Nti *et al. J Big Data* (2021) 8:17
https://doi.org/10.1186/s40537-020-00400-y

Journal of Big Data

RESEARCH Open Access

A novel multi-source information-fusion predictive framework based on deep neural networks for accuracy enhancement in stock market prediction

Isaac Kofi Nti[1,2*], Adebayo Felix Adekoya[1] and Benjamin Asubam Weyori[1]

*Correspondence:
Ntious1@gmail.com
[1] Department of Computer Science and Informatics, The University of Energy and Natural Resources, Sunyani, Ghana
Full list of author information is available at the end of the article

Abstract

The stock market is very unstable and volatile due to several factors such as public sentiments, economic factors and more. Several Petabytes volumes of data are generated every second from different sources, which affect the stock market. A fair and efficient fusion of these data sources (factors) into intelligence is expected to offer better prediction accuracy on the stock market. However, integrating these factors from different data sources as one dataset for market analysis is seen as challenging because they come in a different format (numerical or text). In this study, we propose a novel multi-source information-fusion stock price prediction framework based on a hybrid deep neural network architecture (Convolution Neural Networks (CNN) and Long Short-Term Memory (LSTM)) named IKN-ConvLSTM. Precisely, we design a predictive framework to integrate stock-related information from six (6) heterogeneous sources. Secondly, we construct a base model using CNN, and random search algorithm as a feature selector to optimise our initial training parameters. Finally, a stacked LSTM network is fine-tuned by using the tuned parameter (features) from the base-model to enhance prediction accuracy. Our approach's emperical evaluation was carried out with stock data (January 3, 2017, to January 31, 2020) from the Ghana Stock Exchange (GSE). The results show a good prediction accuracy of 98.31%, specificity (0.9975), sensitivity (0.8939%) and F-score (0.9672) of the amalgamated dataset compared with the distinct dataset. Based on the study outcome, it can be concluded that efficient information fusion of different stock price indicators as a single data source for market prediction offer high prediction accuracy than individual data sources.

Keywords: Deep neural networks, Convolution neural network, Long short-term memory, Information fusion system, Stock market, Google trends, Algorithmic trading

Introduction

The conventional stock market prediction methods usually use the historical stock dataset to predict stock price movement [1, 2]. However, in this information age and technology, information amalgamation is a vital ingredient in decision-making processes [3]. Besides, the abundance of information sources such as the Internet, databases, chat,

email and social networking sites are growing exponentially [4]; and the stock market is one place where several Terabytes and Petabytes of information is generated daily from these sources.

However, stock market data's ubiquitousness makes effective information fusion in market analysis a challenging task [1, 5]. Notwithstanding, stock market information is multi-layered and interconnected [5]; hence, the ability to make intelligence of data, by fusing them into new knowledge would offer distinct advantages in predicting the stock market [6]. Therefore, Multi-source Data-Fusion (MDF) has become a key area of interest in recent studies in this field [7]. i.e., MDF aims to attain a global view of all factors affecting the stock price movement and make the best investment decision. Nonetheless, the ability to fuse all these factors (stock price indicators) into useful information is hindered by the fact that these factors are generated from several sources in different formats (numerical or text).

Primarily, stock-related information can be clustered into two, namely quantitative (numerical) dataset and qualitative (textual) dataset. The quantitative dataset includes historical stock price and economic data, based on these; the analyst predicts stock price movement [2]. Nevertheless, Zhang et al. [8], argued that quantitative stock market data could not convey the complete information concerning companies' financial standings. Hence, qualitative information such as the economic standing of the firm, the board of directors, employees, financial status, balance-sheets, firm's yearly income-reports, regional and political data, climatic circumstances like unnatural or natural disasters enshrouded in the textual descriptions from various data sources can be effectively be used to predicts stock price movement or complementary quantitative data [8]. However, Nti et al. [9] pointed out that there is a limited size of qualitative information on the stock market from developing countries; hence, it is inadequate to solely depend on qualitative information to predict future stock price from undeveloped and developing countries.

Thus, both quantitative and qualitative information sources are very vital in developing better and highly accurate predictive models in the stock market [1, 10]. Therefore, it is reasonable to acquire comprehensive data of both textual and numerical to predict the future stock price of a firm. On the other hand, studies [2] show that few studies (11% out of 122 studies) in stock market prediction attempted to fuse both (quantitative and qualitative) to predict future stock price movement. Moreover, as indicated earlier, the stock market is influenced by several factors; therefore, relying on a single data-source might not be adequate to make accurate predictions.

To examine in totality, and quantify the effects of these factors on stock price movement; we proposed a multi-source information-fusion stock market prediction framework. The framework is based on deep hybrid neural networks architecture (CNN and stacked LSTM), named IKN-ConvLSTM. Specifically, we present an information-fusion technique for amalgamating three quantitative and three qualitative stock-related information sources. It is the first study to put forward such a comprehensive information fusion framework for stock market prediction to the best of our knowledge. Additionally, we try to obtain the effects of individual information-source on stock market price movement. Thus, we detect the significant factors that have decisive impacts. These factors may be collective sentiments or economic variables or some vital features in the trading data or Google

trends index or essential Web news. Finally, we integrate CNN and stacked LSTM architecture for efficient feature selection, detection of unique features in terms of specificity and accurate stock price movement prediction.

In this study, we adopted the CNN and LSTM. The two were adopted because studies show that CNN can automatically notice and extract the appropriate internal structure from a time series dataset to create in-depth input features, using convolution and pooling operations [11, 12]. Additionally, CNN and LSTM algorithms are reported to outperform state-of-the-art techniques regarding noise tolerance and accuracy for time-series classification [11, 13–15]. Furthermore, LSTM and CNN's amalgamation has previously achieved high-accurate results in areas like speech recognition, where sequential modelling information is required [16–18]. Lastly, CNN and LSTM algorithms are competent and capable of learning dependencies within time series without the necessity for substantial historical time series data. Also, lesser time and effort in terms of their implementation [13, 14, 19]. The contributions of the current study to literature can be summarised as follows:

1. A hybrid deep neural networks predictive framework built on CNN and stacked LSTM (named IKN-ConvLSTM) machine learning algorithm; that fuses six heterogeneous stock price indicators (users' sentiments (tweets), Web news, forum discussion, Google trends, historical macroeconomic variables, and past stock data) to predict future stock price movement.
2. We propose a reduction in the data sparsity problems and use the harmonies among stock-related information, by exploring the association among these information sources with deep neural networks. As an alternative to a simple linear combination of the stock-related information, we consider the combined effects among information source to capture their associations.
3. We explored the ideology that traditional technical analysis combined with investors and experts' sentiments or opinions (fundamental analysis) will give better stock price prediction accuracy.
4. We evaluated the effectiveness of the proposed framework experimentally with real-world stock data from the Ghana stock market and compared it with three baseline techniques. The results show that the prediction performance of machine learning models can be significantly improved by merging several stock-related information.

We organised the remainder of this paper as follows. In "Related works" section, we present pertinent literature on stock market analysis. Section "Methodololgy" shows the procedures and techniques applied for combining six heterogenous stock-related information source and analysing their impact on predicting the stock market. We summarised the results and discussion of this study in "Empirical Results and Discussions" section. Finally, Sect. Conclusions shows the conclusions from this work.

Related works

Recently, countless studies have been reported in the literature from journals, conferences, magazine, and many more on stock market analysis. Succinctly, 66% of these studies utilised historical stock price (Quantitative), 23% qualitative (textual) dataset to predict the future stock prices with various models [2]. The following section presents

some recent and relevant literature; we categorised them based on the dataset-type (quantitative, qualitative and both).

Studies Based on Quantitative Dataset

A predictive model based on Deep Neural Networks (DNN) for predicting stock price movement using historical stock data was presented in [17]. The proposed techniques perform favourably compared with traditional methods in terms of prediction accuracy. In the same way, Stoean et al. [20] applied an LSTM based predictive model to predict the closing-price of twenty-five (25) firms enlisted on the Bucharest Stock Exchange, using historical stock price. Notwithstanding the achievement recorded by authors, they acknowledged in their conclusion that the fusion of multiple stock price indicators can improve prediction accuracy. Also, a deep learning predictive framework using CNN and Recurrent Neural Networks (RNN) for predicting future stock price was proposed in [21]. The study reported some improvement in prediction accuracy when compared with analogous earlier studies.

Selvin et al. [22] implemented an LSTM, RNN and CNN based predictive framework for stock price prediction using historical stock prices as input parameters [22]. The proposed system successfully identified the relation within a given stock dataset. Yang et al. [23] proposed a multi-indicator feature-selection for CNN-driven stock index prediction based on technical indicators computed from historical stock data. The study outcome showed a higher performance of proposed deep learning technique than the benchmark algorithms in trading simulations. Additionally, Hiransha et al. [23], proposed a stock market predictive framework based on deep-learning models, like Multilayer Perceptron (MLP), RNN, LSTM and CNN, using past stock data as input features. Their results compared with AutoRegreesive Integrated Moving Average model (ARIMA) showed a higher performance of DNN over ARIMA. The reported outcomes of DNN in market analysis create an excellent platform for additional studies in a wide range of financial times-series prediction based on deep learning approaches. An enhanced SVM ensemble with genetic algorithm predictive model based on historical stock price was presented in [24]. The study outcome revealed that ensemble techniques offer higher prediction accuracy.

However, as mentioned earlier, the historical stock price is limited in disclosing all information about a firms' financial status. Also, as indicated in Zhou et al. [25], stock-prices are highly unstable; hence, using technical indicators only cannot exclusively capture the precariousness of price movements. Furthermore, the theory of behavioural finance shows that the emotions of investors can affect their investment decision-making [26]. Hence, unstructured stock market data enfolded in traditional news and social networking sites can serve as complementary to quantitative data to enhance predictive models, specifically in this age of social media and information technology.

Studies based on qualitative dataset

The effects of sentiments on stock market volatility have received recent attention in the literature [27–32]. One core source of information for sentiment analysis is the news articles [27, 28] and the other commonly used data source is the social media [33–36]. Using a Support Vector Machine (SVM) and Particle Swarm Optimisation (PSO), Chiong et al.

[31] proposed a stock market predictive model based on sentiments analysis. The study recorded a positive association between stock volume and public sentiment.

Similarly, Ren et al. [37] predicted the SSE 50 Index with public sentiment and achieved an accuracy of 89.93% using SVM. Likewise, Yifan et al. [38] examined the predictability of stock volatility based on public sentiment from online stock forum using RNN. They reported a positively high correlation between public sentiments and stock price movement. A combination of three predictive models, namely SVM, adaptive neuro-fuzzy inference systems and Artificial Neural Networks (ANN) was proposed for stock price prediction, using public sentiments [39]. Evaluation of the proposed model with historical stock index from the Istanbul BIST 100 Index yielded promising results. Maqsood et al. [40] examined the predictability of stock price movement from four countries based on sentiments in tweets and reported a high association between stock price and tweets.

The quest for improvement in prediction accuracy has led to the examination of additional data source lately. The following studies [9, 41–43] probed the effect of web search queries on stock market volatility and reported that web search queries could effectively predict stock price volatility. However, search queries are limited to territory where the user is searching from; hence its effects on stock price movement cannot be generalised.

The limitation of previous studies discussed above is that they relied only on a single stock-related data source, which, according to [8] limits predictive power.

Studies based on both qualitative and quantitative datasets

The combination of different data sources to enhance the prediction accuracy of predictive models has increased in recent studies. The combined effect of a user's sentiments from social media and Web news on stock price movement was examined [1]. The study achieved prediction accuracy between 55 and 63%. Also, the authors reported a high association between stock price movement and public sentiments. Also, Zhang et al. [7] proposed an extended coupled hidden Markov stock price prediction framework based on Web news and historical stock data. In [8], the authors proposed Multi-source multiple instance learning framework, based on three different data sources. The study recorded an increase in accuracy by the multiple data sources compared with distinct sources.

Table 1 shows a summary of pertinent works that sought to examine the collective influence of different stock-related information sources on stock price volatility. We examine these studies based on the number of data source, the technique used, the origin of stock data and reported results.

Table 1 affirms earlier report [2] that for every 122 studies on stock market prediction, 89.38% uses a single data source, while 8.2 and 2.46% use 2 and 3 data sources respectively. Then again, as pointed out in the same report, a comprehensive stock market prediction framework should capture all possible stock price indicators that influence the market. Also, in a review paper [10] on the responses of the stock market to information diffusion, explicitly acknowledged that the accuracy of predictive models in the stock market analysis had improved significantly in recent years. Despite that, there is room for further enhancement, by discovering newer sources of information on the Internet

Table 1 A summary of related studies

Reference	Technique	No. of data source	Input data source	Sock data	Reported Results
[1]	Co-evolving tensor-based learning	2	W & SM	China A-share and HK stock	55–63%
[7]	extended coupled hidden Markov	2	W & HSD	China A-share	52–63%
[44]	NN, LR, SVM, KNN, RF, AB & KF	2	HSD & MD	NS	NS
[45]		2	W & HSD	S&P 500 SPY index	p-value better than 0.05
[24]	CNN and RNN	2	HSD & W		
[8]	Multi-Source Multiple Instance Learning	3	HSD, SM & W	China A-share	62.1%
[43]	Delta Naive Bayes (DNB)	3	W, SM & GS	Argentina, Peru & Mexico	p-value (0.583–0.702)
[9]	ANN	3	W, SM and GS	Ghana	Accuracy (49.4 – 77.12)%

HSD historical stock data, *W* = Web news, *SM* = social media, *MD* = macroeconomic data, *NN* = Neural Networks, *LR* = Logistic Regression, *KNN* = K-Nearest Neighbor, *RF* = Random Forest, *AB* = AdaBoost, *KF* = Kernel Factor, *NS* = not stated, *GS* = Google search volumes

to comprehend the existing. Additionally, Pandurang et al. [46] pointed out that different data amalgamation strategies are future directions for better stock market predictions.

Therefore, a holistic fusion of several quantitative and qualitative stock-related data sources to predict the future stock price is a potential way to improve prediction accuracy [2, 10, 46–48], which remains an open research area. Hence, this study put-forward a novel multi-source data-fusion stock market predictive framework built on a deep hybrid neural network architecture (CNN and stacked LSTM) named IKN-ConvLSTM, to produce a more reliable and accurate stock price prediction. On the other hand, different from previous works that commonly exploit single or dual or triple data source, our proposed framework effectively integrates six (6) heterogeneous stock-related information sources.

Methodology

Our objective is to enhance the prediction accuracy, using both quantitative and qualitative stock-related information as input features to a hybrid DNN architecture. We present in detail the methods and techniques used in this study under this section.

Study framework

Figure 1 shows the process flow of our proposed IKN-ConvLSTM framework for predicting stock price movement. The framework follows five (5) steps: datasets download, data preparation, data fusion, machine learning model, and model evaluation. Details of our framework are explained below.

Datasets

Figure 2 shows the used data sources in this study. All datasets for this study was download from January 3, 2017, to January 31, 2020.

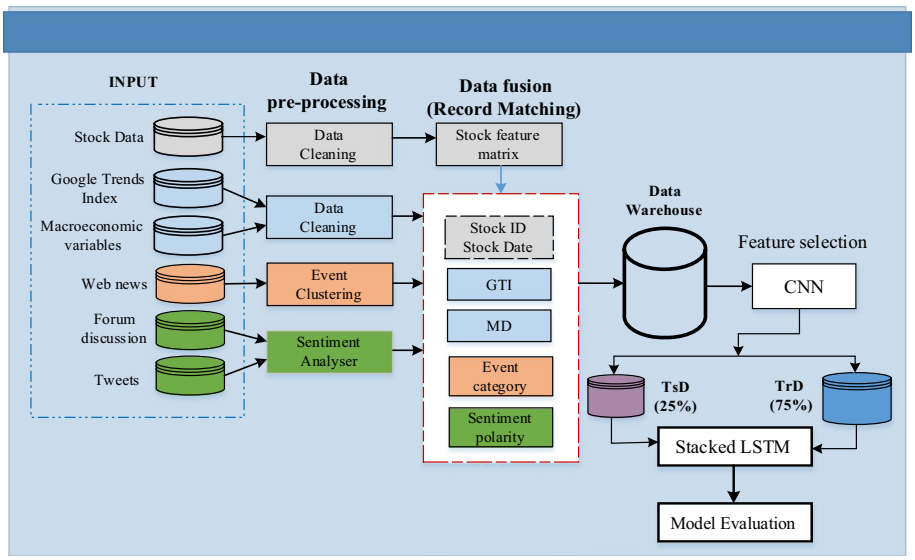

Fig. 1 Process flow of the proposed IKN-ConvLSTM framework

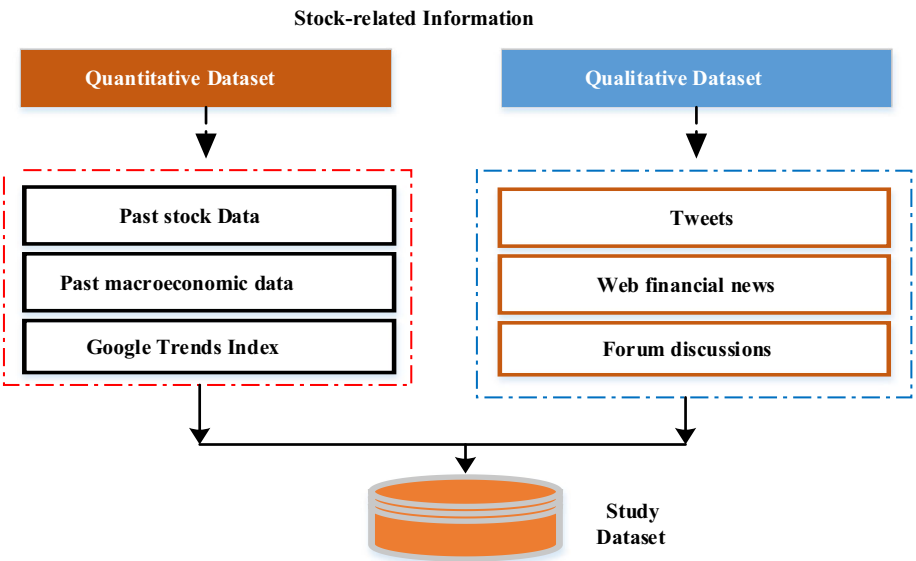

Fig. 2 Proposed heterogeneous sock-related information data sources

Quantitative dataset

As shown in Fig. 2, three quantitative datasets were used in this study; namely, historical stock data (HSD), macroeconomic data (MD) and Google trends index (GTI).

The historical stock price data of two companies listed on the GSE was downloaded from (https://gse.com.gh). We selected these companies because they had minimal missing values in their dataset (HSD). Also, these companies were more discussed in the news and social media platform, which gave the researchers adequate qualitative information on them. Each dataset had ten (10) features and 744 trading days; thus, the downloaded stock dataset was a matrix of size 744×10. Details are given in

Table 2. Similar to several studies [29, 32, 49–51], we aimed at stock returns $\left(R_d^{sk}\right)$ as defined in Eq. (1). Therefore, we normalised $\left(R_d^{sk}\right)$ to reflect the stock-price change compared with the day-before price. We denormalised our model output to get the real-world stock price as expressed by Eq. (2). If $R_d^{sk} > 0$ it implies a rise in *(d)* day's closing price (denoted as 1) and if $R_d^{sk} < 0$ it represents a fall in *(d)* day's closing price, denoted as 0 defined in Eq. (3).

$$R_d^{sk} = \frac{stock\_\text{price}_{(d)} - stock\_\text{price}_{(d-1)}}{stock\_price_{(d-1)}} \tag{1}$$

$$stock\_\text{price}_{(d)} = stock\_price_{(d-1)}\left(R_d^{sk} + 1\right) \tag{2}$$

$$\text{Target } (\hat{y}) = \begin{cases} 1 & \text{if } R_d^{sk} > 0 \\ 0 & \text{else} \end{cases} \tag{3}$$

where $stock\_\text{price}_{(d)}$ = closing price at day *(d)*

Previous studies have shown that fundamental macroeconomic such as inflations, price level, interest rate, exchange rate and composite consumer price are good indicators for stock price movement. Therefore, similar to these studies [44, 52], we downloaded forty-four (44) economic indicators from the official websites of the Bank of Ghana (www.bog.gov.gh) for 744 trading days. Table 2 shows the details of the macroeconomic variable used in this study. Study shows that the accuracy of deep learning algorithms is deeply affected by data quality [3, 12, 53, 54]. Therefore, for better performance of our model, we replaced any missing value of specific MD feature on a day (d) with $x_{(di)}$ as defined in Eq. (4). The dataset was normalised in the range of [− 1,1], using Eq. (5). We save each qualitative dataset separately in a CSV file.

$$x_{(di)} = \frac{x_{(d-1)} + x_{(d+1)}}{2} \tag{4}$$

where $x_{(d-\ 1)}$ = specific MD feature value on the previous day and $x_{(d+1)}$ = value on a day after missing day

$$x_{\text{new}i} = \left(\frac{x_{i\text{original}} - \overline{x_i}}{\sigma}\right) \tag{5}$$

Table 2 Breakdown of fused features

Data source	Number of features	Percentage (%)
Twitter (SM)	6	8.57
Web news (W)	5	7.14
Forum discussions (FD)	4	5.71
Macroeconomic data (MD)	44	62.86
Historical stock data (HSD)	10	14.29
Google trends (GTI)	1	1.43
Total	70	100

where (x_{newi}) is the normalised feature, $x_{i\text{original}}$ = the original values of feature (x), $\overline{x_i}$ and σ are the mean and standard deviation of the dataset *(x).*

Google trend is a service provided by Google, which enables anyone to find out the volume of search on any topic. The search volumes are usually scaled within [0–100], where 100 represent the highest search volume for any given day and 0 the lowest. A total of 221 records were obtained from Google Trends, thus, 221×1 matrix, and we normalised the dataset in the range of [− 1,1] as defined in Eq. (5). The trend search for this study was restricted to only the two companies of focus. Google trend was considered as a potential input because studies show that it can effectively communicate the future volatility of the stock price [41–43].

Qualitative (textual) dataset

Three qualitative datasets, as shown in Fig. 2, were used in this study, namely tweets (SM), web financial news (W) and forum discussion (FD). The tweets used in this study were downloaded from Twitter, using the Twitter API Tweepy [55]. Moreover, like many works in literature [33–36], we used the dollar ($) sign as a means to obtained 1,101 stock market-related tweets and all other tweets concerning our selected companies. Business news, financial news and events headlines concerning our selected companies we downloaded from three popular news sites in Ghana, namely, ghanaweb.com, myjoyonline.com and graphic.com.gh using the BeautifulSoup API. A total of 251 news articles were downloaded. However, unlike previous works [8, 27, 28] which considered only the sentiments in news titles, this study considered the spread of the news among the public and counts of comments made by the public on a news article on the same day. Thus, we excluded comments, and shears counts made any day after the day the news article was published. The reason is using the number of comments and share on an article days after its publication could lead to the use of information occurring after the stock price movement has already taken place. We extract the actual sentiments in the news titles using the Natural Language Toolkit (NLTK) [56].

We obtained our forum discussions dataset from sikasem.org. We use the sentiment analyser [56] to obtain the collective sentiments from the forum messages. All our qualitative datasets were tokenised, segmented, normalised, and freed from noise. Thus, texts were chopped into smaller pieces, called tokens while throwing away certain characters such as punctuation, symbols (URLs, /,?,#, @), extra spaces and stop words like "and," "a" and "the", using the NLTK. We assessed the sentiments in the textual datasets (tweets, news, and forum discussions) in two dimensions, polarity score within the range [− 1.0, 1.0] and subjectivity within the range [0.0, 1.0], where 0.0 is considered to be very objective and 1.0 as very subjective [9]. We also considered diffusion of a tweet and forum message by considering retweeting of a tweet and number of comments made on a forum post. We stored each processed textual data in a separate comma-separated values file for further processing. Table 1 (Appendix A) shows the details of features extracted from the textual dataset for this study.

Data fusion

The fusion stage aims to integrate the six (6) datasets discussed above, based on stock ID and stock price date.

Definition 1: Historical stock data (HSD): we represented HSD features as a matrix of 3-dimensions (i.e. stock ID's (S_{ID}), stock date *(d)* and quantitative features). Thus, for each stock *(k)*, we denoted its quantitative features as a vector *(x_k)*, where $x_k = \{x_{k1}, x_{k2}, x_{k3}, ..., x_{kN}\}$, N is the number of features, x_{kN} is the values of the N^{th} feature. The historical stock data was represented as $X \in \Re^{M \times N}$, where M is the number of stocks.

Definition 2: Google Trends Index (GTI): The GTI dataset is represented by a vector $G \in \Re^{L \times B}$, where B = features of $GTI\{G_{ID,}d, I\}$, GID = unique ID assigned to each GTI record, d = GTI date, I = quantitative value of GTI.

Definition 3: Macroeconomic data (MD): We represent MD by a vector $M_{data} \in \Re^{P \times Q}$, for every (M_{data}) its quantitative feature is represented by where $x_Q = \{x_{Q1}, x_{Q2}, x_{Q3}, ..., x_{QP}\}$ on date (d), Q is the number of feature (44 for this study), x_{pQ} = values on Q^{th} feature, P is the number of records.

Definition 4: Web financial news (W): Let a news article N_a on a date *(d)* be represented by a u-dimensional vector $x_{N_a d} \in \Re^{u \times 1}$, such that the k^{th} news observation of stock (S_{ID}) at date *(d)* can be defined as $x_{N_a, S_{ID}, k} = (x_{N_a}, S_{ID}, d)$, where (x_{N_a}) is the Web news features, S_{ID} = stock ID and d = the date of the news event.

Definition 5: Tweet sentiment (SM): We represent the sentiment extracted from the social media as a vector on a date *(d)* as $x_{SD,d} \in \Re^{S \times T}$, such that the T^{th} SM observation of stock (S_{ID}) at date *(d)* can be defined as $x_{SD, S_{ID}, T} = (x_{SD}, S_{ID}, d)$, where (x_{SD}) is the SM sentiment features, S_{ID} = stock ID and d = the date of a social media message.

Definition 6: Forum Discussion (FD): Let vector $x_{FD,d} \in \Re^{W \times Z}$ represents the sentiment extracted from the FD on a date *(d)*, such that the Z^{th} FD observation of stock (S_{ID}) at date *(d)* can be defined as $x_{FD, S_{ID}, Z} = (x_{FD}, S_{ID}, d)$, where (x_{FD}) is the FD sentiment features, S_{ID} = stock ID and d = the date of a forum discussed message, W = total records.

Finally, to make use of theses six heterogeneous sources, we put-forward a feature fusion framework (Fig. 3) to combine all features using (Algorithm 2, Appendix A). We considered each data source as independent of the another. The stock returns labels are denoted by $y = \{y_d\}$, where y_d represents the stock return class on a date *(d)*. Let vector φ holds the final amalgamation of the six defined vectors above. We apply a strategy for merging all feature from the six data source as a single vector, which can be defined as $\varphi_d = \{\beta_i\}_d, i \in \{1, ..., d\}$, where (β_i) is the combination of six data source observed on the day *(d+1)*, $\beta_i = \{x_i, g_i, x_{Qi}, x_{N_a i}, x_{SDi}, x_{FDi}\}$. Then the prediction problem can be modelled mathematically as a function $f(\varphi) \rightarrow y_{t+d}$. Thus, the combined dataset could be expressed as $(\varphi_d \in \Re^{M \times N})$, where N is the total number of features (N = 70 for this

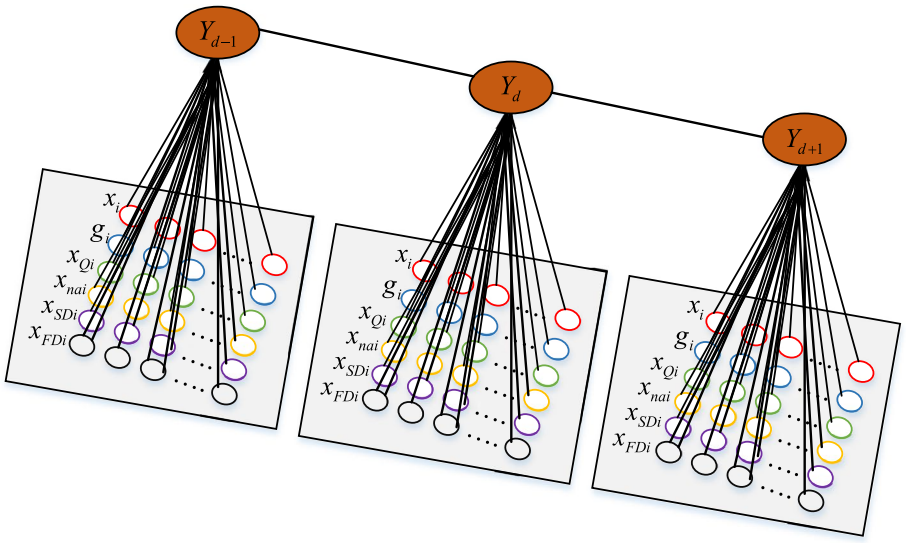

Fig. 3 Data source fusion

study) as shown in (Table 6, Appendix A), M is the number of records. Table 2 shows the breakdown of the initially features of our integrated dataset. The final dataset was a matrix of size 193×70.

Model design Recently, deep learning techniques have gained unprecedented popularity, and several accomplishments can be found in the literature [12, 13, 54, 57]. Therefore, in this study, we introduce a CNN figuration as a feature selection mechanism to select the features that are most significant to feed our LSTM classifier. The following section gives details of the proposed hybrid predictive model.

Feature engineering with CNN

Almost every machine-learning model is integrated with feature selection, to eliminate redundant and irrelevant features among datasets for higher performance in terms of prediction accuracy and computational time [29, 32, 51]. Lately, the CNN algorithm is one among deep learning techniques used in feature selection and extraction [11, 58]. Currently, literature has shown a promising performance of stock market predictive models built on the CNN algorithm [59]. In this paper, a CNN with 1 Convolutional Layer (CL), two dense layers and a MaxPooling was implemented to perform a random search feature selection. This network has 64 filters and kernels of size 2. We placed a pooling layer with max-pooling function (MPF) and ReLU activation to extract unique features after the CL. The MPF layer addresses the essential features by pooling over any feature map bearing a close similarity to the practice of feature selection in finding investments patterns. The ReLU activation function was adopted in this study for its easy implementation and vantage of nimbler convergence. Finally, two dense layers with ReLU and Sigmoid respectively are placed after a flatten layer. We adopted a simple and straight-forward criterion proposed in [60], to detect which features are to be selected or removed. The process utilises the accuracy obtained by the network on the training

dataset. Thus, assumed a trained CNN network *(N)* with input data (g) of *(d)* dimensional of features, $g \rightarrow \{g_1, g_2, ..., g_n\}$. The accuracy of *(N)* is calculated with one less feature, using the cross-entropy error function (Eq. 6). At the same time, a penalty term measures the complexity of the (N). Thus, the set $g - \{g_k\}$, for each $k = 1, 2, ..., n$ is the input feature set. We then calculate the accuracy *(A)* by simply assigning the connection weights from the input feature $\{g_k\}$ of trained *(N)* to zero (0). Afterwards, we ranked the obtained accuracies of each *(N)* with $g - \{g_k\}$ features, and based on the network having the maximum accuracy, the set of features to be reserved is searched. The steps for the CNN feature selection are detailed in algorithm 1.

$$F(w, v) = -\left(\sum_{i=1}^{k}\sum_{p=1}^{c} t_p^i \log S_p^i + \left(1 - t_p^i\right) \log\left(1 - S_p^i\right)\right) \tag{6}$$

where k=number of patterns $t_p^i = 1$ or 0 and is the target value for pattern x^i and the output unit p, p=1,2,...C, C=number of output units, S_p^i = the output of the (N) at unit p

Algorithm 1:

Initialise:

$g \rightarrow \{g_1, g_2, ..., g_n\}$, features to the CNN
g $\rightarrow (DS_{Train}, Ds_{CV})$ training and cross- validation
$\Delta M \rightarrow$ accetable maximum drop of accuracy rate of (DS_{Train})
N $\rightarrow$ network

1. Train (N) to minimise the loss with (g) so that the accuracy rate of training set is acceptable
2. for $i = 1, 2, ..., n$, N_i has the weight from input g_i as 0 and
   weights from other inputs equal to weights of network (N)
3. Calculate the accuracy rate (A_{Traini}) of (DS_{Train}).
4. Calculate the accuracy rate (A_{Testi}) of (Ds_{CV}).
5. Rank networks (N_i) by their A_{Traini}.
6. Calculate $u \rightarrow \Delta\ (A_{Testi})$
7. for each (N_i) from $i = 1$, if $(u) \leq M$, remove (g_i) from input set (g) and $N = N$-1.
If $i < N$, $i++$, Next.
Else stop.

The acceptable maximum drop in the accuracy rate (ΔM) on the (Ds_{CV}) set was set to 2%.

LSTM classifier

In this stage, we introduced a special RNN named LSTM for predicting the stock price movement. The LSTM was invented to solve the overfitting problem of the simple RNN [17, 18, 52]. Figure 4 shows an elaborate scheme of a single LSTM block architecture.

The significant element of the LSTM is the cell state, (C_t), which is regulated by three different gates, namely forget-gate (f_t), input-gate (i_t) and output-gate (o_t). The main computation of the LSTM is as defined in Eq. 7 – 14 [12, 17, 18, 22, 52]. The forget-gate decides to keep or throw away a piece of information from the previous cell state (expressed in Eq. 7) using a sigmoid function (Eq. 8), $f_t \in [0, 1]$.

$$f_t = \sigma\left(W_f(h_{t-1}, v_t) + b_f\right) \tag{7}$$

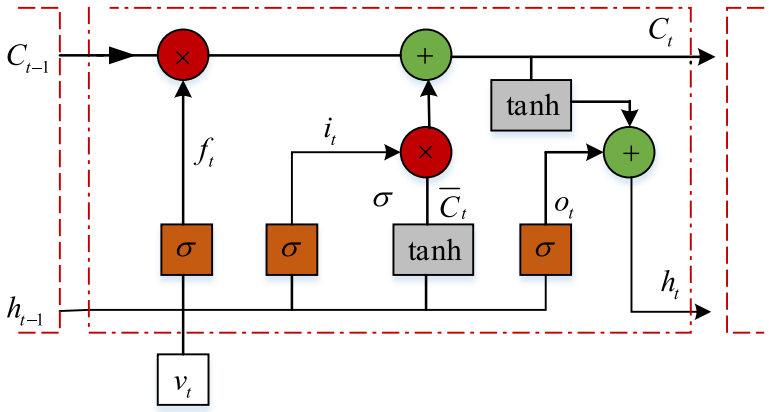

Fig. 4 A single LSTM block architecture

$$S(\sigma) = \frac{1}{1 + e^{(-1)}} \tag{8}$$

The i_t expressed by Eq. (9), determines which values of the cell state are restructured by an input signal, based on sigmoid function, and Hyperbolic tangent *(tanh)* layer (Eq. 10) and create a vector value $(\overline{C}_t)$(expressed in Eq. 11). $i_t \in [0, 1]$

$$i_t = \sigma(W_i(h_{t-1}, v_t) + b_i) \tag{9}$$

$$\tanh = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right) \tag{10}$$

$$\overline{C}_t = \tanh(W_k(h_{t-1}, v_t) + b_k) \tag{11}$$

$$C_t = f_t C_{t-1} + i_t \overline{C}_t \tag{12}$$

The output-gate (o_t) expressed by Eq. (13), permits the cell state either to affect other neurons or not. This is achieved by passing the cell state through a *tanh* layer and multiply it with the outcome of the output gate to get the ultimate output (h_t) defined by Eq. (14). $o_t \in [0, 1]$

$$o_t = \sigma(W_o(h_{t-1}, v_t) + b_o) \tag{13}$$

$$h_t = o_t \tanh(C_t) \tag{14}$$

where f_t = forget gate, i_t = input gate, c_t = update gate and o_t = output gate, W_f, W_i, W_k and W_o represent the weight metrices, b_f, b_i, b_k and b_o denotes the bias vectors, c_t = memory cell and σ = sigmoid activation function, h_{t-1} = LSTM target value at a past time step $t-1$

In this study, we designed a stacked-LSTM network (Fig. 5), which comprised (L_1 and L_2) to predict stock price movement from the optimised features by the CNN model. We implemented (L_1 and L_2) with different size, with ($L_1 > L_2$), a practice

common in literature [61] for detecting unique features in terms of specificity. By this, (L_1) is aimed at recognising general features while (L_2) is aimed at specific features. Knowing that the complexity of LSTM is influenced by the input data size and time steps, we designed (L_1) to accommodate 40 LSTM blocks. Each block linking to a timestep in our dataset to be supplied into our predictive network and $(L_2 = 20\,blocks)$. The preprocessed data from the CNN model is transformed into a 3-dimensional matrix $\left(x \in \Re^{l \times m \times n}\right)$,where l = batch size, m = sequence length and n = features and fed into (L_1). The output $(h(L_1))$ of (L_1) is forwarded to (L_2) and the output $(h(L_2))$ of (L_2) is passed through a SoftMax-layer *(SL) (*defined in Eq. (15) and (16)) to transform the output into two class probabilities $(Y \in [1, 0])$:

$$\mathrm{p}^{(d)} = softmax\left(h_{L_1}^{d} W_{softmax} + b_{softmax}\right) \tag{15}$$

$$\mathrm{softmax}\left(\mathrm{y}_{[1]}\right) = \frac{\mathrm{e}^{\mathrm{y}_{[1]}}}{\mathrm{e}^{\mathrm{y}_{[1]}} + \mathrm{e}^{\mathrm{y}_{[0]}}} \tag{16}$$

We adopted Adam (Adaptive Moment Estimation) with the initial learning rate of 0.001 to train our network. The Adam combines the strength of 2 other optimisers, namely ADAgrad and RMSprop. The grid search technique was used for hyperparameters tuning, where numerous amalgamations of hyperparameter values were tried, and the best amalgamation adopted. Table 3 gives a summary of the optimum hyperparameters used in each NN layer in this study. Only ten epochs were used in our LSTM training as the dataset (no. of records) was very small.

Evaluation metrics In examining the performance of our proposed stock prediction framework, we adopted the Accuracy (Eq. 17), Specificity (Eq. 18), F-score (Eq. 19) and Sensitivity (Eq. 20) metrics, based on their suitableness for measuring the performance of a classification model as indicated in [2, 62]. Accuracy gives a measure of the correctly classified samples to the total number of samples. Specificity estimates the classifier's capability to correctly identify negative labels while sensitivity (also known as recall)

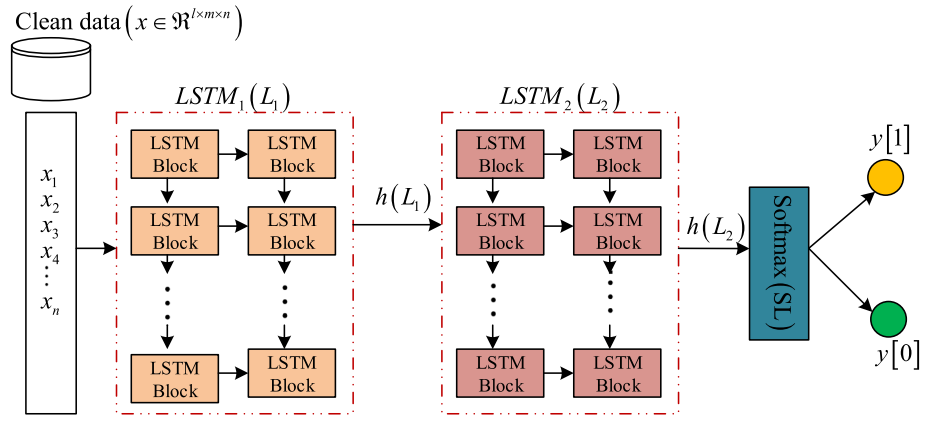

Fig. 5 Proposed LSTM network architecture

Table 3 A summary of study hyperparameters

Parameters	CNN	LSTM
Input layer	1	
Input feature dimension	1–70	62
Dense Layers	2	2
Output Layer	1	1
Dropout rate		0.2
Epoch	100	10
Activation	ReLU/ sigmoid	Tanh/ sigmoid functions
Weight	Normal [1, 1]	Normal [0,1]
Optimiser	Adam	Adam
Learning rate	0.002	1e-3—1e-4
Objective function	Cross-entropy	Cross-entropy

determines the capability of the classifier to classify positive labels correctly. Also, the F-score is a measure of the model's accuracy on the dataset [2, 62].

$$Accuracy = \frac{TN + TP}{FP + TP + TN + FN} \tag{17}$$

$$Specificity = \frac{TN}{TN + FP} \tag{18}$$

$$F - score = \frac{2 \times TP}{2 \times (TP + FP + FN)} \tag{19}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{20}$$

where, FN = incorrectly rejected (is false negative), TP = *correctly identified (true positive),* TN = correctly rejected (true negative), FP = incorrectly identified (false positive).

Empirical Implementation

A practical implementation of the proposed predictive framework (IKN-ConvLSTM) was carried out to assess its performance. The computer used was an HP laptop (Spectre × 360) computer 8th Generation Intel® Core™ i7 processor 16.0 GB RAM. We implemented our model with the Keras library, which supports both the Graphics Processing Unit (GPU) and the Central Processing Unit (CPU). The framework was coded in a modular fashion using Python programming language with Jupyter notebook. We also made use of the numerous modules in Keras such as cost functions and optimisers for implementing deep learning algorithms. To obtained an optimal data portioning of our integrated dataset discussed in Sect. 3.3, we adopted the in-sample and out-of-sample test technique, and the optimal split was training (75%), and testing (25%). Based on the training and testing dataset, we trained and tested our proposed model using the optimum hyperparameters. Table 4 shows a summary of our CNN features selection model.

Table 4 Summary of CNN Model

Layer (type)	Output Shape	Param #
conv1d_20 (Conv1D)	(None, 26, 64)	192
max_pooling1d_20 (MaxPooling)	(None, 13, 64)	0
flatten_20 (Flatten)	(None, 832)	0
dense_39 (Dense)	(None, 50)	41,650
dense_40 (Dense)	(None, 1)	51
Total params: 41,893 Trainable params: 41,893 Non-trainable params: 0		

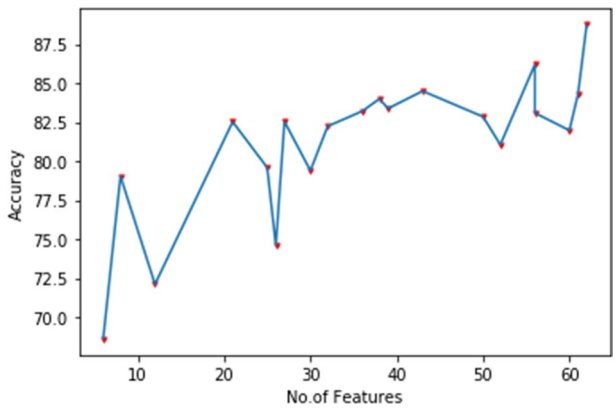

Fig. 6 CNN output for twenty different randomly selected features

Empirical results and discussions

Feature engineering by CNN

Figure 6 shows the accuracy of twenty (20) iterations of different randomly selected features by our CNN model. We observed that 21 features gave an accuracy of 82.52%, while 52 features recorded an accuracy of 81.06%, as shown in Fig. 6. However, the combination of 62 features measured an accuracy of 88.75%, which was the best combination by the CNN model. Nevertheless, another combination of 60 features recorded an accuracy of 81.97%. Thus, a difference of 6.78% in accuracy between 60 and 62 features. Thus, this outcome points out that the performance of a machine learning model does not depend on the quantity of input feature, but the quality of the input features. Thus, it can be inferred from the outcome that combining the right stock price indicators out of the numerous indicators from different stocks related data sources is a good phenomenon for higher prediction accuracy. Thus, not just amalgamation of several features increases prediction but the right ones. Furthermore, this outcome affirms the importance of feature engineering in a machine learning framework, as indicated in [29, 32, 51]. Based on these outcomes, it can be established that the CNN networks are enough and efficient for automatic selection of features from heterogeneous stock data for effective stock price prediction.

The optimised parameter from our base CNN was fed as input to our stacked LSTM model. Details of the best 20 pairs of features and their accuracies recorded by the CNN model is given in Table 7 (Appendix A).

Training and testing results based on the optimised features

The proposed predictive framework was training and tested using the accuracy and loss metrics. The accuracy in this study signifies the number of data samples whose labels were correctly classified by our predictive model, measured as already expressed in Eq. (17). The loss here signifies an error, which indicates how close the predicted values $(\hat{y})$ are to the actual label (y). Figure 7 shows a plot of how the proposed predictive framework performed during training and testing over ten epochs, based on optimised fused features from the CNN base model. From Fig. 7, it can be observed that the training accuracy progressively upsurges and converges around 98.526%, while the testing converges around 98.307%.

The progressive rise in the training accuracy of the proposed predictive framework shows that our stack LSTM classifier acquires better-quality optimised parameters over individual epoch till convergence. Also, the high training accuracy (98.526%) achieved at convergence suggests that the first phase (*LSTM1*) of our proposed stacked LSTM networks was capable of automatically detecting unique features within the 62 input features. Furthermore, the simultaneous progressive rise in both training and testing accuracies points out that the trained predictive framework is not having a variance problem. As an alternative to viewing the performance of the proposed framework, Fig. 8 shows the training and testing losses. Subsequently, the smaller loss values recorded during training and testing show the efficacy of the proposed model. Thus, the lesser the loss value at convergence, the better a model is since loss signifies a measure of error. At convergence, training and testing loss were 0.09264 and 0.04958, respectively.

Figure 9 shows a plot of all textual dataset ($SM + W + FD$) put together as (*Unstructured Dataset*) and all numerical ($MD + HSD + GTI$) put together as (*Structured Dataset*) and a combination of both as (*All Combine*). We aimed at exploring in details the ideology that traditional technical analysis combined with the sentiments or opinions

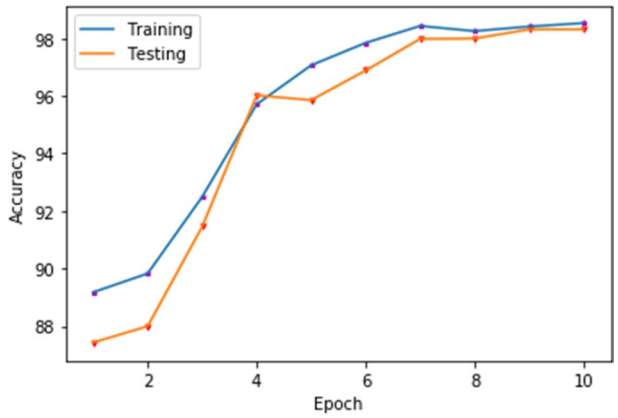

Fig. 7 Training and testing accuracy of proposed framework

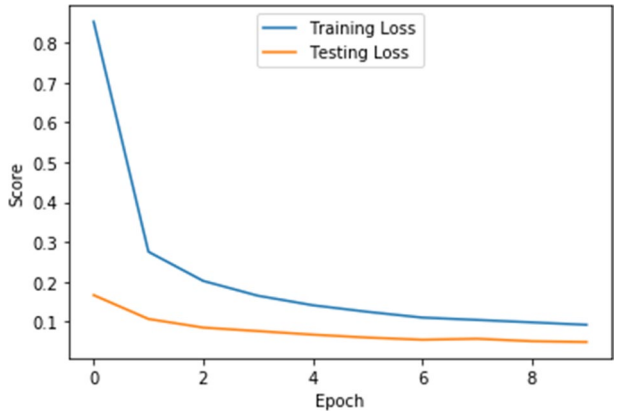

Fig. 8 Training and testing losses

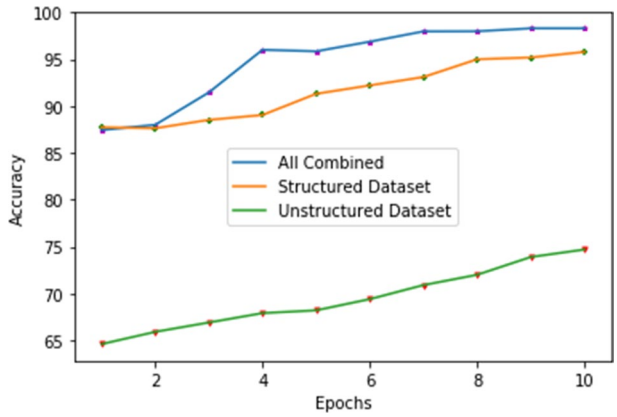

Fig. 9 Accuracy plots of structured, unstructured and combined stock information for stock market prediction

of investors and experts (fundamental analysis) will give better stock price prediction results.

As shown in Fig. 9, the unstructured dataset achieved a convergence accuracy of 74.69%, whiles the structured dataset achieved 95.78% and combined dataset 98.526%. This outcome confirms two opinions in literature. Thus, (1) the difference in accuracy (21.09%) between structured dataset (95.78%) and unstructured dataset (74.69), affirms that the unstructured stock dataset from social media and the Internet are best for argumentation of historical or structured stock dataset to enhance prediction [2, 5]. (2) also, an increase in accuracy of combine dataset compared with the individual (structured and unstructured), supports that a combination of stock-related information has the propensity of improving stock prediction accuracy as pointed out in [2, 10, 46, 47]. Hence, it cannot be overlooked in designing stock prediction frameworks and models. However, we observed that the accuracies of the structured and combined datasets were initially close to each other. However, the gap widens as the epochs increased. Table 5 shows the experimental results for specificity, F-score and sensitivity (recall) of the proposed predictive framework. The results (Table 5) show the effectiveness of the

Table 5 Specificity, precision and recall results

Dataset	Specificity	F-score	Sensitivity
Unstructured dataset	0. 758	0. 6083	0.7445
Structured dataset	0.9743	0.9397	0.9229
All combine	0.9975	0.9672	0.8939

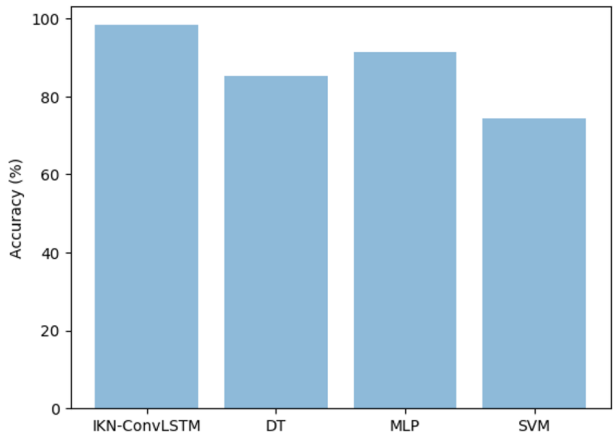

Fig. 10 A comparison of the proposed framework with other techniques

proposed predictive model to correctly identified positive and negative labels. However, from Table 5, it is evident that the neural NN model handles negative label labels a little better than positive labels.

Ref. [1, 7] combined two data sources to predict stock price movement and reported an accuracy of (52–63) %. Also, in [9], three data sources were joined to predict future stock price and achieved prediction accuracy (70.66–77.12)%. In comparison, the current study achieved a prediction accuracy of 95.78% with a combination of six different data sources. The outcome suggested that the accuracy of the stock market prediction can be improved further with data source fusion.

Comparison of the proposed framework with other techniques

Figure 10 shows a plot of prediction accuracies of the proposed framework (IKN-ConvLSTM) compared with Multi-Layer Perceptron (MLP), classical SVM and Decision Trees (DT). We implemented an MLP with three hidden layer (HL), HL1 and HL2 (with 50 nodes), and HL3 (with 30 nodes), maximum iteration = 5000, optimizer = Limited-memory BFGS (lbfgs), activation = relu. The classical SVM parameters were as follows: kernel = Radial Basis Function (RBF), and regularisation (C) = 100. The DT setting were, max_depth = 4 and criterion = entropy. The already implemented MLP, SVM and DT in Scikit-learn library were used for simplicity. Using tenfold cross-validation, the MLP, SVM and DT were trained and tested with the same preprocessed data from the CNN. Their average testing accuracies recorded are MLP (91.31%), classical SVM (74.31%) and DT (85.31%). From the comparative outcome (Fig. 10) the proposed (IKN-ConvLSTM) technique contests well with the other classical techniques (MLP, SVM and DT).

The accuracy of IKN-ConvLSTM outperformed the MLP, SVM, and DT models by 7, 24 and 13% respectively. It indicates that classical classifier models such as DT and SVM cannot effectively extract hidden features in input parameters. Besides, the overfitting problem may occur in training the DT and SVM models owing to the insufficient amount of data used in this study. In contrast, the ability of deep learning models to shear knowledge among nodes (neurons) can reduce the influence, as shown by the proposed deep learning framework.

Conclusions

Previous studies [1, 7, 8, 25, 43–45] have attempted to examine the joint impacted of different stock-related information sources for predicting stock price movement, a high percentage (63%) of these studies employed 2 data sources. In comparison, 37% used 3 data sources (see Table 1). However, current studies [2, 10, 46, 47] on stock price prediction acknowledge that the combination of different stock related data sources has the potential of recording higher prediction performance. However, literature shows that as datasets are becoming bigger, complex and more diverse, there is a big challenge to integrate them into an analytical framework. Besides, if this is overlooked, it will create gaps and lead to incorrect communications and insights. Hence, in this study, a novel framework called IKN-ConvLSTM was proposed. The model was based on a hybrid deep neural networks architecture of a convolutional neural network and long short-term memory to predict stock price movements by using a combination of six heterogeneous stock related data source. Using a novel combination of random search technique and a CNN base model as a feature selector, we optimised our initial training parameters of 70 heterogeneous stock related features from six different stock-related information sources. The final optimised parameters fed into a stacked LSTM classifier to predict future stock price. Our CNN model selected sixty-two (62) features with an accuracy of 88.75%. Which shows that the combination of CNN network and random search technique is useful for automatic feature selection from raw stock data, avoiding the need for manual feature selection in predicting stock price movement. Thus, the random search was found to be a powerful tool to perform feature selection. Stock price prediction accuracy (98.307%) achieved by our proposed stacked LSTM classifier with 62 different input features, shows that the accuracy of stock price predictive framework can be effectively enhanced with data fusion from different sources.

To the best of our knowledge, this study is the first to fuse six heterogeneous stock related information source to predict the stock market. Even though our proposed unified framework recorded satisfactory prediction performance, it still has some weaknesses. First, our framework has many parameters (62) which resulted in training time and computational resources, due to the nature of the deep neural network, compared to other methods. Secondly, though our dataset had a good number of parameters because of the data fusion introduced in this study, the size (volume) of textual data on the stock market in developing economy is scanty, which limited the prediction window of this study to only 30 days ahead. Also, much time was spent by researchers in integrating the six data sources as a single data, because they were of different formats and not in the same sequence. Again, removing comments made on news articles a day after the news was made manually taking much time. Therefore, future works could automate this

process and introduce some data argumentation techniques such as Generative Adversarial Networks (GANs), Autoencoders to enhance the current framework.

Also, a combination of different optimisation techniques to reduce training time while improving prediction accuracy for different trading windows is an excellent approach to be considered in future. Furthermore, incorporation of all the various stock price indicators in a single predictive framework, if done from a deductively-based approach, leads to a requirement to model social interaction, a unique challenge in itself. Whether future studies in this filed will go down that path or not remains to be seen. However, opinions and arguments about the depth of alertness and understanding drawn from a highly-quantitative approach (as typically employed in information fusion frameworks) will likely have to be balanced with the intuitions that can be gained from more social-hypothetical and subjective approaches in future research.

Abbreviations

GSE: Ghana stock exchange; MDF: Multi-source data-fusion; DNN: Deep neural networks; RNN: Recurrent neural networks; MLP: Multilayer perceptron; ARIMA: AutoRegreesive integrated moving average; SVM: Support vector machine; PSO: Particle swarm optimisation; ANN: Artificial neural networks; HSD: Historical tock data; W: Web news; SM: Social media; MD: Macroeconomic data; NN: Neural networks; LR: Logistic regression; KNN: K-Nearest neighbor; RF: Random forest; AB: AdaBoost; KF: Kernel factor; NS: Not stated; GS: Google search volumes; GTI: Google trends index; NLTK: Natural language toolkit; FD: Forum discussions; CL: Convolutional layer; GPU: Graphics processing unit; CPU: Central processing unit; DT: Decision trees; GANs: Generative adversarial networks.

Acknowledgements

Not applicable.

Authors' contributions

IKN obtained the datasets for the research and performed the initial experiments. IKN, AFA and BAW contributed to the manuscript development modification of study objectives and methodology. All authors contributed to the editing and proofreading. All authors read and approved the final manuscript.

Funding

Authors did not receive any funding for this study.

Availability of data and materials

The datasets used and/or analysed during the current study are publicly available.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Department of Computer Science and Informatics, The University of Energy and Natural Resources, Sunyani, Ghana.
[2] Department of Computer Science, Sunyani Technical University, Sunyani, Ghana.

Appendix A

See Appendix Tables 6, 7.

Table 6 Textual Datasets feature

Features	Description	Abbreviation
Textual datasets		
Tweets from Twitter		
	A unique ID of the tweet	
0. Tweet Sentiment	Tweet sentiment	TWS
1. Tweet Subjectivity	The separated subjectivity from the tweet	TWS
2. Tweet Polarity	The separated polarity from the tweet	TWP
3. Favourite count	Number of favourites per tweet	TWF
4. Retweet count	Total number of retweets	RTC
5. Possible sensitive	The sensitivity of the tweet (Boolean true/false)	TPS
Financial web-news		
6. News Sentiment	Sentiment in news	NWS
7. News Subjectivity	Separated subjectivity from news sentiments	NWSS
8. News Polarity	Separated polarity from news sentiments	NWP
9. Shared	number of sheared counts	NSC
10. Comments	Total number of comments on the news by the public	NCC
Forum discussions		
11. Forum Sentiment	Sentiment in forum discussions	FMSS
12. Forum Subjectivity	Separated subjectivity from forum sentiments	FMS
13. Forum Polarity	Separated polarity from forum sentiments	FMP
14. Forum Comments	Total number of comments on a topic posted on a forum	FMCC
Numerical datasets		
Search engine queries		
15. Google Trend Index		GTI
Macroeconomic variable		
16.	Monetary Policy Rate	MPR
17.	91-Day Treasury Bill Interest Rate Equivalent	TB91
18.	182-Day Treasury Bill Interest Rate Equivalent	TB182
19.	Inter-Bank Weighted Average	IBWA
20.	Ghana Reference Rate	GRR
21.	Average Commercial Banks Lending Rate	ACBLR
22.	Average Savings Deposits Rate	ASDR
23.	Average Time Deposits Rate: 3-Month	ATDR
24.	Private Sector Credit	PSC
25.	Capital Adequacy Ratio	CAR
26.	Non-Performing Loans	NPL
27.	Return on Equity (ROE)—After Tax	ROE
28.	Return on Assets (ROA)—Before Tax	ROA
29.	Core Liquid assets to total assets	CLATA
30.	Core Liquid assets to short-term liabilities	CLASTL
31.	Credit to Deposits (From Aug 2018, credit excl loans under receivership)	CD
32.	Headline Inflation (%) Yearly Change	HI
33.	Food Inflation (%) Yearly Change	FI
34.	Non-Food Inflation (%) Yearly Change	NFI
35.	Core Inflation (Adjusted for Energy & Utility) (%) Yearly Change	CI
36.	Bank of Ghana Composite Index of Economic Activity (Nominal Growth)	BoGCIEA1
37.	Bank of Ghana Composite Index of Economic Activity (Real Growth)	BoGCIEA2
38.	International Cocoa Price (US$/Tonne)—Monthly Average	ICP
39.	International Gold (US$/fine ounce)—Monthly Average	IG

Table 6 (continued)

Features	Description	Abbreviation
40.	International Brent Crude Oil (US$/Barrel)—Monthly Average	IBC
41.	Gross International Reserves (In Million US$)	GIR
42.	Net International Reserves (In Million US$)	NIR
43.	Merchandise Exports (f.o.b) (In Million US$)	ME
44.	Merchandise Imports (f.o.b) (In Million US$)	MI
45.	Trade Balance (In Million US$)	TB
46.	Currency outside banks (In Million Ghana Cedis)	COB
47.	Demand deposits (In Million Ghana Cedis)	DD
48.	Savings & Time deposits (In Million Ghana Cedis)	STD
49.	Foreign currency deposits (In Million Ghana Cedis)	FCD
50.	Reserve Money (RM) (In Million Ghana Cedis)	RM
51.	Narrow Money (M1) (In Million Ghana Cedis)	M1
52.	Broad Money (M2) (In Million Ghana Cedis)	M2
53.	Total Liquidity (M2 +) (In Million Ghana Cedis)	M2 +
54.	Gross External Debt/GDP (%)	GED
55.	Gross Domestic Debt/GDP (%)	GDD
56.	Gross Public Debt/GDP (%)	GPD
57.	Inter-Bank Exchange Rate—Month Average (GHC/US$)	GHC-USD
58.	Inter-Bank Exchange Rate—Month Average (GHC/GBP)	GHC-GBP
59.	Inter-Bank Exchange Rate—Month Average (GHC/EURO)	GHC-EURO
Historical stock data		
60.	Year High	YH
61.	Year Low	YL
62.	Previous Closing Price	PCP
63.	Opening Price	OP
64.	Closing Price	CP
65.	Price Change	PC
66.	Closing Bid Price	CBP
67.	Closing Offer Price	COP
68.	Total Shares Traded	TST
69.	Last Transaction Price	LTP

Table 7 CNN output for twenty different randomly selected features

S/N	No. of Features	Selected Feature	Acc.
1	62	[24, 9, 23, 54, 43, 48, 31, 35, 65, 13, 1, 46, 49, 47, 42, 6, 58, 29, 36, 14, 45, 38, 64, 34, 21, 41, 3, 32, 20, 19, 59, 44, 25, 4, 26, 50, 63, 2, 61, 30, 27, 57, 10, 55, 39, 12, 7, 62, 51, 16, 11, 0, 18, 5, 53, 28, 60, 15, 8, 37, 33, 52]	0.8875
2	56	[63, 13, 47, 5, 54, 40, 48, 44, 58, 53, 18, 22, 34, 41, 24, 45, 3, 52, 46, 62, 4, 29, 19, 8, 28, 9, 42, 10, 7, 50, 6, 2, 61, 12, 37, 39, 21, 30, 51, 26, 16, 20, 43, 35, 31, 15, 23, 36, 64, 56, 32, 17, 25, 55, 0, 27]	0.8619
3	43	[27, 44, 64, 15, 60, 53, 11, 13, 46, 22, 54, 32, 41, 56, 23, 59, 36, 24, 55, 26, 62, 31, 6, 16, 12, 43, 33, 7, 45, 50, 63, 28, 29, 14, 61, 49, 19, 48, 17, 30, 18, 5, 34]	0.84493
4	61	[33, 31, 35, 64, 54, 14, 32, 44, 36, 58, 45, 34, 57, 65, 9, 29, 37, 0, 7, 24, 59, 49, 10, 23, 26, 56, 11, 42, 48, 4, 18, 39, 51, 40, 13, 20, 50, 63, 17, 6, 38, 47, 28, 53, 15, 1, 43, 60, 22, 21, 62, 30, 27, 46, 5, 8, 3, 55, 41, 52, 16]	0.8430
5	38	[62, 27, 53, 21, 26, 35, 2, 43, 49, 31, 59, 40, 57, 47, 38, 33, 46, 48, 44, 54, 3, 34, 39, 10, 61, 19, 36, 24, 8, 17, 55, 60, 58, 52, 56, 64, 13, 22]	0.8399
6	39	[14, 46, 44, 41, 65, 13, 56, 10, 23, 47, 25, 42, 58, 59, 36, 6, 22, 29, 61, 33, 51, 2, 52, 55, 21, 53, 7, 40, 34, 19, 20, 45, 1, 5, 49, 50, 4, 30, 26]	0.8338
7	36	[60, 26, 35, 64, 20, 30, 42, 8, 37, 15, 49, 54, 61, 32, 13, 41, 53, 3, 18, 57, 38, 39, 29, 9, 1, 36, 23, 45, 48, 51, 31, 33, 17, 12, 55, 11]	0.8320
8	56	[24, 38, 39, 40, 27, 50, 47, 29, 52, 53, 1, 4, 34, 6, 46, 59, 18, 37, 61, 11, 0, 26, 57, 19, 5, 43, 25, 9, 58, 7, 62, 15, 21, 33, 31, 45, 41, 36, 44, 3, 10, 51, 48, 65, 49, 64, 30, 16, 8, 35, 32, 12, 17, 13, 55, 23]	0.8309
9	50	[38, 1, 7, 2, 44, 18, 26, 20, 22, 16, 17, 3, 27, 12, 37, 62, 32, 61, 30, 14, 5, 45, 24, 56, 10, 50, 8, 43, 57, 59, 48, 28, 34, 53, 9, 46, 15, 23, 25, 65, 42, 4, 13, 40, 55, 33, 29, 19, 58, 60]	0.8285
10	27	[2, 43, 63, 9, 5, 23, 53, 55, 18, 65, 21, 15, 59, 51, 44, 6, 50, 29, 19, 17, 8, 56, 26, 60, 57, 54, 46]	0.8254
11	21	[53, 23, 45, 61, 63, 27, 4, 14, 59, 22, 2, 25, 8, 50, 35, 20, 32, 3, 47, 13, 62]	0.8252
12	32	[2, 42, 60, 63, 26, 9, 31, 56, 17, 62, 28, 3, 20, 49, 53, 55, 24, 29, 14, 65, 41, 27, 46, 44, 12, 45, 0, 33, 43, 11, 30, 48]	0.8224
13	60	[53, 10, 17, 3, 50, 61, 30, 65, 2, 33, 9, 7, 49, 16, 35, 41, 32, 34, 62, 28, 38, 46, 24, 42, 31, 14, 11, 19, 12, 4, 21, 29, 52, 44, 37, 63, 56, 5, 13, 51, 6, 0, 1, 64, 25, 22, 47, 58, 26, 23, 45, 43, 20, 48, 15, 39, 40, 18, 59, 57]	0.8197
14	52	[63, 5, 4, 43, 53, 14, 16, 7, 26, 17, 56, 31, 41, 2, 49, 22, 32, 11, 30, 60, 25, 39, 23, 48, 6, 50, 54, 18, 13, 37, 62, 36, 0, 29, 20, 8, 47, 21, 1, 38, 28, 19, 12, 46, 27, 40, 34, 64, 55, 3, 45, 15]	0.8106
15	25	[40, 27, 15, 25, 59, 39, 11, 31, 50, 19, 54, 36, 34, 56, 14, 24, 65, 13, 35, 0, 7, 32, 51, 29, 62]	0.7959
16	30	[27, 20, 45, 34, 62, 65, 11, 38, 64, 16, 61, 2, 21, 15, 52, 23, 18, 39, 12, 46, 37, 0, 51, 19, 14, 4, 32, 41, 7, 6]	0.7941
17	8	[6,13,14,16,43,44,60,61]	0.7904
18	26	[38, 49, 52, 7, 19, 17, 59, 30, 18, 20, 33, 45, 42, 27, 25, 26, 63, 62, 39, 24, 46, 5, 29, 54, 10, 50]	0.7458
19	12	[55, 57, 46, 22, 45, 23, 26, 49, 64, 35, 6, 16]	0.7212
20	6	[57, 63, 37, 40, 44, 17]	0.6860

Algorithm 2: Algorithm for contruction of data fusion

Initialise datasets and counters :
$DS_{HSD}, DS_{MD}, DS_{SM}, DS_{W}, DS_{FD}, DS_{GTI}$
Temporal dataset $D_1, D_2, D_3, D_4, D_5, D_6$
$N \leftarrow$ *total records* (DS_{HSD})
$DS_{all} \leftarrow$ *Combined dataset dataframe*
$q \leftarrow 0$
Order all datasets by data :
for $i \leftarrow 0$ *to* N *do*
  $date \leftarrow (DS_{HSD}\_date[i])$
  $D_1 \leftarrow DS_{HSD}\_Row[i]$
  for $j \leftarrow 0$ *to* $Total(DS_{MD})$ *do*
    if $(DS_{MD}\_date[i] = date)$
    then
      $D_2 \leftarrow DS_{MD}\_Row[i]$
    end
  end
  for $k \leftarrow 0$ *to* $Total\ (DS_{SM})$ *do*
    if $(DS_{SM}\_date[i] = date)$
    then
      $D_3 \leftarrow DS_{SM}\_Row[i]$
    end
  end
  for $m \leftarrow 0$ *to* $Total\ (DS_{W})$ *do*
    if $(DS_{W}\_date[i] = date)$
    then
      $D_4 \leftarrow DS_{W}\_Row[i]$
    end
  end
  for $n \leftarrow 0$ *to* $Total\ (DS_{FD})$ *do*
    if $(DS_{FD}\_date[i] = date)$
    then
      $D_5 \leftarrow DS_{FD}\_Row[i]$
    end
  end
  for $p \leftarrow 0$ *to* $Total\ (DS_{GTI})$ *do*
    if $(DS_{GTI}\_date[i] = date)$
    then
      $D_5 \leftarrow DS_{GTI}\_Row[i]$
    end
  end
  if any not empty $(D_1, D_2, D_3, D_4, D_5\ \&\ D_6)$
  then
    $DS_{all}[q].append(D_1, D_2, D_3, D_4, D_5\ \&\ D_6)$
  end
  Reset $(D_1, D_2, D_3, D_4, D_5\ \&\ D_6) \rightarrow empty$
  $q \leftarrow q + 1$
end

Received: 16 July 2020 Accepted: 14 December 2020
Published online: 09 January 2021

References

1. Zhang X, Zhang Y, Wang S, Yao Y, Fang B, Yu PS. Improving stock market prediction via heterogeneous information fusion. Knssowledge-Based Syst. 2017;143:236–47. https://doi.org/10.1016/j.knosys.2017.12.025.
2. Nti IK, Adekoya AF, Weyori BA. A systematic review of fundamental and technical analysis of stock market predictions. Artif Intell Rev. 2020;53:3007–57. https://doi.org/10.1007/s10462-019-09754-z.
3. Guiñazú MF, Cortés V, Ibáñez CF, Velásquez JD. Employing online social networks in precision-medicine approach using information fusion predictive model to improve substance use surveillance: A lesson from Twitter and marijuana consumption. Inf Fusion. 2020;55:150–63. https://doi.org/10.1016/j.inffus.2019.08.006.
4. Giraldo-forero F, Cardona-escobar F, Castro-ospina E. Hybrid artificial intelligent systems. Cham: Springer International Publishing; 2018. https://doi.org/10.1007/978-3-319-92639-1.
5. Huang J, Zhang Y, Zhang J, Zhang X. A tensor-based sub-mode coordinate algorithm for stock prediction. In: 2018 IEEE third international conference on data science in cyberspace. IEEE; 2018. p. 716–721. doi: https://doi.org/10.1109/DSC.2018.00114
6. Guo Z, Zhou K, Zhang C, Lu X, Chen W, Yang S. Residential electricity consumption behavior: Influencing factors, related theories and intervention strategies. Renew Sustain Energy Rev. 2018;81:399–412. https://doi.org/10.1016/j.rser.2017.07.046.
7. Zhang X, Li Y, Wang S, Fang B, Yu PS. Enhancing stock market prediction with extended coupled hidden Markov model over multi-sourced data. Knowl Inf Syst. 2019;61:1071–90. https://doi.org/10.1007/s10115-018-1315-6.
8. Zhang X, Qu S, Huang J, Fang B, Yu P. Stock market prediction via multi-source multiple instance learning. IEEE Access. 2018;6:50720–8. https://doi.org/10.1109/ACCESS.2018.2869735.
9. Nti IK, Adekoya AF, Weyori BA. Predicting stock market price movement using sentiment analysis: evidence from ghana. Appl Comput Syst. 2020;25:33–42. https://doi.org/10.2478/acss-2020-0004.
10. Agarwal S, Kumar S, Goel U. Stock market response to information diffusion through internet sources : a literature review. Int J Inf Manage. 2019;45:118–31. https://doi.org/10.1016/j.ijinfomgt.2018.11.002.
11. Zhao B, Lu H, Chen S, Liu J, Wu D. Convolutional neural networks for time series classification. J Syst Eng Electron. 2017;28:162–9. https://doi.org/10.21629/JSEE.2017.01.18.
12. Karim F, Majumdar S, Darabi H, Harford S. Multivariate LSTM-FCNs for time series classification. Neural Networks. 2019;116:237–45. https://doi.org/10.1016/j.neunet.2019.04.014.
13. Karim F, Majumdar S, Darabi H. Insights into lstm fully convolutional networks for time series classification. IEEE Access. 2019;7:67718–25. https://doi.org/10.1109/ACCESS.2019.2916828.
14. Qu Y, Zhao X. Application of LSTM neural network in forecasting foreign exchange price. J Phys Conf Ser. 2019. https://doi.org/10.1088/1742-6596/1237/4/042036.
15. Chong E, Han C, Park FC. Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies. Expert Syst Appl. 2017;83:187–205. https://doi.org/10.1016/j.eswa.2017.04.030.
16. Zhu Y, Xie C, Wang GJ, Yan XG. Comparison of individual, ensemble and integrated ensemble machine learning methods to predict China's SME credit risk in supply chain finance. Neural Comput Appl. 2017;28:41–50. https://doi.org/10.1007/s00521-016-2304-x.
17. Liang X, Ge Z, Sun L, He M, Chen H. LSTM with wavelet transform based data preprocessing for stock price prediction. Math Probl Eng. 2019;2019:1–8. https://doi.org/10.1155/2019/1340174.
18. Kim T, Kim HY. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. PLoS ONE. 2019;14:e0212320. https://doi.org/10.1371/journal.pone.0212320.
19. Tian C, Ma J, Zhang C, Zhan P. A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network. Energies. 2018;11:3493. https://doi.org/10.3390/en11123493.
20. Stoean C, Paja W, Stoean R, Sandita A. Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations. PLoS ONE. 2019;14:e0223593. https://doi.org/10.1371/journal.pone.0223593.
21. Vargas MR, de Lima BSLP, Evsukoff AG. Deep learning for stock market prediction from financial news articles. In: 2017 IEEE international conference on computational intelligents virtual environment for measuremnt systems and applications. IEEE; 2017. p. 60–65.https://doi.org/10.1109/CIVEMSA.2017.7995302
22. Selvin S, Vinayakumar R, Gopalakrishnan EA, Menon VK, Soman KP. Stock price prediction using LSTM, RNN and CNN-sliding window model. In: 2017 International conference on advances in computer communication and informatics. IEEE; 2017. p. 1643–1647. https://doi.org/10.1109/ICACCI.2017.8126078.
23. Hiransha M, Gopalakrishnan EA, Menon VK, Soman KP. NSE stock market prediction using deep-learning models. Procedia Comput Sci. 2018;132:1351–62. https://doi.org/10.1016/j.procs.2018.05.050.
24. Nti IK, Adekoya AF, Weyori BA. Efficient stock-market prediction using ensemble support vector machine. Open Comput Sci. 2020;10:153–63. https://doi.org/10.1515/comp-2020-0199.
25. Oncharoen P, Vateekul P. Deep Learning for Stock Market Prediction Using Event Embedding and Technical Indicators. In: 2018 5th International conference on advanced informatics: concept theory and applications. IEEE; 2018. p. 19–24. https://doi.org/10.1109/ICAICTA.2018.8541310.
26. Zhou Z, Xu K, Zhao J. Tales of emotion and stock in China: volatility, causality and prediction. World Wide Web. 2018;21:1093–116. https://doi.org/10.1007/s11280-017-0495-4.
27. García-Medina A, Sandoval L, Bañuelos EU, Martínez-Argüello AM. Correlations and Flow of Information between The New York Times and Stock Markets. Phys A. 2018. https://doi.org/10.1016/j.physa.2018.02.154.

28. Xing FZ, Cambria E, Welsch RE. Allocation via market sentiment views. IEEE Comput Intell Mag. 2018;13:25–34. https://doi.org/10.1109/MCI.2018.2866727.
29. Souza TTP, Aste T. Predicting future stock market structure by combining social and financial network information. Phys A. 2019;535:122343. https://doi.org/10.1016/j.physa.2019.122343.
30. Alshahrani HA, Fong AC. sentiment analysis based fuzzy decision platform for the saudi stock market. In: 2018 IEEE international conference on electro/information technology. Rochester, MI: IEEE; 2018. P. 23–29. doi: https://doi.org/10.1109/EIT.2018.8500292
31. Chiong R, Fan Z, Adam MTP, Neumann D. A sentiment analysis-based machine learning approach for financial market prediction via news disclosures. In: genetic and evolutionary computation conference companion. Kyoto: ACM Press; 2018. P. 278–279. doi: https://doi.org/10.1145/3205651.3205682.
32. Wang Y, Li Q, Huang Z, Li J. EAN: event attention network for stock price trend prediction based on sentimental embedding. In: Proceedings of the 10th ACM conference on web science; 2019. p. 311–320. doi: https://doi.org/10.1145/3292522.3326014.
33. Pimprikar R, Ramachadran S, Senthilkumar K. Use of machine learning algorithms and twitter sentiment analysis for stock market prediction. Int J Pure Appl Math. 2017;115:521–6.
34. Checkley MS, Higón DA, Alles H. The hasty wisdom of the mob: how market sentiment predicts stock market behavior. Expert Syst Appl. 2017;77:256–63. https://doi.org/10.1016/j.eswa.2017.01.029.
35. Nisar TM, Yeung M. Twitter as a tool for forecasting stock market movements: a short-window event study. J Financ Data Sci. 2018;4:1–19. https://doi.org/10.1016/j.jfds.2017.11.002.
36. Maknickiene N, Lapinskaite I, Maknickas A. Application of ensemble of recurrent neural networks for forecasting of stock market sentiments Equilibrium-Quarterly. J Econ Econ Policy. 2018;13:7–27. https://doi.org/10.24136/eq.2018.001.
37. Ren R, Wu DD, Wu DD. Forecasting stock market movement direction using sentiment analysis and support vector machine. IEEE Syst J. 2019;13:760–70. https://doi.org/10.1109/JSYST.2018.2794462.
38. Liu Y, Qin Z, Li P, Wan T. Stock Volatility Prediction Using Recurrent Neural Networks with Sentiment Analysis. In: Benferhat S, Tabia K, Ali M, editors. Advances in Artificial Intelligence: From Theory to Practice. IEA/AIE 2017. Lecture Notes in Computer Science, vol. 10350. Cham, Springer; 2017. https://doi.org/10.1007/978-3-319-60042-0_22.
39. Oztekin A, Kizilaslan R, Freund S, Iseri A. A data analytic approach to forecasting daily stock returns in an emerging market. Eur J Oper Res. 2016;253:697–710. https://doi.org/10.1016/j.ejor.2016.02.056.
40. Maqsood H, Mehmood I, Maqsood M, Yasir M, Afzal S, Aadil F, Selim MM, Muhammad K. A local and global event sentiment based efficient stock exchange forecasting using deep learning. Int J Inf Manage. 2020;50:432–51. https://doi.org/10.1016/j.ijinfomgt.2019.07.011.
41. Neri K, Katarína L, Peter M, Roviel V. Google searches and stock market activity: evidence from Norway. Financ Res Lett. 2018. https://doi.org/10.1016/j.frl.2018.05.003.
42. Zhong X, Raghib M. Revisiting the use of web search data for stock market movements. Sci Rep. 2019. https://doi.org/10.1038/s41598-019-50131-1.
43. Fang J, Wei W, Prithwish C, Nathan S, Feng C, Naren R. Tracking multiple social media for stock market event prediction, In: Perner P, editor. Advances in data mining applications theory asp 17th ICDM. Cham: Springer International Publishing; 2017. p. 16–30. doi: https://doi.org/10.1007/978-3-319-62701-4_2.
44. Ballings M, Ldirk Poel VD, Hespeels N, Gryp R. Evaluating multiple classifiers for stock price direction prediction. Expert Syst Appl. 2015;42:7046–56. https://doi.org/10.1016/j.eswa.2015.05.013.
45. Geva T, Zahavi J. Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news. Decis Support Syst. 2014;57:212–23. https://doi.org/10.1016/j.dss.2013.09.013.
46. Pandurang GD, Kumar K. Ensemble computations on stock market: a standardized review for future directions. In: 2019 IEEE international conference on electrical computer and communicating technologies. IEEE; 2019. p. 1–6. doi: https://doi.org/10.1109/ICECCT.2019.8869158
47. Nguyen T, Yoon S. A novel approach to short-term stock price movement prediction using transfer learning. Appl Sci. 2019. https://doi.org/10.3390/app9224745.
48. Thakkar A, Chaudhari K. Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions. Inf Fusion. 2021;65:95–107. https://doi.org/10.1016/j.inffus.2020.08.019.
49. Ruan Y, Durresi A, Alfantoukh L. Knowledge-based systems using Twitter trust network for stock market analysis. Knowl Based Syst. 2018. https://doi.org/10.1016/j.knosys.2018.01.016.
50. Batra R, Daudpota SM. Integrating StockTwits with sentiment analysis for better prediction of stock price movement. In: 2018 international conference on computing, mathematics engineering and technology inventing innovative integration socioeconomic development ICoMET 2018—Proceedings 2018; Jan 2018. p. 1–5. doi: https://doi.org/10.1109/ICOMET.2018.8346382.
51. Picasso A, Merello S, Ma Y, Oneto L, Cambria E. Technical analysis and sentiment emb e ddings for market trend prediction. Expert Syst Appl. 2019;135:60–70. https://doi.org/10.1016/j.eswa.2019.06.014.
52. Nti IK, Adekoya AF, Weyori BA. Random forest based feature selection of macroeconomic variables for stock market prediction. Am J Appl Sci. 2019;16:200–12. https://doi.org/10.3844/ajassp.2019.200.212.
53. Rundo F. Deep LSTM with reinforcement learning layer for financial trend prediction in FX high frequency trading systems. Appl Sci. 2019;9:4460. https://doi.org/10.3390/app9204460.
54. Karim F, Majumdar S, Darabi H, Chen S. LSTM fully convolutional networks for time series classification. IEEE Access. 2017;6:1662–9. https://doi.org/10.1109/ACCESS.2017.2779939.
55. Roesslein J. Tweepy Documentation. 2009. Available: http://docs.tweepy.org/en/latest/.
56. Bird S, Edward L, Ewan K. Natural language processing with python. Newton: O'Reilly Media Inc.; 2009.
57. Guo Y, Wu Z, Ji Y. A hybrid deep representation learning model for time series classification and prediction. In: 2017 3rd international conference on big data computing and communications. IEEE; 2017. p. 226–231. doi: https://doi.org/10.1109/BIGCOM.2017.13

58. Zheng Y. Methodologies for cross-domain data fusion: an overview. IEEE Trans Big Data. 2015;1:16–34. https://doi.org/10.1109/tbdata.2015.2465959.
59. Yang H, Zhu Y, Huang Q. A multi-indicator feature selection for CNN-driven stock index prediction. In: lecture notes in computer science (including its subseries lecture notes in artificial intelligence lecture notes in bioinformatics. Springer International Publishing; 2018. p. 35–46. doi: https://doi.org/10.1007/978-3-030-04221-9_4.
60. Setiono R, Liu H. Neural-network feature selector. IEEE Trans Neural Netw. 1997;8:654–62. https://doi.org/10.1109/72.572104.
61. Borovkova S, Tsiamas I. An ensemble of LSTM neural networks for high-frequency stock market classification. J Forecast. 2019. https://doi.org/10.1002/for.2585.
62. Tharwat A. Classification assessment methods. Appl Comput Inform. 2018. https://doi.org/10.1016/j.aci.2018.08.003.

Publisher's Note