Automated Software Engineering (2022) 29:16
https://doi.org/10.1007/s10515-021-00303-z

Convolutional neural network for stock trading using technical indicators

S. Kumar Chandar[1]

Received: 21 April 2021 / Accepted: 3 October 2021 / Published online: 3 January 2022

Abstract
Stock market prediction is a very hot topic in financial world. Successful prediction of stock market movement may promise high profits. However, an accurate prediction of stock movement is a highly complicated and very difficult task because there are many factors that may affect the stock price such as global economy, politics, investor expectation and others. Several non-linear models such as Artificial Neural Network, fuzzy systems and hybrid models are being used for forecasting stock market. These models have limitations like slow convergence and overfitting problem. To solve the aforementioned issues, this paper intends to develop a robust stock trading model using deep learning network. In this paper, a stock trading model by integrating Technical Indicators and Convolutional Neural Network (TI-CNN) is developed and implemented. The stock data investigated in this work were collected from publicly available sources. Ten technical indicators are extracted from the historical data and taken as feature vectors. Subsequently, feature vectors are converted into an image using Gramian Angular Field and fed as an input to the CNN. Closing price of stock data are manually labelled as sell, buy, and hold points by determining the top and bottom points in a sliding window. The duration considered over a period from January 2009 to December 2018. Prediction ability of the developed TI-CNN model is tested on NASDAQ and NYSE data. Performance indicators such as accuracy and F1 score are calculated and compared to prove effectiveness of the proposed stock trading model. Experimental results demonstrate that the proposed TI-CNN achieves high prediction accuracy than that of the earlier models considered for comparison.

Keywords Artificial neural network · Convolutional neural network · Deep learning · Gramian angular field · Stock trading and technical indicators

✉ S. Kumar Chandar
kumar.chandar@christuniversity.in

[1] School of Business & Management, CHRIST (Deemed To Be University), Bangalore, India

1 Introduction

Financial market provides a platform to the buyers and sellers to meet for trading assets at a price determined by the demand and supply forces. It acts as an intermediary between the savers and investors by mobilizing funds between them. Some common forms of financial markets are stock market, bond market, commodity market and derivatives market. Stock market is a public market for security where organized issuance and trading of company stocks take place either through exchange or over the counter in physical or electronic form. It is nowadays a common notion that huge amounts of capital are traded through stock market all around the world (Chandar et al. 2016; Wadi et al. 2011). The stock market or share market is one of the most important places for investment where investor can get high profits by investing their resources in shares. However, stock trading is very risky task, the decision-making process in stock trading is a very important process because it must be taken correctly and in right time.

Three approaches are commonly used to analyze and predict behaviors of financial market are: (i) Intrinsic or fundamental analysis and (ii) Technical or chartist analysis and (iii) Time series analysis. Fundamental analysis is suitable for long term prediction (Selvin et al. 2017). Technical analysis is most suited for short term forecasting. Time series analysis further divided into linear and non-linear models.Linear models such as the Autoregressive Integrated Moving Average (ARIMA) and the Autoregressive Conditional Heteroscedasticity (ARCH) has been used to predict the stock price. These models are based on some predefined assumptions such as normality postulated (Chung and Shin 2018). However, these models failed to capture the patterns present in the stock data.

Non-linear models involve Artificial Neural Networks (ANNs), Adaptive Neuro Fuzzy Interference System (ANFIS) and deep learning neural networks (Chandar 2019). These models have been commonly used to make predictions about stock market because they can extract nonlinear relationships between stock data without prior knowledge of the input data (Atsalakis and Valavanis 2009). Recently, Deep learning Neural Networks (DNNs) has proven itself to be a powerful soft computing tool and has a wide variety of applications (Kim 2014). Deep learning-based classification or prediction model started emerging as the best performance achieved in various fields, outperforming the conventional models (Sezer and Ozbayoglu 2018) (Graves et al. 2013) (Xu et al. 2015). Deep Boltzmann Machine (DBM), Conventional Neural Network (CNN), Long Short-Term Memory (LSTM) and Auto Encoders (AE) are the widely used deep learning architectures for time series forecasting. Amidst, CNN based model perform better due to various reasons.

This paper proposes a stock trading model to predict stock movement by integrating Technical Indicators with CNN named as TI-CNN. The developed stock trading model employed to determine the buying and selling points in stock price based on image information. Set of ten technical indicators such as Simple Moving Average (SMA7), SMA21, Exponential MA (EMA7), EMA21, Moving Average Convergence/Divergence (MACD), Williams %R, Stochastic Oscillator (SO)

%K, SO % D, Relative Strength Index (RSI), Rate of Change (ROC) and Accumulation/Distribution Oscillator (ADO) are derived from the historical data. The obtained technical indicators are converted into an image using Gramian Angular Field (GAF) method. Closing price of stock data are manually labelled as sell, buy, and hold points by determining the top and bottom points in a sliding window. The duration is considered for ten years, from January 2009 to December 2018. Prediction ability of the TI-CNN model is validated using NASDAQ and NYSE stock market data. Performance measures like accuracy and F1 score are calculated and compared to prove effectiveness of the proposed stock trading model. Further, this contribution compares the proposed stock trading model with that of earlier models found in the literature. Simulation results show that the proposed TI-CNN performs better than that of the earlier models considered for comparison.

The rest of the paper is structured as follows: Sect. 2 provides a brief review of earlier methods related to stock market prediction. Section 3 presents proposed TI-CNN model for stock trading. Section 4 discusses the data, detailed analysis and comparison of the experimental results. Section 5 concludes the paper followed by relevant references.

2 Literature survey

In the following section, the paper focuses on the review of the prior studies regarding the prediction of stock market data with deep learning architectures. Sezer and Ozbayoglu (2018) used CNN to create a trading system that can give buy, sell and hold points in Dow Jones 30 stocks. 15 technical indicators such as RSI, Williams's % R, WMA, EMA, SMA, HMA, Triple EMA, CCI, CMO, MACD, PPO, ROC, CMFI, DMI and PSI for different intervals, 6 to 20 are computed from the historical data. An image of size 15X15 is created and then fed as input to the CNN. Proposed model consists of 9 layers such as input layer, two convolutional layers, a max pooling layer, two dropout layer, two fully connected layers and output layer. Nelson et al. (2017) developed a model for predicting stock price movement using Long Short-Term Memory (LSTM). Two technical indicators such as MACD and RSI derived from historical data and used an input to the LSTM. Result showed that LSTM model provides an accurate prediction than other models including Multilayer Perceptron (MLP), Random Forest (RF) and pseudo random model. Three stage stock prediction model developed by Bao et al. (2017). In this approach, wavelet transform was used to reduce the dimension of the stock data. Data were reproduced by Stacked Auto Encoder (SAE) and prediction done by LSTM. The authors utilized three different input variables such as (i) daily trading data (opening price, closing price, lowest price, highest price and volume), (ii) technical indicators (MACD, CCI, EMA20, SMA5, SMA10 and ROC) and (iii) two macroeconomic variables (exchange rate and interest rate) as input variables. Experimental results proved that the performance of LSLTM model is superior to RNN, standard LSTM and wavelet LSTM. Vargas et al. (2017) predicted the intraday directional movement of S&P 500 with

Recurrent Convolutional Neural Network (RCNN). RCNN designed to have strength of both RNN and CNN. Proposed model utilizes two inputs: technical indicators and financial news. In this model, word2vec model employed to generated word. Results showed improved performance. Hiransha et al. (2018) made use of four deep learning models namely LSTM, CNN, RNN and MLP for stock prediction. The authors used day-wise closing price of stock as in input because day wise stock price is preferred since investors make decision on buying which stock or forfeiting which stock based on the closing price of the market. Proposed models were tested on NSE and NYSE. The network was trained with the stock price of a single company from NSE and predicted for five different companies from both NSE and NYSE. Experimental results revealed that the CNN model outperforms the other models such as LSTM, RNN and MLP. LSTM based stock prediction model was proposed by Gao et al. (2018). In this approach, fifteen technical indicators such as Accumulation Distribution (ACD), MACD, oscillator, highest price, lowest price, stochastic %K, stochastic %D, Volume Price Trend (VPT), William's %R, RSI, momentum, acceleration, ROC, Volume ROC and OBV. Dimension of the feature vectors were reduced by Principal Component Analysis (PCA). The developed model was tested on S&P 500, NASDAQ and AAPL. Results showed slightly higher prediction accuracy compared to MLP. Chen and He (2018) used 1D CNN to predict the stock price movement of Chinese stock market. Opening price, highest price, lowest price, closing price and volume are taken as feature vectors to be trained by CNN. Kim and Kim (2019) utilized LSTM and CNN for developing stock forecasting model that combines features from different representations of the same data, stock time series and stock chart image to forecast stock prices. LSTM and CNN employed to extract the temporal features and image features respectively (Tripathi 2021; Sungheetha and Sharma 2021; Dhaya 2021a, b; Chen and Lai 2021). Results showed that the LSTM-CNN model outperforms the single models in predicting stock prices.

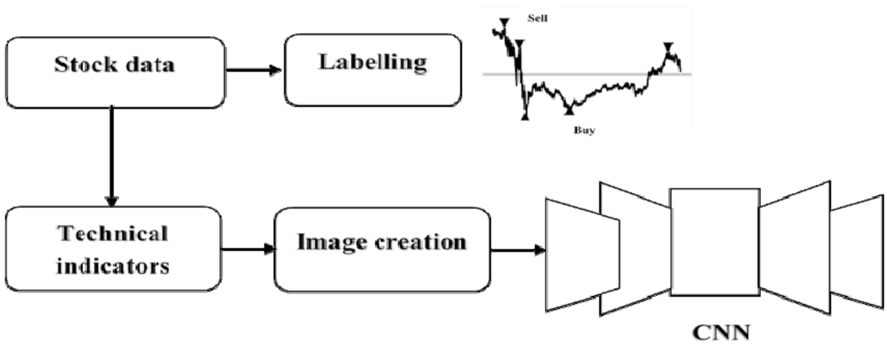


Fig. 1 Proposed TI-CNN model for stock trading

3 Proposed methodology

The main goal of this work is to design a CNN based stock trading model to determine the buying and selling points in financial data using technical indicators as input. The developed model is named TI-CNN and is depicted in Fig. 1.

Historical data collected from the publicly available resource and considered as the input. Technical indicators are derived from the past data and taken as feature vector. Closing price is manually labelled as sell, buy, and hold points by determining the top and bottom points in a sliding window. In this work, CNN is used to develop a predictive model. Therefore, the computed technical indicators are transformed to image using GAF and fed as input to the predictive model. Finally, the predictive model makes decision about stock trading. The following steps are involved in the proposed model.

- Labelling
- Technical indicators computation
- Image creation and
- CNN

The detailed approach of each stage is explained as follows:

3.1 Labelling

The daily closing prices of eight stocks representing different sectors are gathered from NASDAQ and NYSE. The whole data set spanning from 2009 to 2018. After collecting data, closing prices of selected stocks marked manually as buy, sell and hold by computing highest (top or maximum) and lowest prices (bottom or minimum) in a sliding window. In this work, sliding window with retraining approach were 6 years used for training, 2009–2014 and the following one year, 2015 for testing. As in Fig. 1, top points are labelled as sell and bottom points are marked as buy. Remaining points are labelled as hold.

3.2 Technical indicators computation

Technical indicators are the derived attributes. Several technical indicators are reported in the literature (Caley 2013). Technical indicators are computed from historical data to make up initial variables, as determined by the review of experts along with previous studies. In this work, ten technical indicators are calculated and then used to create an image by reviewing previous research and experts (Chung and Shin, 2018; Kara et al. 2011; Majhi et al. 2014; Rout et al. 2014). Selected technical indicators along with their formula tabulated in Table 1.

Table 1 Technical indicators

Technical indicators	Formula
SMA7	$SMA_7 = \frac{1}{7}\sum_{i=1}^{7} C_i$
SMA21	$SMA_{21} = \frac{1}{21}\sum_{i=1}^{21} C_i$
EMA7	$\mathrm{K}\left(\mathrm{C_t} - \mathrm{EMA(t-1)}\right) + \mathrm{EMA(t-1)}$ $\mathrm{K} = \frac{2}{1+n}$
EMA21	$\mathrm{K}\left(\mathrm{C_t} - \mathrm{EMA(t-1)}\right) + \mathrm{EMA(t-1)}$ $\mathrm{K} = \frac{2}{1+n}$
RSI	$\mathrm{RSI} = 100 - \frac{100}{1+\frac{EMA(U,n)}{EMA(D,N)}}$ $\mathrm{U = C_t - C_{t-1}} \quad \mathrm{D = 0}$ $\mathrm{D = C_{t-1} - C_t} \quad \mathrm{U = 0}$
MACD	$\mathrm{MACD = EMA(12) - EMA(26)}$ $\mathrm{Signal = EMA(MACD, 9)}$ $\mathrm{Histogram = MACD - Signal}$
Stochastic %K	$\%\,\mathrm{K} = 100\frac{\mathrm{C_t - C_l(n)}}{\mathrm{C_h(n) - C_L}}$
Stochastic %D	$\%\,\mathrm{D} = \mathrm{EMA}(\%\,\mathrm{K}, 3)$
ROC	$\left(\frac{\mathrm{Price(t)}}{\mathrm{Price(t-n)}}\right) * 100$
William's R %	$\left(\frac{\mathrm{H_t - C_t}}{\mathrm{H_t - L_t}}\right) * 100$

C-Closing price, H-Highest price, L-Lowest price

3.3 Image creation

Gramian Angular Field (GAF) employed to create an image. GAF represents time series in a polar coordinate system instead of Cartesian coordinates (Damaševicius et al. 2018). Ten technical indicators listed in Table 1 are computed from the historical stock prices. Subsequently, the indicators are normalized using Eq. (1) to scale the data range between 0 and 1 and to improve the convergence rate.

$$\tilde{\mathrm{X}} = \frac{\mathrm{X} - \min(\mathrm{X})}{\max(\mathrm{X}) - \min(\mathrm{X})} \tag{1}$$

where $\tilde{\mathrm{X}}$ is the normalized value, x is the real or actual value and min and max represents the minimum and maximum value respectively. Rescaled time series $\tilde{\mathrm{X}}$ is represented in polar coordinates by encoding the value as the angular cosine and time stamp as the radius. Mathematically, encoding process can be expressed as:

$$\begin{cases} \phi = \arccos\left(\tilde{x}_i\right) & 0 \le x_i \le 1, \tilde{x}_i \in \tilde{X} \\ r = \frac{t_i}{M} & t_i \in M \end{cases} \tag{2}$$

where t_i is the time stamp and M is a constant factor to regularize the span of the polar coordinate system. Finally, polar coordinate values are converted into image by considering the trigonometric sum between each point to identify the temporal correlation within different time intervals, given in Eqs. (3) and (4).

$$GAF = \begin{bmatrix} \cos(\phi 1 + \phi 1) & \cdots & \cos(\phi 1 + \phi m) \\ \vdots & \ddots & \vdots \\ \cos(\phi m + \phi 1) & \cdots & \cos(\phi m + \phi m) \end{bmatrix} \tag{3}$$

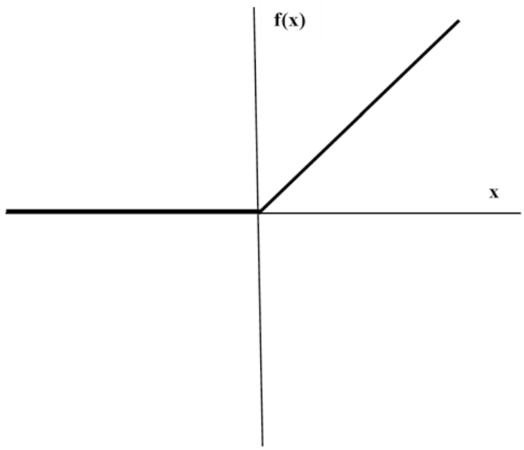
$$= \tilde{X}'.\tilde{X} - \left(\sqrt{I - \tilde{X}'}\right)^2.\sqrt{I - \tilde{X}^2} \tag{4}$$

3.4 Convolutional neural network

Convolutional Neural Networks (CNNs) are kind of deep neural networks, introduced to solve the problems of standard neural network like Multilayer Perceptron (MLP). CNNs are feed forward networks offer the potential of extracting the most important details from the unprocessed or raw input data. It does not require any preprocessing. Additionally, CNNs have the capability of recognizing patterns with extreme variability and some geometric transformations such as scaling, rotation, translation and noise (Kim and Kim 2019) (Khalajzadeh et al. 2014). CNN is composed of multiple convolutional layers, pooling or subsampling layers followed by fully connected layer. In this work, proposed CNN consists of 8 layers namely input layer (64 × 64), two convolution layers (64 × 64 × 16, 32 × 32 × 32), two max pooling layers (2 × 2), a dropout layer (0.2), a fully connected layer and an output layer (3). Dropout layer added to avoid overfitting problem. In literature, different size of receptive field adapted: 3 × 3, 5 × 5, 7 × 7 and 9 × 9. Minimum size of receptive filed can catch more details of the image. Therefore, 3 × 3 receptive filed is preferred since small image is used. Convolution layer performs the convolution operation. Mathematically convolution operation can be defined as,

$$I(i,j) = (I(i,j) * R(m,n) = \sum_m \sum_n I(m,n)R(i-m)(j-n) \tag{5}$$

Fig.2 ReLU Activation Function



To introduce non-linearity *ReLU* activation function is used; The *ReLU* function can be defined as follows and graphically represented in Fig. 2.

$$ReLU(x) = \begin{cases} 0 & (x \leq 0) \\ x & (x > 0) \end{cases} \tag{6}$$

The feature maps of first convolution layer will be the input of first pooling layer and they will be down sampled to reduce the dimension and computation. Down sampled features are used will be input to the second convolution layer. Second feature set again pooled by second pooling layer. Subsequently, a dropout layer followed by fully connected layer and output layer. *SoftMax* function is employed to obtain the output. Equation (7) shows the *SoftMax* function,

$$Softmax(y_i) = \frac{e^{y_i}}{\sum_{i=1}^{j} y_j} \tag{7}$$

4 Experimental results and discussions

This section depicts the simulation results of the proposed stock trading model. It also provides detail regarding simulation platform, data set, performance indicators which have been utilized to make fair comparison of the proposed model with the earlier models.

4.1 Simulation platform

The prime objective of simulation tool is to produce better results. In this work, the developed stock trading model is implemented on MATLAB2018a platform using deep learning toolbox and executed in Intel core i5 processor with 2.5 GHz speed and 12 GB RAM.

Table 2 Selected Stocks

Stock name	Stock ID	Stock market	Start date	End date	Number of samples
Apple Inc	AAPL	NASDAQ	02.01.2009	28.12.2018	2515
Bank of America corporation	BAC	NYSE	02.01.2009	28.12.2018	2515
CTS Corporation	CTSH	NASDAQ	02.01.2009	28.12.2018	2515
The Goldman Sachs Group, Inc	GS	NYSE	02.01.2009	28.12.2018	2515
Halliburton company	HAL	NYSE	02.01.2009	28.12.2018	2515
Microsoft corporation	MSFT	NASDAQ	02.01.2009	28.12.2018	2515
Oil States International, Inc	OIS	NYSE	02.01.2009	28.12.2018	2515
Oracle corporation	ORCL	NASDAQ	02.01.2009	28.12.2018	2515

4.2 Dataset

In this work, daily stock prices of eight companies representing different sectors form NASDAQ and NYSE employed for training and testing purposes. List of selected stocks and duration are provided in Table 2. Historical stock data of eight companies spanning the period from January 2009 to December 2018 sourced from yahoo finance (xxxx). To investigate the robustness of the proposed model, sliding method is adopted with retraining approach where data from 2009 to 2014 used for training and the following year, 2015 used for testing. Then, both training and testing periods are moved one year ahead, retrained the model and tested. Now, training period is 2010 to 2015 and testing period is 2016.

4.3 Performance indicators

The overall performance of developed stock trading model, TI-CNN is measured using two statistical measures such as accuracy and F-measure.

4.3.1 Accuracy

Given test data-sets, the accuracy is the number of samples correctly classified by the classifier divided by the total number of samples.

4.3.2 F1score

Is the harmonic mean precision and recall.

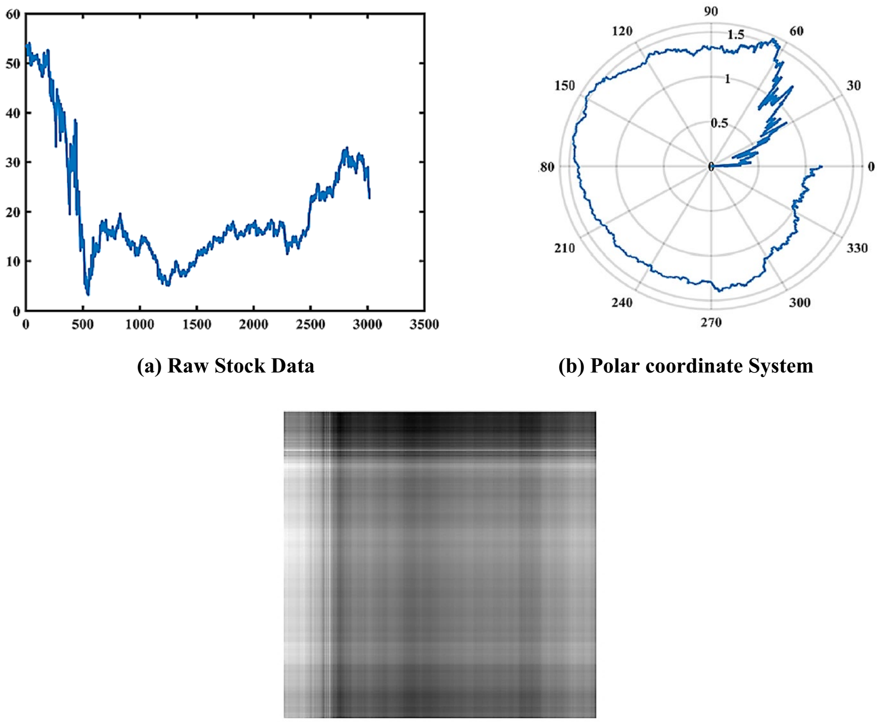
4.4 Experimental treatment

Ten technical indicators such as SMA7, SMA21, EMA7, EMA21, MACD, RSI, ROC, %K, %D and Williams's %R are computed from the historical data. The obtained values are converted into image using GAF. Sample of encoding time series to an image is shown in Fig. 3. Figure 4 shows the structure of CNN used for developing stock trading model. As in Fig. 4, input image is applied to the convolution layer followed by pooling layer and so on. Adam optimizer is employed and the model is trained for 500 epochs.

Table 3 shows the efficiency of the proposed stock trading model, TI-CNN in terms of accuracy and precision. From the Table 3, it is observed that the developed model TI-CNN possess higher prediction accuracy and F1 score in almost all the stocks. Figures 5 and 6 provides the graphical delineation of this Table 3.

4.5 Effect analysis

The developed TI-CNN stock trading model is successfully implemented to determine the buy and sell points in the selected stocks and to validate its performance is compared with the other existing methods such as CNN-Technical Analysis



(a) Raw Stock Data

(b) Polar coordinate System

(c) Stock Image

Fig. 3 Illustration of encoding time series to an image conversion

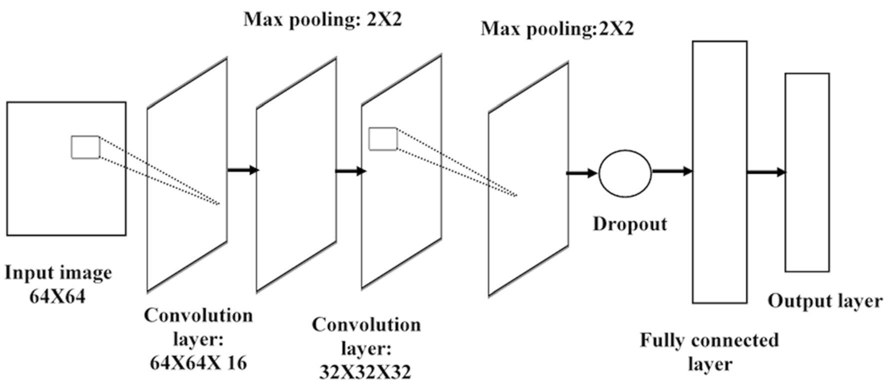


Fig. 4 Convolutional neural network

(CNN-TA) (Sezer and Ozbayoglu 2018), 1D CNN (Chen and He 2018), and CNN (Xu et al. 2018) in terms of accuracy and F1 score. Sezer et al. (2018) used CNN for determining the sell, buy and hold points for stock trading. Technical indicators, triple EMA, CCI, CMO, MACD, PPO, ROC, CMFI, DMI, PSI, RSI, Williams's %R,

Table 3 Performance of the proposed TI-CNN model

Stock ID	Accuracy	F1 score
AAPL	65	0.68
BAC	89.6	0.89
CTSH	70.8	0.73
GS	78	0.74
HAL	85.6	0.84
MSFT	67	0.65
OIS	84	0.77
ORCL	82	0.74

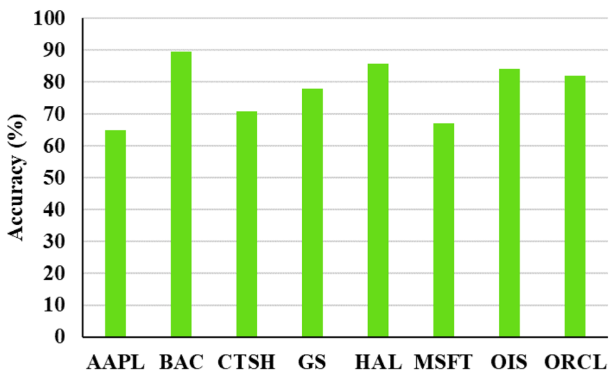


Fig. 5 Performance of the proposed method in terms of accuracy

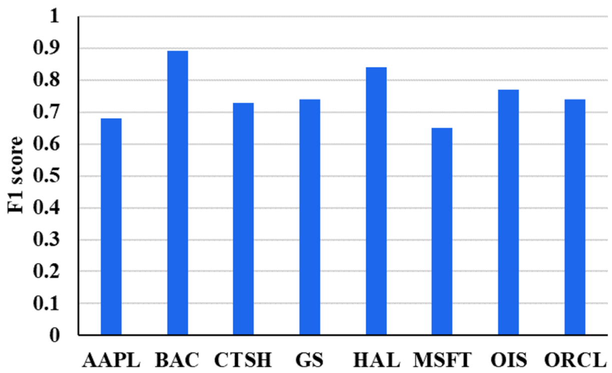


Fig. 6 Performance of the proposed method in terms of F1 score

EMA, SMA and HMA are calculated different intervals (6 to 20) from historical data. Image (15X15) is created using technical indicators. Then, the image is used as input to CNN. Chen et al. (2018) developed a model for stock trading using CNN. Five features namely opening price, highest price, lowest price, closing prince and

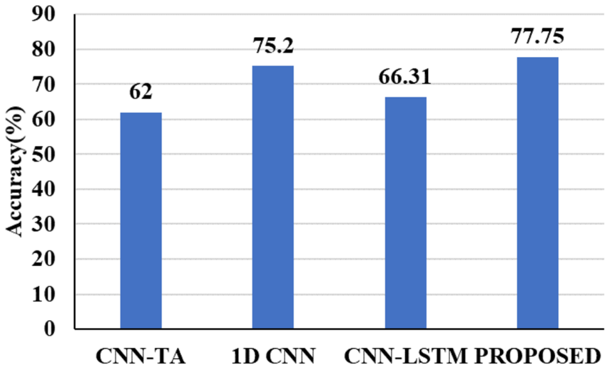


Fig. 7 Accuracy Comparison

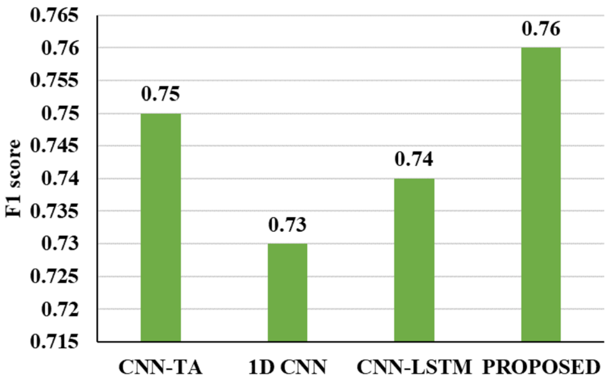


Fig. 8 Performance Comparison in terms of F1 Score

volume are taken as feature vectors and used input to the CNN. 1D CNN perform the classification task. Xu et al. (2018) proposed stock trading model using CNN. The authors used financial information as an input to the proposed model. Figures 7 and 8 graphically compares the performance of the developed TI-CNN with the other existing models with respect to performance indicators mentioned in Sect. 4.3. It is proved that the image created based on the selected technical indicators can accurately trend of stock market. It is evident that the selected indicators have high predictive power.

From Figs. 5 and 6, it can be seen that the developed TI-CNN model gives better performance compared to the other models considered for comparison. Mean accuracy of 77.75% is obtained using TI-CNN model which is higher than other models. In the models proposed and implemented by Sezer et al. (TA-CNN) Sezer and Ozbayoglu (2018), Chen et al. (1D CNN) Hiransha et al. (2018) and Xu et al. (CNN) Khalajzadeh et al. (2014), it is found that its accuracy and F1 score is lower than the developed model proving the TI-CNN model gives better results in stock trading. It is evident that the proposed TI-CNN model can determine the buy, hold and sell points in the stock prices since CNN has a superior ability to extract most important features from the image.

5 Conclusion

This paper presented a deep learning neural network-based stock trading model is to determine buy, sell and hold points in stock data. Stock trading model is developed using convolutional neural network. First of all, ten technical indicators such as SMA7, SMA21, EMA7, EMA21, MACD, RSI, ROC, %K, %D and Williams's %R are computed from historical data. Then, it is adopted to convert time series data to an image. The resultant image is taken as feature and fed as an input to the CNN. Experiments are conducted on ten years of stock data of eight companies namely AAPL, BAC, CTSH, GS, HAL, MSFT, OIS and ORCL from NASDAQ and NYSE stock markets. Performance indicators used for assessing efficiency of the proposed model includes accuracy and F1 score. It is observed from the obtained experimental results that the proposed TI-CNN model achieves better accuracy and F1 score in comparison with that of the earlier deep learning models available in the literature for finding entry and exit points in the stock prices.

References

Atsalakis, G.S., Valavanis, K.P.: Surveying stock market forecasting techniques–Part II: Soft computing methods. Expert Syst. Appl. **36**, 5932–5941 (2009). https://doi.org/10.1016/j.eswa.2008.07.006

Bao, W., Yue, J., Rao, Y.: A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS One (2017). https://doi.org/10.1371/journal.pone.0180944

Caley, J.A.: A survey of systems for predicting stock market movements, combining market indicators and machine learning classifiers. Dissertations and Theses (2013)

Chandar, S.K.: Fusion model of wavelet transform and adaptive neuro fuzzy interference system for stock market prediction. J. Ambient Intell. Humaniz. Comput. (2019). https://doi.org/10.1007/s12652-019-01224-2

Chandar, S.K., Sumathi, M., Sivanandam, S.N.: Prediction of stock market using hybrid of wavelet transform and artificial neural network. Indian J. Sci. Technol. (2016). https://doi.org/10.17485/ijst/2016/v9i8/87905

Chen, S., He, H.:Stock prediction using convolutional neural network. In: IOP Conference Series in Material Science and Engineering,435 (2018)

Chen, J.L., Lai, K.L.: Deep convolution neural network model for credit-card fraud detection and alert. J. Artif Intell **3**(02), 101–112 (2021)

Chung, H., Shin, K.S.: Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability MDPI **10**(10), 3765 (2018)

Damaševicius, R., Maskeliunas, R., ozniak, M., Polap, D.: Visualization of physiologic signals based on Hjorth parameters and gramian angular fields. In: IEEE 16th World Symposium on Applied Machine Intelligence and Informatics. (2018)

Dhaya, R.: Hybrid machine learning approach to detect the changes in SAR images for salvation of spectral constriction problem. J. Innov. Image Process. (JIIP) **3**(02), 118–130 (2021a)

Dhaya, R.: Light weight CNN based robust image watermarking scheme for security. J. Inf. Technol. Digit World **3**(2), 118–132 (2021b)

Gao, T., Chai, Y.: Improving stock closing price prediction using recurrent neural networks and technical indicators. Neural Comput. **30**(10), 2833–2854 (2018). https://doi.org/10.1162/neco_a_01124

Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (icassp). pp. 6645–6649 (2013)

Hiransha, M., Gopalakrishanan, E.A., Menon, V.K., Soman, K.P.: NSE stock market prediction using deep learning models. Procedia Comput. Sci. **132**, 1351–1362 (2018). https://doi.org/10.1016/j.procs.2018.05.050

https://finance.yahoo.com/

Kara, Y., Boyacioglu, M.A., Baykan, Ö.K.: Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. Expert Syst. Appl. **38**, 5311–5319 (2011). https://doi.org/10.1016/j.eswa.2010.10.027

Khalajzadeh, H., Manthouri, M., Teshnehlab, M.: Face recognition using convolutional neural networks and simple logistic classifier. Adv. Intell. Syst. Comput. **223**, 197–207 (2014)

Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)

Kim, T., Kim, H.Y.: Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. PloS ONE (2019). https://doi.org/10.1371/journal.pone.0212320

Majhi, B., Rout, M., Baghel, V.: On the development and performance evaluation of a multi objective GA based RBF adaptive model for the prediction of stock indices. J. King Saud Univ. Comput. Inf. Sci. **26**, 319–331 (2014). https://doi.org/10.1016/j.jksuci.2013.12.005

Nelson, D.M.Q., Pereira, A.C.M., De Oliveira, R.A.: Stock market price movement prediction with LSTM neural networks. In: International Joint Conference on Neural Networks. Pp. 1419–1426 (2017). https://doi.org/10.1109/IJCNN.2017.7966019

Rout, M., Majhi, B., Majhi, R., Panda, G.: Forecasting currency exchange rates using an adaptive ARMA model with differential evolution-based training. J. King Saud Univ. Comput. Inf. Sci. **26**, 7–18 (2014). https://doi.org/10.1016/j.jksuci.2013.01.002

Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K Soman, K.P.:Stock price prediction using lstm,rnn and cnn-sliding window model. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, IEEE, 2017. https://doi.org/10.1109/ICACCI.2017.8126078

Sezer, O.M., Ozbayoglu, A.M.: Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Appl. Soft Comput. **70**, 525–538 (2018). https://doi.org/10.1016/j.asoc.2018.04.024

Sungheetha, A., Sharma, R.: Design an early detection and classification for diabetic retinopathy by deep feature extraction based convolution neural network. J. Trends Comput. Sci. Smart Technol. (TCSST) **3**(02), 81–94 (2021)

Tripathi, M.: Analysis of convolutional neural network based image classification techniques. J. Innov Image Process. (JIIP) **3**(02), 100–117 (2021)

Vargas, M.R., de Lima, B.S., Evsukoff, A.G.:Deep learning for stock market prediction from financial news articles. In: Proceedings of the IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications. (2017). Doi: https://doi.org/10.1109/CIVEMSA.2017.7995302

Wadi, S.A., Ismail, M.T., Alkhahazaleh, M.H., Karim, S.A.A.: Selecting wavelet transforms model in forecasting financial time series data based on ARIMA Model. Appl. Math. Sci. **5**(7), 315–326 (2011)

Xu, B., Zhang, D., Zhang, S., Li, H., Lin, H.: Stock market trend prediction using recurrent convolutional neural networks. In: Zhang, M., Ng, V., Zhao, D., Li, S., Zan, H. (eds.) Natural Language Processing and Chinese Computing NLPCC 2018 Lecture Notes in Computer Science. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-99501-4_14

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.:Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning. vol. **14**, pp. 77–81 (2015)