**Detection of anomaly Stock Price Based on Time Series Deep Learning Models**

**Abstract:**

anomaly detection is a critical task for financial market, investors, and regulatory authorities, where conventional methods employ rule-based models. With the development of machine learning and deep learning techniques, it becomes more promising to detect anomalous trading behaviors from data. Here we present a deep learning model based on time series LSTM model to detect anomalous behaviors in Chinese stock market. The model is composed of 1dConv-LSTM neurons, which can predict time series stock price data from historical data. We analyzed the price of 14 stocks variations ranging from 2015/01/05 to 2019/12/31 and used univariate and multivariate time series models to generate MAE less than XX consistently. Our model successfully predictes the anomalous price behaviors of XX stock in the range of XXXX. [conclusion] **The proposed method improved MSE to 0.0171 on validation datasets.**

Keywords: Finance, DL, LSTM, Time Series, Anomalous Stock Price

**1. Introduction**:

The Luckin Coffee fraud scandal in the last few months has been spread worldwide. According to the internal investigation, it shows that the fabrication of sales began in April 2019, which included inflating costs and expenses by almost $200 million, as well as booking $300 million in false revenue. After the announcement, Luckin’s stock has slumped 32%. Such a scandal made it harder for other Chinese companies to debut in the United States and thus lose a huge amount of US potential investors due to the untrustworthiness of Chinese companies. Besides, the event magnifies the defect of the Chinese stock market that requires the mechanism to oversee anomalous behaviors of stocks and judge which is a fraud for regulatory authorities. Only if the Chinese authorities establish the mechanism to identify anomalous stock price fluctuations and investigate hoax can investments be secure and the stock market be more stable.

Various detection systems to monitor abnormal stock price changes have been developed. Previously, most of the detection methods rely on a prediction-based method or rule-based method. [prediction-based method or rule-based method explain more]

Since mid-1997, the National Association of Securities Dealers (NASD) in the United States developed Advanced Detection System (ADS) that has been used to monitor trades and quotations in the NASDAQ stock market.The ADS uses two pattern matches to detect abnormal behaviors. The system relies on a rule matcher, which detects predefined suspicious behaviors, and a time-sequence matcher, which looks for temporal relationships between events that exist in a potential violation pattern.

According to Time Series Contextual Anomaly Detection for Detecting Market Manipulation in Stock Market, the author proposed Contextual Anomaly Detection (CAD) method, which aims to use unsupervised way to exploit the behavior of similar time series to predict the expected values. The biggest problem of such a method is its lack of precision and accuracy of the result due to unknown and unlabelled data. It generally has a recall about 7%. [Golmohammadi, K., Zaiane, O.R.: Time series contextual anomaly detection for detecting market manipulation in stock market. In: The 2015 Data Science and Advanced Analytics (DSAA’2015). pp. 1–10. IEEE (2015)]

Currently, the automatic anomoly detection system in China is still under development.

[what we did] We propose a DL method → improve precision and acc. [Chinese]

DL method what other did → examples/algos [1-2 para][水货]

Summary our advance. %, detect event [干货]

**2. Dataset:**

We analyzed **13** stocks in total, including companies that become listed in Shanghai and Shenzhen Stock exchanges. We gathered these data from dataset YouKuang[Ref], using codes such as “DataAPI.MktEqudGet(secID=u600448.XSHG",beginDate=u"20150101",endDate=u"",field=u"",pandas="1")”, and saving data in the form of CSV for further analysis. Besides, we recorded the Open, Close Price, Turnover Value, Deal Amount, etc of each stocks from 2015 to 2020.

Data is splited into training and testing data according to the ‘tradeDate’, where it starts from 2015/01/05 to 2019/12/31. The trading date range is 1219 days, where we set days = 1000 as the split. The first 1000 days are set to training data, and the last 219 days are validating data.

Table 1. Example columns of dataset.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **secID** | **ticker** | **secShortName** | **exchangeCD** | **tradeDate** | **preClosePrice** | **actPreClosePrice** | **openPrice** | **highestPrice** | **lowestPrice** | **closePrice** | **turnoverVol** | **turnoverValue** | **dealAmount** | **turnoverRate** | **accumAdjFactor** | **negMarketValue** | **marketValue** | **chgPct** | **PE** | **PE1** | **PB** | **isOpen** | **vwap** |
| 600000.XSHG | 600000 | 浦发银行 | XSHG | 2015-01-05 | 15.69 | 15.69 | 15.88 | 16.25 | 15.56 | 16.07 | 513568709 | 8182820911 | 126351 | 0.0344 | 0.6151186824 | 239809027997 | 299761285398 | 0.0242 | 6.3744 | 6.3744 | 1.1522 | 1 | 15.933 |
| 600000.XSHG | 600000 | 浦发银行 | XSHG | 2015-01-06 | 16.07 | 16.07 | 16.0 | 16.68 | 15.82 | 16.13 | 511684535 | 8311084820 | 144444 | 0.0343 | 0.6151186824 | 240704394623 | 300880493682 | 0.0037 | 6.3982 | 6.3982 | 1.1565 | 1 | 16.243 |
| 600000.XSHG | 600000 | 浦发银行 | XSHG | 2015-01-07 | 16.13 | 16.13 | 15.9 | 16.17 | 15.53 | 15.81 | 385716820 | 6114241100 | 130363 | 0.0258 | 0.6151186824 | 235929105951 | 294911382834 | -0.0198 | 6.2712 | 6.2712 | 1.1335 | 1 | 15.852 |
| 600000.XSHG | 600000 | 浦发银行 | XSHG | 2015-01-08 | 15.81 | 15.81 | 15.87 | 15.88 | 15.2 | 15.25 | 330627172 | 5101310595 | 111539 | 0.0222 | 0.6151186824 | 227572350775 | 284465438850 | -0.0354 | 6.0491 | 6.0491 | 1.0934 | 1 | 15.429 |
| 600000.XSHG | 600000 | 浦发银行 | XSHG | 2015-01-09 | 15.25 | 15.25 | 15.2 | 16.25 | 15.11 | 15.43 | 491999937 | 7692348549 | 147858 | 0.033 | 0.6151186824 | 230258450653 | 287823063702 | 0.0118 | 6.1205 | 6.1205 | 1.1063 | 1 | 15.635 |

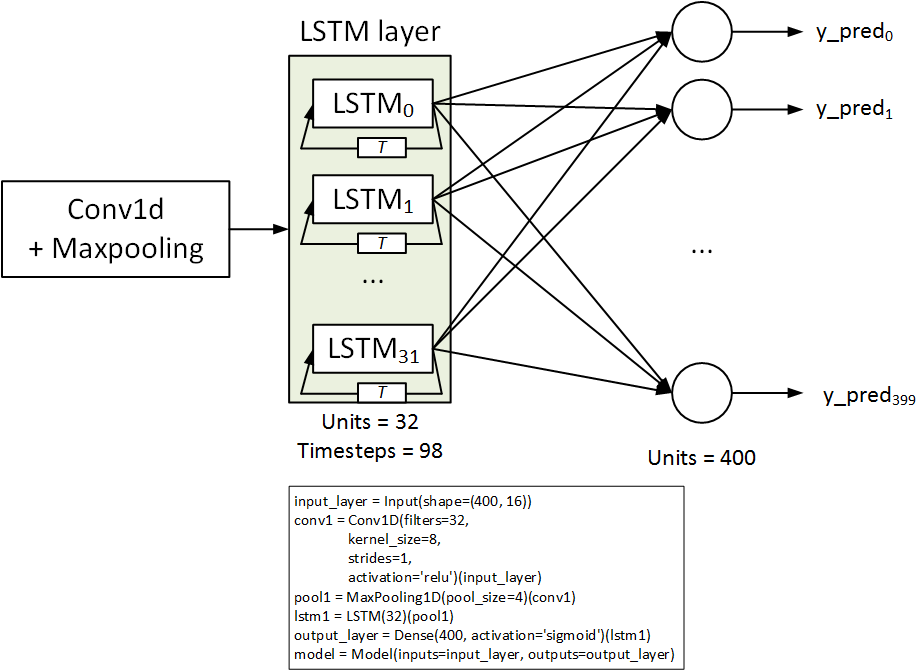
**3. Methodology:**

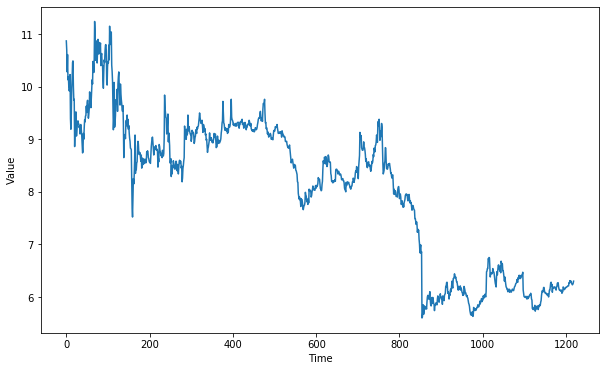
**3.1 Models**

The model is tested for 13 random selected stock tickers, where we slice the data using 1D conv method and used the data to feed into the LSTM model that can predict the time series of ‘stockPrice’.

**3.1.1 Conv1D-LSTM:**

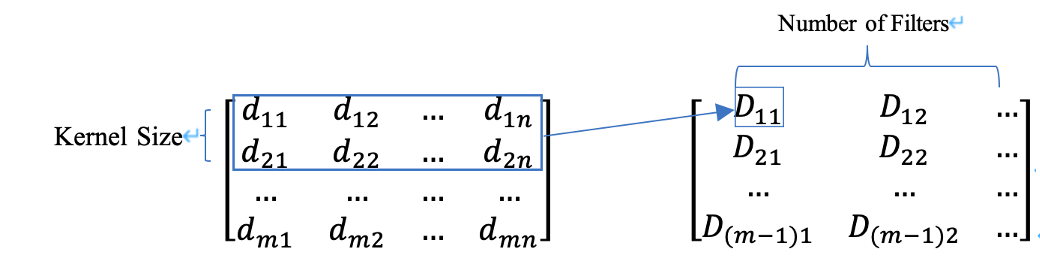
Conv1D-**LSTM**（解释） is used to derive features from datasets, especially in Time-Series. As is shown in fig[1]:





Fig[1]: Time-Series Data from MinSheng Stock

By sliding kernels（解释） along the preprocessed data, we can get feature maps.(Shown in Fig[2])

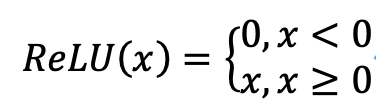


Fig[2]: An Example of Conv1D-LSTM

**先讲conv，再将LSTM，然后ReLU**

**3.1.2 Non-Linear Activation Layer:**

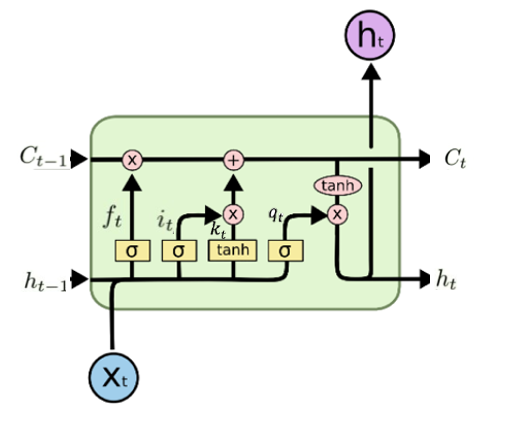
Often, non-linear activation layer is employed after convolutional layer and fully-connected layer, since non-linear operation can help keep every change made in linear operation. There are many types of non-linear function, such as Logistic Function, Hyperbolic Tangent Function, Rectified Linear Function (ReLU), etc. In this article, we use ReLU as activation function:

(1)

which makes calculation speed faster than the other functions.

**3.1.3 Long Short Term Memory Networks (LSTM):**

LSTM model is an autoregressive model much more efficient than RNNs in solving problems like vanishing and exploding gradients（需修改）. The network is made up of three parts: An input layer, some hidden layers and an output layer.（放后面） The following graph (Fig 3) demonstrates the inner structure of this model:

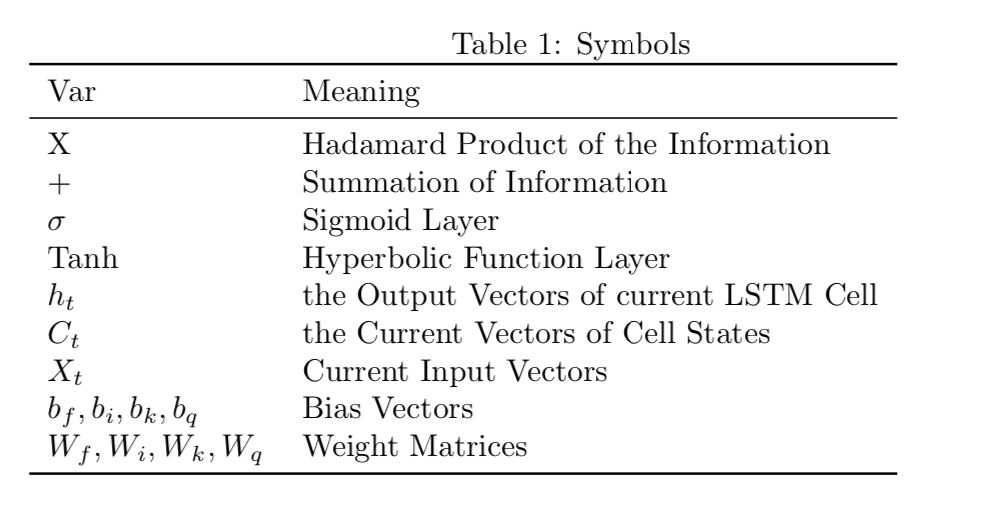


Fig[3]: The Inner Structure of LSTM Unit (Picture Originally from

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. This is the modified version)

The yellow rectangle represents Neural Network Layer, the pink circle represents pointwise operation and the arrows mainly stand for vectors. [func]

Table 1 listed variables that will be deployed later in the formula:



There are four steps to pass through LSTM:

The first step is to determine which part of the input vector ht-1 should be removed. Let ft denotes the activation value of forget gates at time t, then ft depends mainly on ht-1 and Xt. The sigmoid layer forces the output ft[0,1], where no information will pass if fx=0:

ft=[Wf(ht-1+Xt)+bf] (2)

The second step is to add additional information to the previous matrix, which consists of two parts: using a sigmoid layer to help decide which information will pass and computing the candidate vector which passes the Tanh layer. Let it, kt denote the activation function of the input gate and the candidate vector, respectively:

it=[Wi(ht-1+Xt)+bi] (3)

kt=tanh[Wk(ht-1+Xt)+bk] (4)

The third step is to combine the first and the second step by multiplying the information and their corresponding weights (activation values):

Ct=ftCt-1+itkt (5)

The last part is to calculate the output of this LSTM model. Similar to the second step, denote qt to be the activation value of output:

qt=[Wq(ht-1+Xt)+bq] (6)

ht=qttanh(Ct) (7)

**3.2 Workflow**

As there are about 1219 data in a stock, thus each stock is divided into 1000 training data and 219 testing data. First, we plot the whole data to see the stock’s trend. Then, we train the training data through one Conv1D-LSTM layer, two LSTM layers and three Dense layers with “ReLU” as activation functions and figure out the best learning rate（修改）. [call back exponential decay] In this way, RNN model can be used to forcast the stock. Later, we decide to consider “open price” and “turnover volume” as two important factors of stock price and use multistep model to build the corresponding neural network -- two LSTM layers and a Dense layer with “ReLU” activation function. By comparing predicted values with the actual values, anomalous stock prices can be found.

**4. Result**

**4.1 Model Performance**

**4.2 Predicted anomaly behavior**

**5. Conclusion**

**References:**

[1] Kim, Sangyeon, and M. Kang. "Financial series prediction using Attention LSTM." Papers (2019).

[2]<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[3] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research. <https://doi.org/10.1016/j.ejor.2017.11.054>