neural network can achieve 65.08% accuracy on S&P 500 Index which is 6% improvements than the state-of-the-art baseline model using Support Vector Machine(SVM).

In essence, stock data are time-series multidimensional vectors. More appropriate approaches should be taken in processing this special type of data. According to Martin Langkvist[16], deep neural network outruns other models in many time-series problems because it provides better representation and classification compared to shallow approaches. Complex model are preferred in dealing with multivariate problems. Long short-term memory network(LSTM) [11] is a specific recurrent neuron network which is good at tackling complex long-term dependency tasks well and is resistant to vanishing gradient problem. Recently, there are a great deal of successful applications, for instance, in Alex's work [12], LSTM was used to generate highly realistic handwriting with many styles. Google deployed two-layer deep LSTM in large scale speech recognition [13] and their model exceeded the state-of-the-art work. Since LSTM performs well in time-series problem, they might yield better advancement in stock prediction than conventional ones. Kai Chen al. [14] proposed a LSTM-based approach for stock returns prediction with 13% improvement of accuracy compared with random method, but they didn't use it for stock movements prediction. Ryo Akita al. [15] converted newspaper articles to distributed representations for LSTM training, also, not directly on past transaction data. The model of Luca Di al. [16] indicated 72% accuracy can be achieved for 5-days prediction interval with LSTM and dropout, they trained only on plain price series data.

However, few of the publications actually utilized LSTM in modeling stock index trends, nor did they treat stock data at different stages differently. Unlike those models, this paper will focus on predicting the moving directions of stock index and put more emphasis on differentiating the importance of stock data at different time.

III. Problem Clarification and Solution Design

This paper aims to predict stock movement trends which can be divided into two subproblems: 1) What is *trend*? 2) How does the *predictions* being made? The following sections centre around the two questions. We firstly quantify trend in a formal way then propose practical model to tackle it.

A. Trend Definition

Prices move in trends is the most basic principle in technical analysis of stock market. Generally, markets do not move linearly, instead, their trace are often a series of *zigzags*. Though the essential concept - trend is widely acknowledged, the interpretation varies. In this section, we put forward a way to quantify trend. Elloit described the price movements figuratively as *waves*, the stock has high price and low price which is similar to wave that has *peaks* and *troughs*. An interesting observation is that peaks and troughs are always coming in pairs. Specifically, before the market resume moving towards the original direction, up or down, it tends to move to the opposite direction to some degree. The behavior of such countertrend is specially described by a financial term, *retracement*. As Figure.1 shows, the percentage of retracement falls into a certain range. 50% retracement is the case best known to technician, while 33% is considered as the minimal portion and 66%, i.e. two-thirds the maximum which is often seen as the signal of stop-loss because the trends are more likely to reverse under such circumstance. Obviously, retracement is an important part of trend and can be used in identifying trends. In Dow Theory, trends are divided to

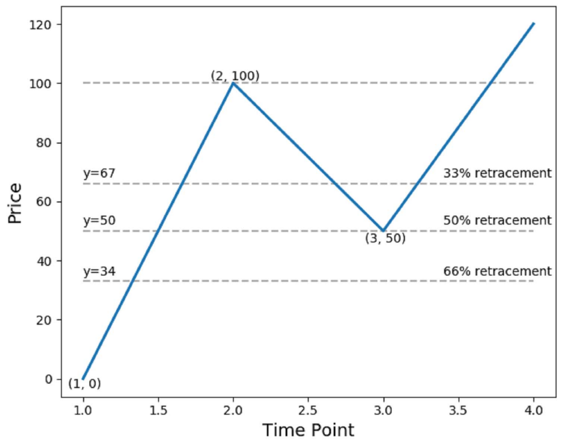

Figure 1. An example of 50% retracement. Instead of continuing to increase from (2, 100), the price tends to roll back half of the previous change.

three categories by direction: *uptrend*, *downtrend* and *sideways trend*, i.e. non-trending. An uptrend would be characterized as a sequence of successively higher *peaks* and *troughs*; a downtrend, oppositely, has a sequence of declining peaks and troughs; the sideways trend means the price movements have no clearly recognizable form but oscillate in a certain range. This classification is quite intuitive and easy to comprehend, however, the interpretation of it varies from people to people which makes it difficult to implement. Here, we give a formal definition of these three trends for the good of latter study. We use *UP*, *DOWN*, *NO* to shortly stands for uptrend, downtrend and sideways trend, respectively. To begin with, we define a peak $P_i = p(t_i)$ as a point which satisfies

$$P_i > p(t) \qquad \forall t \in (t_i - \epsilon, t_i + \epsilon), \qquad \epsilon \geq constant \tag{1}$$

That is to say P_i is a local maximum of price curve. And we can define a trough T_i similarly.

1) UP trend satisfies: a) At least K pairs of peaks and troughs are present; b) each peak should outrun the closest preceding peak; c) the retracement should be less than α of the previous price rise.

We use P to denote peak price value while T for trough price value and the subscript represents its order in the time series. Every trough is coming right after a peak. Therefore, the trend classified as UP should conform to the following definition: For two increasing sequences $\{P_i\}, \{T_i\}$ with size large than $K, i \in \mathbb{N}, i > 0$, satisfy

$$P_i - T_i < \alpha(P_i - T_{i-1}) \tag{2}$$

Figure.2.a shows an example of UP trend, there are 3 pairs of peaks and troughs with $P_3 > P_2 > P_1$.

2) DOWN trend satisfies: a) At least K pairs of peaks and troughs are present; b) each peak should be lower than the preceding one; c) the retracement should be greater than α of the previous fall. Similar to UP trend, its formal definition is as follows: For two decreasing sequences $\{P_i\}, \{T_i\}$ with size large than $K, i \in \mathbb{N}, i > 0$, satisfy

$$P_i - T_i > (1 + \beta)(P_i - T_{i-1}) \tag{3}$$

As depicted in Figure.2.b, P_1, P_2, P_3 are in descending order and the degree of decline is greater than the defined percentage of tracement.

3) NO trend: the trends not belonging to UP or DOWN. When $\{P_i\}$ is not an ordered sequence or maybe the retracement is too large or small, it goes to the NO trend category. It is noteworthy the market does not always show a distinct trends, instead, for at least half of the time it either move in a flat pattern or volatile, fluctuating extremely in a way that most techniques are unable to capture. The flat pattern often takes place when the supply and demand are in a relatively balanced state. As showed in Figure.2.c, the prices at peaks and troughs are roughly the same. Not taking any action would be the recommended strategy in this trendless environment. This also applies to the extreme situation where the market takes random walks before returning to the main trend. We here categorize such case as nontrend as well. Formal definitions of trends are presented above, the best practice of α, β and K will be explored in the experiment section.

Algorithm 1: Trend Labeling Algorithm

```
Input: all past data $D$ and its size $S$
       retracement for UP, DOWN trend $\alpha$, $\beta$
       number of pairs of peak and trough $K$
Output: $Label_{list}$
 1: $Label_{list}$ = []
 2: while $i < S - 1$:
 3:     while $j < K$:
 4:         cursor = $i$
 5:         while cursor $< s - 1$ and $D[i] \leq D[i+1]$:
 6:             cursor += 1
 7:         $p$ = cursor
 8:         while cursor $< s - 1$ and $D[i] \geq D[i+1]$:
 9:             cursor += 1
10:         $t$ = cursor
11:         retrace = ($D[p]$ - $D[t]$) / ($D[p]$ - $D[i]$)
12:         if retrace $< \alpha$:
13:             UP += 1
14:         else if retrace $> 1 + \beta$:
15:             DOWN += 1
16:         else
17:             NO += 1
18:     trend = 1
19:     if UP == $K$
20:         trend = 2
21:     if DOWN == $K$
22:         trend = 0
23:     while $i <$ cursor
24:         $Label_{list}.append(trend)$
25:         $i += 1$
```

Algorithm.1 implements the function of labeling data to different trends according to the definition in this section. From

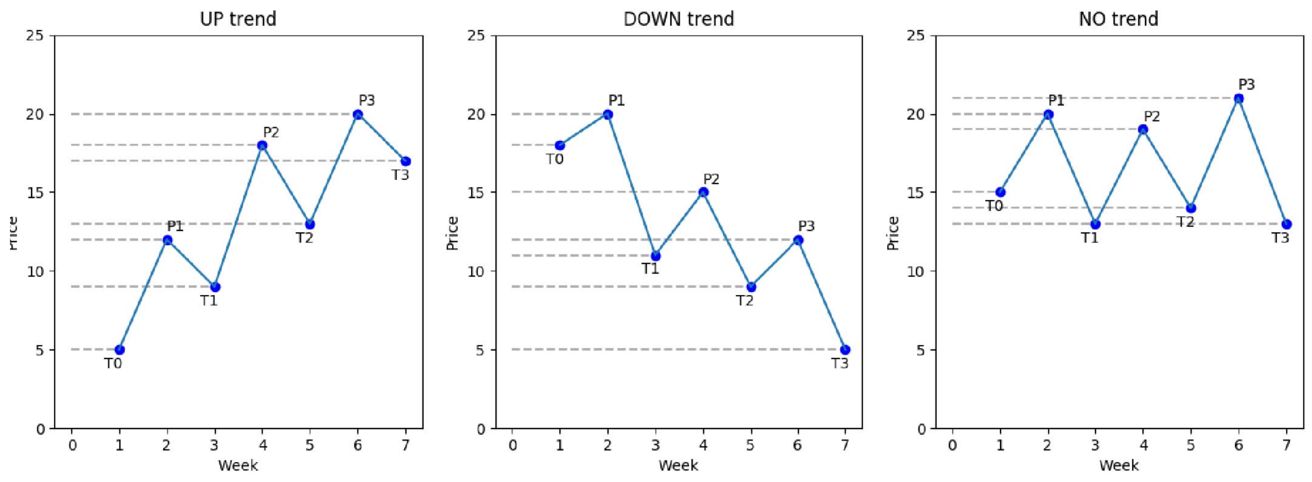

Figure 2. Different forms of trend. a) example of stock movement in UP trend, $\alpha = \frac{1}{2}$; b) example of DOWN trend, $\beta = \frac{2}{3}$; c) example of NO trend

line 4 to 10, it searches peaks and troughs in data series, then identifies specific trends from line 11 to 17, lastly, labels them correspondingly.

B. Prediction Model

Our model mainly focus on exploiting the intrinsic *time* nature in time-series stock data. The exploration primarily covers three aspects. First of all, we believe that the importance of data varies from time to time, accordingly, they should be treated differently. The straightforward way to achieve this is assigning different weight to data at different time, therefore time-relevant weight functions are defined in the following subsection. Secondly, since the stock data is time-series, we utilized LSTM, a type of neuron network excelling in solving time-series problems, to discover the possible temporal dependence in the complex data. Lastly, we carefully chose the time frame for training and predicting backed by classic theories in finance. Detailed illustrations are as follows.

1) Time Relevant Importance of Data: We hypothesize that data of different point of time have different effects on the results of predictions. Usually, the more recent data are more likely to have more influence on the price movement. Hence, instead of treating historical data indiscriminately like many other methods, we introduce a weight function $\omega(t)$ to indicate the varying importance of data along time. Suppose every instance in our training data set covers K time steps, let x_t denote the data at tth step. We define $\omega(t)$ as the weight function which satisfies

$$\int_0^K \omega(t)dt < \infty \tag{4}$$

Thus, for every x_t, it has an associated weight coefficient $\omega(t)$ with which are fed into the model. Experiments will be carried out to search for the most fitting weight function with relatively small variance. Then the mean value of weight $\bar{\omega}_t$ will be used for the corresponding time step data.

Generally, the market will reflect the effects of an event immediately, for instance, the stock price of a company might go up quickly when it raised several million in the latest financing. However, as time goes by, the importance of that event will fade away since the market gradually discounts its effects. The scenario well demonstrates the idea that nearer data may have more powerful influence. It can be reflected in the Equation.4, in the time slot from 0 to K, the greater the t, the nearer it is to the period to be predicted, the weight $\omega(t)$ of tth data should be larger accordingly. Thus, attempts will be made to trial various monotonically increasing functions that fit the time-varying importance of data. What we are doing here is incorporating insights into model because preprocessing by prior knowledge might effectively help in model training.

2) Time-series Network: RNN distinguishes itself in sequence data modeling. It differs from FNN in its self-connection which acts as the channel for historical output impacting the present. Because of the unique structure, RNN can remember the past by persisting previous data in its internal structure which often called memory cell. Then the historical data might play a part in the possible future. The

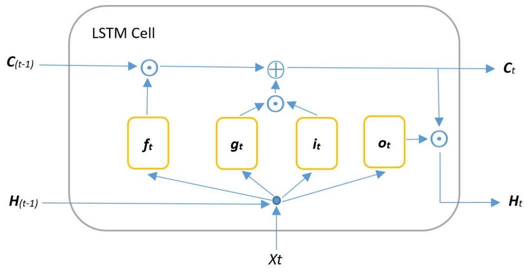

Figure 3. Interal structure of a single LSTM memory cell. New information is added to the cell by $g_t \odot i_t$ while previous state is eliminated by forget gate f_t. The cell output short-term state H_t and update long-term cell state C_t at each time step

problem of RNN is that the complexity grows exponentially when the dependency spans over long periods. LSTM does well in solving long-term dependency problem due to its special structure. It has three gates: *input*, *output* and *forget* gate for controlling the influence of data on the state of each neuron. It enables the network to take in important data while drop insignificant information. There are plenty of LSTM variants, they are slightly different in the internal structure.

In this paper, we used the structure similar to Sak's conventional LSTM model [17] without peephole connections. It is specified as follows:

$$i_t = sigm(W_{xi}^T x_t + W_{hi}^T h_{t-1} + b_i) \tag{5}$$
$$f_t = sigm(W_{xf}^T x_t + W_{hf}^T h_{t-1} + b_f) \tag{6}$$
$$o_t = sigm(W_{xo}^T x_t + W_{ho}^T h_{t-1} + b_o) \tag{7}$$
$$g_t = tanh(W_{xg}^T x_t + W_{hg}^T h_{t-1} + b_g) \tag{8}$$
$$c_t = i_t \odot g_t + f_t \odot c_{t-1} \tag{9}$$
$$h_t = tanh(c_t) \odot o_t \tag{10}$$

where i, f, o denote input, forget, output gate respectively, b_* are bias, W_* represents weight matrices, for example, W_{xi} is the weight for the input gate's connection to the input vector x_t, W_{hi} is the weight matrix for previous state h_{t-1}. As can be seen, the gates use logistic activation function $sigmoid$ whose output value ranges from 0 to 1. They act as the controller for information flow through element-wise multiplication operation $\odot$ with original or new-coming data. As Figure.3 shows, at each time step, the long-term state c_t was generated by dropping some pervious memories through forget gate and adding some current memories selected by input gate. The long-term state then is squashed by tanh function and filtered by the output gate to short-term state h_t which in practice equals to the output for this time step. The special structure of LSTM enables it to partially store new information and selectively erase previous one which performs well in dealing with long sequence problem. Here, we set forget gate bias to 1.0 inspired by the best practice in Rafal al.'s work. Further exploration in hyperparameters such as number of network layers, neurons in each layer, learning rate etc., would be showed in experiment section.

3) Prediction Periods Choosing: In classic Dow Theory, trend usually has three scales: *primary*, *intermediate* and *minor*. The primary trend is longer than a year in effect, the intermediate trend is defined as three weeks to several months while the minor trend is usually less than two weeks. Besides, the smaller scale trend is a portion of the larger one, for example, an intermediate trend comprises several short-term trends. In reality, most technical approaches focus on the study of intermediate trends. Because it might be easy to find the proper trading time with a broad view of included short-term trends and also beneficial for tracing primary trends with the study of adjacent intermediate trends.

Guided by this rule of thumb, our prediction model is primarily focus on intermediate trends. Following the trend definition in section.III-A, we took a period of data as a *calculation interval*, say 7 weeks like Figure.2 showed, then labeled the interval as 'UP' in this case. Next, couple of such intervals' data with labels will be fed to the model to make a prediction on the trend in the next interval. The idea is demonstrated in Figure.4, the example uses the data of five calculation intervals i.e. 35 weeks as the input sequence aiming to predict the trend in the next stage. At each time step, the model takes in 1 interval's data, do the calculation then output. Particularly, the outputs before the last time step are ignored because the five intervals are grouped together as a roughly major trend to generate the next possible intermediate trend, only the final output is significant. In addition, to eliminate the effects of daily perturbations, the *average* close price of a week will be used to represent the input of the week. After that, the algorithm of trend labeling will be ran on the smoothed data. The whole architecture of our model is depicted in Figure.5. It shows how the prediction model works. At the beginning, data are divided and labeled according to the methods we discussed above. Next, a customized time-varying weight function will act upon the input data, assigning weight to data at corresponding time point before feeding to LSTM neuron network. Then the model starts training and adjusts its parameters by calculating the loss between predicted values and true labels. The whole process will be repeated for enough iterations until desirable accuracy is obtained.

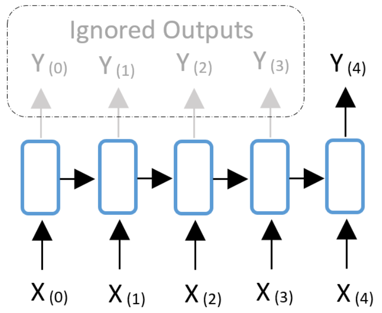

Figure 4. Example of the unrolled sequence of one 5-timestep LSTM neuron. $X_{(t)}$ denotes the input at time step t, $Y_{(t)}$ is the output. First four time steps' outputs will be ignored, only the last output is needed for the prediction.

IV. Experiments and Results

A. Data set and Evaluation Method

Fifteen years' daily transaction data of five main stock indexes in China: SSE Composite Index (000001.SS), SSE 50 (000016.SS), CSI 500(000905.SS), CSI 300(000300.SS) and SZSE Composite Index(399001.SZ) was used in the experiments. They cover the daily *close* price from 2002 to 2017, 3733 days in total. These data are publicly available in many financial data providers such as Yahoo! Finance.

As for evaluation, *accuracy* of prediction was used to measure the performance of models. It was calculated by the number of correct hits divided by total predictions. To validate the correctness of results, several rounds of experiments were carried out. It's noteworthy classical *cross-validation*

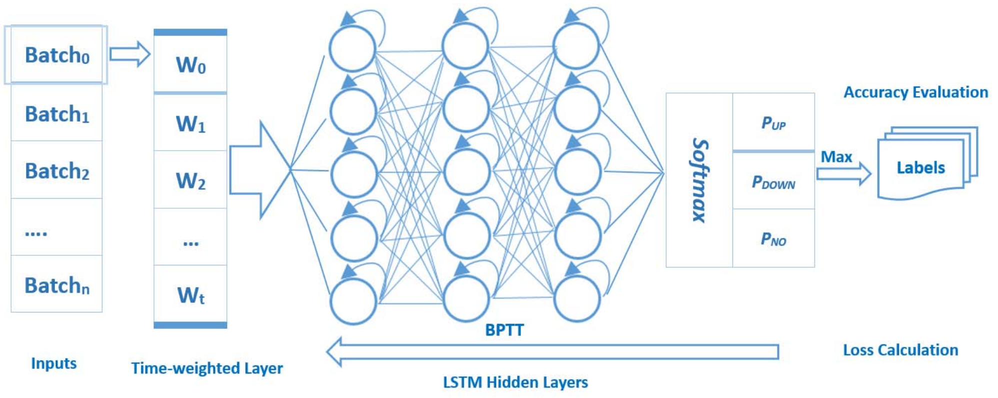

Figure 5. Architecure of prediction model. A time-weighted layer is added to assign different weights for data at different time before entering the training model. The output of the model is a probability distribution over three trends.

techniques, K-Fold for example, assume the samples are independent and identically distributed which usually does not apply on time series data and would inevitably yield poor generalization ability. To achieve this, we implemented cross-validation function similar to TimeSeriesSplit in scikit-learn [18]. The above section clarified data source and evaluation method, the experiments were conducted in four aspects: 1) how the form of waves and retracement affects the result; 2) which time-weighted function best describe the time-varying importance; 3) how well the model performs; 4) whether it can be generalized to other markets.

B. Trends Form and Retracement

Different pairs of peaks and troughs were tested, we found that it yields better result when pair number K equals to 1. It probably because the single pair of peak and trough is the smallest unit for any trend and any other larger forms could be composed by groups of it. Also, the case where β equals to α outperforms others. As for retracement, in practice, the reasonable percentage of retracement falls in the range from 33.3% to 66.7%. This empirical rule covers over 75% cases in our statistics about retracement of indexes. Therefore, several set of experiments were conducted to test the proper retracement in this search space. A baseline RNN model was used and its prediction accuracy was used to measure the performance of retracement percentage. The results were recorded in Table.I. It shows a relatively higher precision will be obtained when the percentage of retracement is set between 0.53 and 0.63. Hence, we set golden section i.e. 0.618 which is recommended in Wave Theory as the retracement percentage for the following experiments.

Figure.6 shows the comparison between two labels with different retracement. The orange line 'trend's retracement is 0.618 while 'label' has no retracement which is the common way of defining trend in other methods. 'trend' can be seen as trend complying with our definition but different from other models. Obviously, whenever subtle changes happen, 'label' become more volatile than 'trend' that makes it more difficult to discover the underlying pattern. Since the goal of this paper is to capture the main trend in the market, using 'trend' without unwanted noise leads to better results.

TABLE I
COMPARISON OF DIFFERENT RETRACEMENT PERCENTAGE

Data sets	0.33	0.43	0.53	0.63	0.67
SSEC	71.31%	**73.61**%	72.46%	71.64%	71.80%
SSE 50	72.62%	74.10%	**74.92**%	73.44%	73.11%
CSI 300	69.34%	67.87%	**70.49**%	69.34%	68.03%
CSI 500	72.62%	72.62%	73.61%	**74.75**%	73.61%
SZSEC	69.51%	68.69%	67.05%	**69.84**%	69.18%

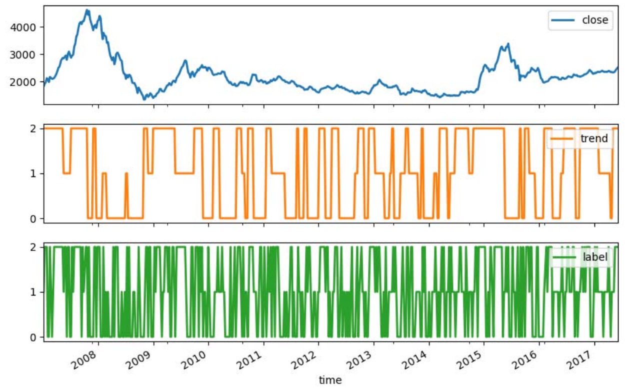

Figure 6. Comparison between trends used in our model with trends in other models. The blue line is the real close price series of SSEC from 2007 to 2017. The orange line 'trend' conforms to the trends definiton in this paper while the green line 'label' is the typical way of defining trend.

C. Time-Weighted Function

To find out the proper time-weighted function, we firstly tested groups of common increasing function in the experiment. Network with different time steps were tried to verify the underlying relation. The results validated our hypothesis that more recent data has more important effects on the prediction. As can be seen from Figure.7, all models with weighted

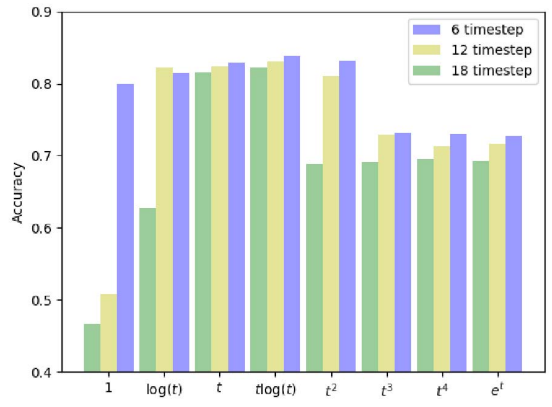

Figure 7. Experiments on different time weight functions. Obviously, models equipped with time-varying importance functions outperform that directly works on plain data.

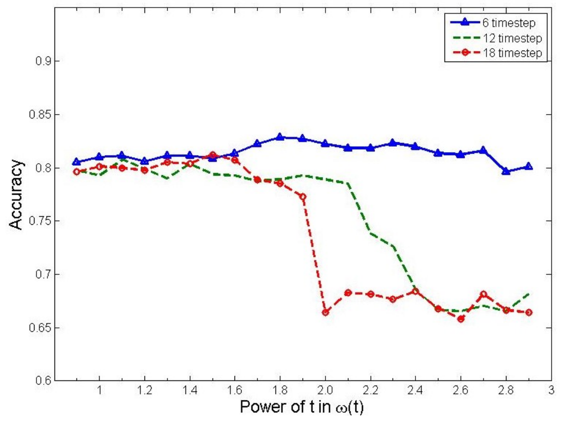

Figure 8. Fine-grained time-weighted function exploration between n and n^3. The longer the time steps, the more distinct difference it shows among time-weighted functions.

function outperform the model with plain data i.e. $\omega(t) = 1$. The longer the time steps, the bigger the difference. Among those functions, $\omega(t) = t\log(t)$ and $\omega(t) = t^2$ gained better results which were not only greater than the linear function, but also exceeded those with fast increment rate such as t^3, $e^(t)$. It implies that the importance of data does not grow infinitely with time, instead, its relation might fall within $\omega(t) = t$ and $\omega(t) = t^3$. To verify the implication, we carried out more fine-grained experiments on the power of t. The results were showed in Figure.8, it once again validated the hypothesis that the time-varying importance dose exist. More specifically, it does not grow exponentially with the temporal nearness towards data to be predicted but follow a relation between linear and quadratic function, especially when the time dependency spans long periods. With this observation, in the following experiments, $\omega(t) = t\log(t)$ will be adopted as the time-weighted function.

TABLE II
TUNING RANGE AND RESULTS OF MAIN HYPERPARAMETERS

	neurons	**layers**	**t_steps**	**l_rate**	**train_steps**
Range	10-500	1-20	1-50	0.0001 0.1	100-40000
Best	320	3	4	0.0024	4500

D. LSTM Model

Our model was built on TensorFlow framework [19] which provides off-the-shelf LSTM cells. We customized the cells according to our needs then implemented ideas rapidly. The model selection is virtually a hyperparameter search process. Dozens of hyperparameters were tested in our experiments to select the proper value. The best value of five key hyperparameters are showed in the Table.II for any interested readers wanting to reproduce it. Increasing the number of neurons did help in improving the prediction accuracy. It added the complexity of the model improves its capability of representation for complex patterns. But going too far will not strengthen but weaken its ability, in our test, 320 neurons worked well for this problem. The number of network layers seemed to not play an important part in the model, the increasing of layers made the result worse. Probably because LSTM model is deep in its nature with self-connection pointing over periods. Time steps stands for the length of effective past data related to the value to be predicted. In this problem, past 4 weeks data produced better results.

E. Comparison with Other Models

To show the effectiveness of time-weighted LSTM model, we conducted groups of experiments among commonly used models. Support vector machine(SVM) was widely used in predicting stock trends and achieved desirable results. In our experiment, we used Grid Search to search for the best parameters of SVM, the logarithmic search space of γ and C are between 2^{-10} to 2^{10}. Ensemble models such as Random Forest(RF) and AdaBoost(AB) are also the good candidates for stock trends classification problem. Therefore, they were taken as the baseline for comparison. In the experiment, RF was implemented slightly different from Breiman's paper [20], instead of letting each classifier vote for a single class but by averaging their probabilistic prediction. The number of classifier is 42 which is the best out of 100 combination. For AB, we used AdaBoost-SAMME [21] as the implementation for this multi-class classification problem. A basic RNN model was also adopted to show the performance difference between LSTM and other time-series models. Additionally, for all models, there were two set of experiments, one was fed with relabeled trend data while the other not. Relabeled trends are data with retracement, to differentiate them, using suffix -R to denote the model with trend retracement, for example, T-LSTM-R represents time-weighted LSTM model with trend retracement.

The results were showed in Table.III. As can be seen, time-weighted LSTM model outperforms other models in almost all data sets and achieve 83.91% in accuracy in the index CSI 300. It is better than common ordinary RNN and other

TABLE III
BEST PREDICTION ACCURACY OF MODELS ON 5 DATA SETS

Data sets	T-LSTM-R	RNN-R	SVM-R	RF-R	AB-R	T-LSTM	RNN	SVM	RF	AB
SSEC	**82.98**%	79.88%	81.02%	80.34%	82.31%	82.61%	73.04%	54.72%	63.03%	67.41%
SSE 50	**83.39**%	75.47%	79.25%	79.16%	81.41%	80.68%	69.11%	51.51%	51.94%	59.63%
CSI 300	**83.91**%	80.50%	81.92%	81.64%	**83.74**%	81.98%	52.57%	45.85%	57.63%	67.69%
CSI 500	81.68%	61.43%	**82.79**%	80.40%	82.33%	81.49%	64.53%	49.28%	58.05%	68.17%
SZSEC	**83.28**%	70.31%	81.13%	78.70%	82.23%	79.57%	77.47%	52.83%	48.37%	60.10%

ensemble models. Overall, the models with trend retracement perform better than those without. This can be comprehend by the case that in a chaos or noisy period, the trend might be wrongly labeled frequently which makes it difficult to forecast. Efficient labeling helps in capturing the main trend while eliminating noises. Therefore, AB-R and RF-R gained at least 12% improvement with the retracement. To further the discovery of time-varying importance of data, we also performed time-weighted function experiments on SVM which was only secondary to LSTM model.

F. Tests in Other Stock Indexes

The experiments were also carried out in other markets to test the effectiveness of the model. In S&P 500, 80.24% accuracy can be achieved while in Dow Jones Industrial Index, 83.21% is the best result. Using time-weighted LSTM model with best parameters, the model shows satisfactory generalization ability in stock indexes.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a time-weighted model for stock trend prediction by leveraging the power of time-series neural network, time-varying importance functions for data and the insights from classic financial theories. Following the guidance of Wave Theory, the form of trend was formally defined. Then the hypothesis that nearer data have more influence on prediction was validated in the experiments. The results implied that there might exist a quasilinear relation between the importance of data and its time series. Above all, an LSTM based model was built to make predictions with the redefined trends which achieved 83.91% accuracy on CSI 300. By utilizing proper time-weighted functions and redefined trends, the method outperformed other models and it was able to be generalized to other stock indexes as well.

For future work, we plan to explore the data's time-varying importance in depth which may benefit time-series problem modeling. Furthermore, trading strategies will be combined with our prediction model to construct a complete algorithmic trading system.

REFERENCES

[1] E. Dimson and M. Mussavian, "A brief history of market efficiency," *European financial management*, vol. 4, no. 1, pp. 91–103, 1998.

ACKNOWLEDGMENT

This work is supported by China National Aeronautical Radio Electronics Research Institute. We would like to thank all the colleagues for their great help.

[2] B. G. Malkiel, "The efficient market hypothesis and its critics," *The Journal of Economic Perspectives*, vol. 17, no. 1, pp. 59–82, 2003.

[3] F. E. T. Burton and S. N. Shah, "Efficient market hypothesis," *CMT Level I 2017: An Introduction to Technical Analysis*, 2017.

[4] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7046–7056, 2015.

[5] J. J. Murphy, *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin, 1999.

[6] J. Agrawal, V. Chourasia, and A. Mittra, "State-of-the-art in stock prediction techniques," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, no. 4, pp. 1360–1366, 2013.

[7] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng, "Exploiting topic based twitter sentiment for stock prediction." *ACL (2)*, vol. 2013, pp. 24–29, 2013.

[8] A. J. Frost and R. R. Prechter, *Elliott wave principle: key to market behavior*. Elliott Wave International, 2005.

[9] G. S. Atsalakis and K. P. Valavanis, "Surveying stock market forecasting techniques–part ii: Soft computing methods," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5932–5941, 2009.

[10] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction." in *Ijcai*, 2015, pp. 2327–2333.

[11] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[12] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

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

[14] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2823–2824.

[15] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in *Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*. IEEE, 2016, pp. 1–6.

[16] L. Di Persio and O. Honchar, "Recurrent neural networks approach to the financial forecast of google assets."

[17] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 2342–2350.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[19] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[20] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] J. Zhu, H. Zou, S. Rosset, T. Hastie *et al.*, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.