A deep learning method DCWR with HANet for stock market prediction using news articles

Saleh Albahli¹ · Awais Awan² · Tahira Nazir² · Aun Irtaza² · Ali Alkhalifah¹ · Waleed Albattah¹

Received: 14 August 2021 / Accepted: 17 January 2022 / Published online: 8 February 2022
© The Author(s) 2022

Abstract

Stock movement prediction is a challenging problem to analyze in both academic and financial research areas. The advancement of deep learning (DL) techniques has grasped the attention of researchers to employ them for predicting the stock market's future trends. Few frameworks can understand the financial terms in literature, and the volatile nature of stock markets further complicates this process. This paper has tried to overcome the existing challenges by introducing a DL-based framework using financial news articles to forecast the stock market. After performing preprocessing step, the deep contextualized word representation (DCWR) approach is applied for feature extraction. In the next step, the independent component analysis (ICA) method is used for feature reduction. Finally, the resultant features train the hierarchical attention networks (HANet) classifier to predict the stock movements. The proposed scheme is evaluated over the 7 years of data from a publicly available dataset gathered from the Reuter's website and attained an average prediction accuracy of 92.5% which shows our framework's robustness.

Keywords Deep learning · Stock prediction · Hierarchical attention networks · News articles

Introduction

The stock market is currently a significant source of information for the financial market. Speculators, especially investors always want to increase the chance of earning more profit with the help of market patterns analysis. In the past, it was an assumption in efficient market hypothesis (EMH) that the stock prices have all information related to stock patterns, and that the best possible and the most natural way for the stock market should be is random walk [1, 2]. However, there is an argument by the researchers in behavioral finance that hypothesis may be wrong due to the outrageous behavior of players, which can be affected by various types of market information and the psychological interpretation of information by the individuals [3]. Although the two theories differ from each other, both agree with the importance and effect of market information [4, 5].

Widely used and analyzed by investors, financial news is considered the most critical source of market information [6, 7]. Recent development in technology, primarily internet-based technologies has resulted in a tremendous increase in news broadcasting speed and the number of news articles [8]. Two examples can support this argument: (1) financial, software, data, and media companies like Bloomberg and Thompson Reuters can globally provide investors with real-time news related to the financial market; and (2) as compared to the news available in newspapers in the past, the online sources could provide probably thousand times more news [9, 10]. Processing and analysis of such a significant and increasing volume of information require modern computing resources equipped with efficient algorithms. Investors could easily filter out unnecessary information and make wiser decisions assisted by predictions given by support systems. Thus, modeling and analyzing market information for making better predictions becomes an exciting problem [11]. Researchers with computer science knowledge have studied this problem, and some of those works have categorized it as a classification problem [12–15]. Based on the recently released news articles, these works could predict a direction [sell, hold, buy] based on the recently released news articles. However, a problem with text classification techniques in this

✉ Saleh Albahli
Salbahli@qu.edu.sa

¹ Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia

² Department of Computer Science, University of Engineering and Technology-Taxila, Taxila 47050, Pakistan

Fig. 1 Expected scenarios vs. actual scenarios predicted by new articles as “Good news” and “Bad news”

scenario is that they ignore the critical information hidden in the stock market prices right before the news release. If we consider short-time history prices along with news articles, we can see that polarity of the news is not the only factor that indicates the stock prices [16, 17]. For instance, if there is primarily positive news related to a stock, it does not necessarily mean that the price will go up immediately; it might just indicate a flat price trend, stopping it from falling. Similarly, if there is mostly negative news related to a stock, it may not always predict the price as going down immediately, it might just make the price trend to appear flat [18]. Figure 1 provides a set of possible scenarios that could happen in the real world. It can be noted here that these scenarios are different from the traditional view in which an up-trend corresponds to a “good news” and a trend going down represents a “bad news”. To make these scenarios work and combine these multiple sources of information into a single system, we make use of multi-kernel learning (MKL) in our system. The MKL consists of two sub-kernels: one makes use of news articles and the other gets the short-time prices as input. The output of these sub-kernels is the weights that are used by the derived models in predicting the stock trends more accurately as compared to traditional methods [19].

The research community around the globe has been investigating and examining the changes in the stock markets. At the same time, the advancement in AI has urged researchers to employ DL-based models for predicting stock movements. However, there is still room for performance improvement due to the volatile nature of the stock market. This paper presents a robust DL framework for predicting stock market behavior by employing published financial articles. Following are the main contributions of our work:

1. Presented a learning-based approach namely DCWR for keypoints extraction, which enhanced the stock market prediction accuracy while minimizing both the training and testing time complexity.
2. A low-cost solution to stock market prediction due to the employment of the ICA is a feature reduction technique that provides a more representative set of keypoints by reducing the feature space.
3. Presented such a framework that provides consistent results under the varying stock market conditions.
4. Rigorous evaluation has been conducted compared to other state-of-the-art stock market predictions over a standard dataset comprising diverse and unstable nature data to show the robustness of the proposed solution.
5. Effective classification accuracy due to the hierarchal nature of the HANet which enables it to better deal with the over-fitted model training data.

The rest of the paper is organized as follows: “[Related work](#)” has a brief review of some significant works that have already been carried out in stock market directional prediction. A detailed overview of the proposed system is given in “[Proposed methodology](#)”. “[Experimental results](#)” provides a detailed report on the experiment results. Finally, we conclude this paper and give a glimpse of the future work in “[Conclusion](#)”.

Related work

Several approaches have been presented for stock trend forecasts using news articles. Seong et al. [20] proposed an approach to predict the stock market future trends from an analysis of financial news. The method [20] identified the related firms from the financial news employing data mining approaches namely K-means clustering and multiple kernels learning to reveal heterogeneity in the industry classification system. The approach in [20] exhibits better stock movements prediction accuracy. However, the system is only designed for Korean companies. Hao et al. [21] introduced a framework for stock market future price prediction using business news articles. This approach introduced a new twin SVM along with a fuzzy hyperplane that merged the classification power of SVM and fuzzy set theory to determine stock movement trends. The technique [21] worked well to tackle the outliers. However, performance needs to be improved. Another approach based on the investigation of business news articles has been presented in [22] for stock market movement prediction. Several stock-specific variables have been

extracted from the articles using a bag-of-words, noun sentences, and named articles. The computed features were used to train the SVM classifier to predict the future stock market trends. The approach in [22] works well for stock prediction, however, the results are reported over a small dataset.

Kaya et al. [23] presented an approach to predict future stock trends from published news articles. Initially, the published business articles together with price values were obtained. Then, delta prices per day were computed on which word couples comprising a noun and a verb were extracted as features. The computed key points were used to train the SVM classifier to perform the classification. The method [23] worked well for stock movement prediction, however, performance needs further improvements. Vargas et al. [24] proposed a DL-based framework namely recurrent convolutional neural network (RCNN) to determine the future stock movements. The method employed seven technical indicators as input which were calculated from the published business news articles. This method followed a two-stage procedure to show each article in the dataset. In the first step, the word2vec model was utilized to produce the word representation. Then in the next step, the average of all the word vectors with the same label was calculated. The computed features were used by the RCNN to predict future trends. The approach in [24] shows better stock prediction accuracy, however, at the expense of increased computational cost. Similarly, in [25], various ML-based approaches have been employed to predict the lifecycle of a selected firm using financial articles.

Hu et al. [26] presented a review of several ML and DL-based approaches used to predict future stock trends by analyzing financial articles. It is concluded in [26] that the existing approaches lack to apply a hybrid model which can improve the prediction performance. Dang et al. [27] introduced an approach for determining future stock movements through performing online business news analysis. After obtaining the news files from online sites, the content from the articles was mined to plain text. After this, the preprocessing step was applied to get a better representation of the dataset. In the next step, each article was labeled with a positive, negative, or neutral score on which further weightage policy was performed. Finally, the computed features were used to train the SVM classifier to achieve the future prediction. The approach [27] works well for stock market prediction. However, the method needs to be evaluated on standard and challenging datasets. Another research was presented in [28] to predict the future stock movements via employing several ML-based approaches namely Decision Tree (DT), Adaboost, Bagging, XGBoost, RNN, CNN, Random Forest (RF), LSTM, and Gradient Boosting. It is concluded in [28] that LSTM performs well for stock market prediction, however, at the expense of increased computational cost.

Schumaker et al. [29] proposed a solution to give the future stock prediction by employing the published financial articles. The work computed three types of features namely Bag of Words, Noun Phrases, and Named Entities from the articles on which the SVM classifier was trained to perform the classification. The approach is computationally efficient; however, performance needs further improvement. Another approach was presented in [30], in which the authors analyzed to describe market styles to enhance the stock future trends prediction accuracy under changing market styles. In the first step, stock time-series data were separated into varying window sizes. The obtained windows were shown with articles sentiment key points and technical indicators. In the next step, hierarchical clustering was applied to group the windows and classify the related market styles. Next, a distance metric was used to discriminate between rotating configurations within the market styles to validate the applicability of the market styles. Finally, the SVM classifier was trained over the computed features for stock market prediction. The method [30] exhibits better stock market prediction performance, however, not applicable for those market styles that lack the clustering properties. Similarly, Duarte et al. [31] proposed a technique to employ the news articles for predicting the stock movements. The technical indicator from the published news articles was used to train the several ML-based classifiers where the MLP and Naïve Bayes (NB-G) exhibited better performance. The approach [31] improves the stock future trends performance, however, it is specifically designed for the Brazilian Market.

Although extensive work has been presented to predict the future stock trends via employing financial news articles. However, these methods either use a small amount of data for model evaluation or testing their approach on local market data. Moreover, these studies require improvements in classification performance in tackling real-world problems. To deal with the aforementioned problems, there is a need for such a technique that can provide an efficient and effective solution to stock future trends prediction. In this work, we have tried to cover these gaps and presented such an approach that has enhanced the stock market future prediction performance and has evaluated over a large and diverse set of examples comprising on 12 years of historic data.

Proposed methodology

In this section, we introduce and explain the proposed method utilized for stock market prediction. Initially, the data preprocessing step is performed to clean the data and make it more proficient to be used in the later steps. Then, the DCWR technique is used to compute the features, on which the ICA algorithm is applied to reduce the feature space and obtain a more representative set of keypoints. Finally, the HANet

Fig. 2 Proposed method diagram

classifier is trained over the computed features for predicting the stock market prices. The whole flow of the proposed scheme is presented in Fig. 2.

Preprocessing

Most text and document datasets contain many unnecessary words such as stop words, misspelling, slang, etc.; in many algorithms, mainly statistical and probabilistic learning algorithms, noise, and unnecessary features can adversely affect system performance. For the said reason, we apply preprocessing before proceeding to the model training and testing phase. Significant preprocessing steps include tokenization, stop-word removal, spelling correction, and noise removal [32].

Stop-words

Text and document classification includes some words (e.g., the, and, for, a, about, after) which do not contain important significance to be used in classification algorithms. These semantically empty stop words are removed from the news articles. The remaining terms are usually semantically more significant and thus used as the textual representation. However, the removal of stop words from the target text is not a critical task in classification. It definitely helps in feature reduction resulting in a decently sized model.

Noise removal

In text analysis, most of the documents contain unnecessary things like punctuations and special characters (#, @, !

etc.). As this type of noise may affect the performance of the classifier, so we need to remove it from the dataset before proceeding to the model training phase.

Lowercasing

Converting the target text into its lowercase form is an essential step as repeated programming logic involves text comparisons. Hence it is better to convert the whole text into its lowercase format to make the comparison operations more accurate and efficient, resulting in a reduction in the execution time.

Tokenization

We perform tokenization to break a stream of text into phrases, words, or different tokens. The primary purpose of conducting this step is to break a sentence or set of sentences into manageable chunks for later steps involving iterative processing. An array-like structure can serve better for managing this type of data.

Stemming

This process deals with the inflections found in words (e.g., happiness, happily) by reducing them to their root form (e.g., happi) using an algorithm that uses a simple heuristic process. As described in the above example, the “root” does not necessarily mean the actual root word. It can just be a canonical form of the original word.

Stemming helps improve the efficiency of searching algorithms by standardizing the vocabulary. To bring up the most

relevant documents in the search process, we want to match or search for all word variations. Stemming helps us get this task done in a more efficient manner requiring less execution time.

Lemmatization

Lemmatization is similar to stemming, with the difference that it also maps the word to its original root form. For example, stemming will map “happiness” and “happily” to the root word “happy”, which is grammatically incorrect. Lemmatization, here, will map these words to their original root form which is “happy”.

Features extraction using DCWR

For feature extraction, we have used a deep learning-based approach that is DCWR [32, 33]. The DCWR approach employed a two-layered bidirectional LSTM framework and trained over the Billion Word Benchmark database to compute the word vector. The calculated word vector is known as Embeddings from Language Models (ELMo) [34]. The approach comprises both complicated attributes of word usage (i.e., format and semantics) and how their deployment differs in various linguistic settings (i.e., to model polysemy).

The basic objective of using the word embedding approach is that the output word vectors are computed using the bidirectional language model (biLM), which comprises both forward and backward language models (LMs).

Equation (1) presents the forward LM as follows:

$$p(t_1, t_2, t_3, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, t_2, t_3, \dots, t_{k-1}) \quad (1)$$

While the backward LM is presented by the Eq. (2).

$$p(t_1, t_2, t_3, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, t_{k+2}, t_{k+3}, \dots, t_N) \quad (2)$$

Both forward and backward LMs together enhance the log-likelihood in both directions as given in the following equation:

$$\sum_{k=1}^N \left(\log p(t_k | t_1, \dots, t_{k-1}; \Theta_x, \vec{\Theta}_{\text{LSTM}}, \Theta_s) + \log p(t_k | t_{k+1}, \dots, t_N; \Theta_x, \vec{\Theta}_{\text{LSTM}}, \Theta_s) \right) \quad (3)$$

Here Θ_x is showing the token representation, while the Θ_s is presenting the softmax layer.

In the next step, the job-specific weights from all the biLM layers are used to calculate the ELMo as follows:

$$\text{ELMo}_k^{\text{task}} = E(R_k; \Theta^{\text{task}}) = \gamma^{\text{tas}} \sum_{i=0}^L S_j^{\text{task}} h_{k,j}^{\text{LM}} \quad (4)$$

Here, the h is computed using the following equation:

$$h_{k,j}^{\text{LM}} = [\vec{h}_{k,j}^{\text{LM}}, h_{k,j}^{\text{LM}}] \quad (5)$$

In the Eq. (4), S^{task} is representing the softmax-normalized weight, while the γ^{task} is a scaling parameter.

Dimensionality reduction

After features extraction, we have used the feature reduction method because the resulted features suffer from the problem of high dimensionality that possesses significant computationally cost for text classification processing. For feature reduction, we employed the ICA method. ICA was introduced for signal processing and particularly for continuously distributed signals, however, it has also been utilized for textual data analysis [35, 36]. This technique was introduced in [37] and relies on a concept of statistical independence. ICA attempts to transform observed data into columns of independent components and considers high order statistical dependencies. As compared to PCA [38], ICA computes statistically independent linear projections that are not necessarily orthogonal to each other, therefore, it can find important representation for multivariate data. The ICA in vector–matrix notation can be defined as:

$$C = AV, \quad (6)$$

where V is independent variables, components expressed as $V = (v_1 + v_2 + v_3 + \dots + v_n)^T$. C is an observed mixture model represented as $C = (c_1 + c_2 + c_3 + \dots + c_n)^T$ obtained by multiplying A , where $A = (a_1 + a_2 + a_3 + \dots + a_n)$ is a constant $n \times n$ square mixing matrix. It can be expressed as:

$$C = a_1 v_1 + a_2 v_2 + a_3 v_3 + \dots + a_n v_n, \quad (7)$$

$$C = \sum_{k=1}^n a_k v_k. \quad (8)$$

Both A and V are learned by unsupervised method from observed data C . The objective of ICA is to compute V and A for a given C , where both A and V are statistically independent.

Fig. 3 Architecture of HANet

Classification

After features extraction, we have performed classification of news articles for prediction stock market using deep network namely (HANet). Our proposed technique is a deep model based on RNN and comprised of pyramids in which the outcome of the lower pyramids turns into the inputs to the higher pyramids. HANet [39] focuses on the document level classification that a document has K sentences, and each sentence contains T_i words, where w_{it} with t integral $[1, T]$ represents the words in the i th sentence. HANet structural design is explained in Fig. 3, where the lower stage encompasses word encoding and attention and the higher stage comprises sentence encoding and sentence attention (Table 1).

Each pyramid or level in HANet is comprised of a vigorous bidirectional LSTM or GRU with attention structures. LSTM or GRUs are employed for the reason that they consent

the system to selectively procedure input information built on how appropriate it is to the classification. Similarly, the attention structure is fed to allow the system to emphasize the LSTM or GRU outcomes accompanying with the arguments and lines that are most revealing of a specific class. We then established both models, i.e., LSTMs and GRUs through the hypermeter optimization process [40].

We utilized successive optimization through gradient boosted trees to discover the optimal hyperparameters for our model. This optimization scheme employs a gradient boosted tree-based regression approach to forecast the performance at new hyperparameter situations. We utilized this optimization method because it has been exhibited to meet more rapidly than usual Bayesian optimization. The hypermeter details are given in Table 2.

Table 1 Overview of existing methods

References	Technique	Limitation
[20]	K-means clustering along with multiple kernel learning	The approach applies to Korean stock only
[21]	A novel twin SVM classifier together with the fuzzy set theory	Performance needs improvements
[22]	The features, i.e., a bag-of-words, noun sentences, and named articles were used with the SVM classifier	Results are analyzed over a small dataset
[23]	The technical indicators like price values were utilized with the SVM classifier	Prediction accuracy needs further enhancement
[24]	A deep learning model, namely RCNN was employed	The approach is suffering from high computational costs
[25]	An analysis of several ML-based approaches has been presented	A more robust approach is required for better stock market trends prediction
[26]	A review of several ML and DL-based approaches was presented	The existing approaches lack to the application of the hybrid model, which can improve the prediction performance
[27]	The news article features along with the SVM classifier were used	More performance testing is required to be performed over a standard dataset
[28]	Numerous ML-based methods, i.e., DT, Adaboost, Bagging, XGBoost, RNN, CNN, Random Forest (RF), LSTM, and Gradient Boosting, were applied	The techniques are computationally expensive
[29]	The technical indicators from published articles, i.e., Bag of Words, Noun Phrases, and Named Entities, along with the SVM classifier, were used	Performance needs further improvement
[30]	The hierarchical clustering along with the SVM classifier was used	Not applicable for those market styles that lack clustering properties
[31]	The ML-based classifiers, i.e., the MLP and NB-G, were used	The method is designed only for the Brazilian Market

Table 2 Hyperparameters detail

Sr. no.	Hyperparameter	Value
1	Word embedding	DCWR
2	Size	100–500
3	RNN unit	GRU/LSTM
4	Number of hidden cells in each bidirectional RNN layer	50–200
5	Size of hidden layer in attention process	50–300
6	Dropout on ultimate article embedding	0.5 or none

Experimental results

In this section, we have implemented the introduced approach or model using the python platform. The large dataset is employed for the performance evaluation and results are calculated using the different evaluation parameters. To further evaluation of our technique, we conduct a comparative analysis with other models.

Dataset

This research work incorporates a publicly available dataset consisting of 207,902 financial news articles gathered from the Reuter's website [41] as shown in Table 3. These news articles were posted during the period of about 12 years from October 2006 to November 2018. The dataset contains three important columns, named title, content, and date of publishing. We aligned the publishing date of the news articles with the relevant financial news time series. The experiments carried out in [24, 41] showed that the titles of the news articles played a more significant role in predicting the stock prices as compared to the news contents. To verify this, we take both the news article title and news article content as input to the training model. Related to financial time-series data, we selected the Standard and Poor's 500 index series as the base time-series measure. This series is built using the Yahoo! Finance data posted during the same period that is used for taking the news articles. The purpose of selecting this source is that it is one of the most reliable sources as well as it contains the largest number of related news covering almost every major stock market worldwide. The information gained from this series served as the base for calculating the target output, and the same is used as input to the training model. In addition to its reliability and level of coverage, another purpose to select the employed dataset is that it is well suited to the nature of our problem domain due to its extensive content which better assists us to check the performance and ability of our model to deal with the volatile nature of stock market.

Related to the target output, we created a binary variable to indicate the expected out. The value [1,0] indicates that the closing price is expected to go up as during the next day

Table 3 Brand-wise distribution of the dataset

Stock market	Symbol	No. of news articles
Apple	AAPL	12,203
Microsoft	MSFT	17,585
Walmart	WMT	13,206
Nvidia	NVDA	13,123
Amazon	AMZN	15,389
Grand total		71,506

Table 4 Dataset demographics

Dataset	Complete	Training	Testing
From date	October 2, 2006	October 2, 2006	October 2, 2015
To date	November 21, 2018	October 1, 2015	November 21, 2018
Instances	4,160	3,076	1085
Total articles	71,506	50,055	21,451
Avg. News articles per day	16	15	19

compared to the closing price during the current day. Similarly, the value [0,1] indicates that the close price is expected to go down during the next day compared to the closing price reported the current day.

As this work provides a daily prediction, all news articles posted during the same day (instance) are aligned to represent one single day. However, the financial news in the general market normally contains a bunch of irrelevant information. This problem is also confirmed by the authors in [41]. To filter out this irrelevant information, we apply a filter that only selects the news articles directly related to a specific stock. The said filter is implemented using a python function which searches for the news articles from a news portal mentioning a specific stock name like #AAPL, #MSFT, etc. This filtering process resulted in a reduced dataset consisting of 71,506 news articles mentioning at least one of the stock symbols mentioned in Table 4. Finally, the days without any news are ignored and removed from the time series to simplify the process. Detailed demographics of the dataset are shown in Table 4.

The resulted reduced dataset consisting of 71,506 news articles with 4160 instances is our final dataset which is ready to be fed to the model for training and testing. However, we split the dataset in 70/30 ratio for training and testing purpose, respectively. The training split consists of 50,055 news articles with 3076 instances, while the testing portion contains a total of 21,451 news articles with 1085 instances.

Evaluation parameters

As mentioned in the previous section, we employ important ML/DL algorithms to observe their performance concerning evaluation matrices named accuracy, precision, recall, and $F1$ -measure.

We evaluated the performance of selected algorithms by various evaluation metrics named Accuracy, precision, recall, and F -measure. Accuracy is an essential categorization metric and able to be utilized for the evaluation of classification algorithms. Accuracy can be expressed as:

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{Total predictions}}. \quad (9)$$

Precision can be employed to determine the sample classification skill of the model can be computed as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (10)$$

where TP and FP are used to measure the true-positive rate and false-positive rate of the algorithm, respectively.

We can use recall to measure the ability of the classification paradigm to classify the highest potential examples. Recall can be calculated as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (11)$$

where FN denotes the false-positive rate of the algorithm.

$F1$ -measure is the combination of precision and recall and it is calculated as:

$$F1 - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (12)$$

Results

We evaluated our model on two-time intervals (i.e., daily, monthly, and yearly, respectively) on the dataset described in the previous section. Even if the prediction model is the same, the event-based approach performs better as compared to the bag-of-words (BoW) approach over both time intervals. This could be due to one or both of the following two reasons. As the events are predicate-argument structures, they carry the most important information, whereas the performance of the bag-of-words approach may be affected due to more irrelevant information. Secondly, information of actor and object is essential for stock market prediction, which is directly provided in structured events.

It is noteworthy here that uneven distribution of data or lesser data representation of a specific group can affect the model's performance of the model for that group, where performance for the brands with less data representation got

Fig. 4 RoC performance of the prediction model on yearly distributed data from 2006 to 2012

degraded to some extent which is also evident from Figs. 4 and 5.

The short-term volatility of stock prices can be determined by incorporating event information as an indicator. This can help improve the performance of the short-term prediction. This fact is supported in the presented Figs. 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15, where improved performance can be seen

in the daily predictions as compared to the monthly predicted values.

Summarizing the depictions in the above figures related to event-based distribution for five brands under experimentation, Table 5 presents a performance comparison of the training model concerning percentage accuracy for monthly and daily instances, respectively. It is evident from Table 5

Fig. 5 RoC performance of the prediction model on yearly distributed data from 2012 to 2018

that the daily prediction approach performed well as compared to the monthly prediction approach. Our experimental results confirmed that information embedded in the news takes approximately 12–24 h to reflect in the stock price response. The stock wise performance is presented in Fig. 16. Additionally, some events may have an immediate effect

on the stock prices. For example, in 2013, former CEO of Microsoft said that he would resign within a year. It took less than an hour for Microsoft shares to jump up to 9%. This fact demonstrates the possibility of the prediction of stock market prices for shorter than 1-day time intervals.

Fig. 6 Daily event-based prediction for AAPL

Fig. 7 Daily event-based distribution for MSFT

Table 5 Comparison of monthly and daily event-based approaches

Stock market	Mean accuracy % (monthly)	Mean accuracy % (daily)
AAPL	0.93	0.925
MSFT	0.95	0.93
WMT	0.91	0.92
NVDA	0.9	0.93
AMZN	0.89	0.96
Mean accuracy	0.916	0.933

Comparison with ML-based algorithms

We presented numerous experiments on the news article dataset using several ML-based models with our introduced technique. The techniques are trained for all sets of possibilities to attained outcomes. All the experimentations were done for three labeling techniques.

We have trained various ML-based classifiers namely Naïve Bayes [42], Random Forest [43], Logistic Regression [44], and Gradient Boosting [45] over the dataset, and obtained results are presented in Table 6. From the reported results, it can be seen that our approach is more robust to stock future trends prediction. More specifically, the accuracy values for Naïve Bayes, Random Forest, Logistic Regression, and Gradient Boosting models are 69%, 53%, 68%, and 30%

Fig. 8 Daily event-based distribution for WMT

Fig. 9 Daily event-based distribution for NVDA

Fig. 10 Daily event-based distribution for AMZN

Fig. 11 Monthly event-based distribution for AAPL

Fig. 12 Monthly event-based distribution for MSFT

Fig. 13 Monthly event-based distribution for AMZN

Fig. 14 Monthly event-based distribution for NVDA

Fig. 15 Monthly event-based distribution for WMT

Fig. 16 Stock-wise performance

Table 6 Comparison with ML-based classifiers

Technique	<i>F1</i> score	Accuracy (%)	Best time per sample (ms)	Average run time per sample (ms)
Naïve Bayes	0.41	69	2.031	2.415
Random forest	0.33	53	1.102	1.316
Logistic regression	0.30	68	2.150	2.374
Gradient boosting	0.11	30	1.224	1.483
Proposed (mean)	0.92	92.5	0.105	0.116

respectively, while our approach attains an average accuracy value of 92.5%. Similarly, the *F1* score for Naïve Bayes, Random Forest, Logistic Regression, and Gradient Boosting models is 0.51, 0.43, 0.40, and 0.21, respectively, whereas the presented technique has obtained the *F1* score of 0.92. As it can be observed that in terms of both evaluation metrics, our approach is more efficient due to the employment of HANet classifier which is capable of dealing with large datasets and model over-fitted training data. Moreover, we have performed the comparative analysis in terms of processing time containing both the best and average execution time and obtained results are reported in Table 6. More specifically, we have attained the average execution time of 0.116 ms which is lowest from all the competitor techniques. Hence, it can be concluded that our work is both efficient and effective to stock market future trends prediction.

Comparison with DL-based techniques

To further investigate the prediction accuracy of the proposed solution, we have compared it with DL-based classification techniques and results are presented in Table 7. From Table 7, it can be seen that our approach attains the highest accuracy and *F1* score with the values of 92.5%, and 0.92, respectively, while the BRET approach obtains the second highest accuracy and *F1* score with the values of 48% and 0.33, respectively. Moreover, the CNN with BERT embeddings shows the lowest accuracy and *F1* score of 23% and 0.17, respectively. The main reason for the proficient performance of the proposed solution is that the DCWR approach computes the more representative set of features which presents the word embeddings in more viable manners. Moreover, the employment of the HANet classifier assists to perform better prediction accuracy due to its hierarchal nature as the lower layers include both word encoding and attention while the upper layers comprise sentence encoding and attention. Such architecture of the HANet classifier enables it to better deal with the large feature space and can better tackle the model over-fitted training data. Moreover, the comparative approaches employ a very deep network which can easily encounter the problem of model over-fitting and are suffering from high computational cost. Therefore, it can be concluded

Table 7 Comparison of DL-based techniques

Technique	<i>F1</i> score	Accuracy (%)
BERT [46]	0.33	48
FinBERT [46]	0.32	33
CNN with Word2Vec [46]	0.26	42
CNN with BERT embeddings [46]	0.17	23
Word2Vec with BiLSTM and attention [46]	0.21	46
Fasttext with BiLSTM and attention [46]	0.21	47
FinALBERT with BiLSTM and attention [46]	0.21	46
Proposed (DCWR with HANet)	0.92	92.5

Table 8 Comparison with other techniques

References	Accuracy (%)
Xu et al. [47]	54.58
Deepika et al. [48]	90.42
Vignesh et al. [50]	66.83
Sadorsky et al. [49]	90.00
Proposed	92.50

that our framework is more efficient and effective in stock market prediction than the peered approaches.

Comparison with state-of-the-arts methods

Here, we have examined the stock future trends prediction accuracy of our approach with the other modern approaches using the same dataset. To show the performance comparison, we have performed the comparative analysis of our approach by comparing its average highest accuracy results with the average highest prediction results of the approaches mentioned in [47–50]. The obtained quantitative results are shown in Table 8.

Xu et al. [47] presented a DL-based approach namely the attention-based LSTM framework employing the financial articles to predict the stock future behavior. The work in [47] attained an average accuracy value of 54.58%. In contrast, the method in [48] introduced Kalman filter-based Accelerated Gradient LSTM to determine the future move-

ments of the stock market and showed an average accuracy of 90.42%. Similarly, in [50] the author proposed the LSTM-based model for stock market prediction and obtained an average accuracy of 66.83%. Moreover, Sadorsky et al. [49] proposed an ML-based approach namely Random Forest (RF) to predict the stock prices based on the analysis of the financial news articles with an average accuracy of 90%. Whereas, it can be seen from Table 8, that our method attained an accuracy value of 92.5% which is higher than all techniques under comparison. More specifically, the methods in [47–50] show an average accuracy value of 75.4%, which is 92.5% in our case, therefore, the presented framework exhibits an average performance gain of 17.1% approx.

For the reported quantitative results, it can be said that our method is more robust to stock market movements prediction than the other approaches, due to the employment of DCWR which results in the more discriminative set of feature selection. Furthermore, the techniques in [47–50] are economically inefficient and suffering from the model over-fitting problem. Whereas our method applies HANet which is capable of dealing with over-fitted training data and assists in attaining effective prediction accuracy. Therefore, it can be said that our approach is more proficient than the peer methods.

Conclusion

Prediction of stock market prices is an important and challenging task concerning both academic and financial research. Recent advancements in machine learning, especially deep learning has made it possible for the researchers to devise automated and intelligent methods to predict the stock prices depending on the indicators, financial news, or social media posts. This work concentrates on the prediction of stock market prices using financial news and validating the predictions using factual data such as stock market opening and closing prices. By nature, stock markets are volatile to short-term events which further increases the complexity of this task. The proposed method attempts to overcome this issue by introducing a deep learning-based technique that uses the financial news articles and tries to predict the stock market prices using the information embedded there. We first applied some renown preprocessing techniques to cleanse our data and make them more friendly to machine learning algorithms. After that, we performed feature extraction by applying the DCWR approach. For feature reduction, we incorporated ICA method, and finally, resultant features are fed to HANet classifier for prediction of stock market. After evaluation of the proposed method over the standard dataset, we can say that our method performs well as compared to the state-of-the-art techniques. In future, we aim to

investigate other DL-based methods for stock market future trends prediction.

Funding Funding was provided by Qassim University, represented by the Deanship of Scientific Research under the number (10302-coc-2020-1-3-I), 1441 AH/2020 AD.

Declarations

Conflict of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Fama EF (1965) The behavior of stock-market prices. *J Bus* 38(1):34–105
2. Chen W, Zhang H, Mehlaawat MK, Jia L (2021) Mean–variance portfolio optimization using machine learning-based stock price prediction. *Appl Soft Comput* 100:106943
3. Barberis N, Thaler R (2005) A survey of behavioral finance. Princeton University Press
4. Chen W, Jiang M, Zhang W-G, Chen Z (2021) A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Inf Sci* 556:67–94
5. Albahli S, Shiraz M, Ayub N (2020) Electricity price forecasting for cloud computing using an enhanced machine learning model. *IEEE Access* 8:200971–200981
6. Gite S, Khatavkar H, Kotecha K, Srivastava S, Maheshwari P, Pandey N (2021) Explainable stock prices prediction from financial news articles using sentiment analysis. *PeerJ Comput Sci* 7:e340
7. Mohan S, Mullapudi S, Sammeta S, Vijayvergia P, Anastasiu DC (2019) Stock price prediction using news sentiment analysis. In: 2019 IEEE fifth international conference on big data computing service and applications (BigDataService), pp 205–208
8. Ananthi M, Vijayakumar K (2021) Stock market analysis using candlestick regression and market trend prediction (CKRM). *J Ambient Intell Humaniz Comput* 12(5):4819–4826
9. Li Q, Tan J, Wang J, Chen H (2020) A multimodal event-driven lstm model for stock prediction using online news. *IEEE Trans Knowl Data Eng* 33(10):3323–3337
10. Tuarob S et al (2021) DAVIS: a unified solution for data collection, analyzation, and visualization in real-time stock market prediction. *Financ Innov* 7(1):1–32
11. Long J, Chen Z, He W, Wu T, Ren J (2020) An integrated framework of deep learning and knowledge graph for prediction of stock price trend: an application in Chinese stock exchange market. *Appl Soft Comput* 91:106205

12. Fung GPC, Yu JX, Lam W (2002) News sensitive stock trend prediction. In: Pacific-Asia conference on knowledge discovery and data mining, pp 481–493
13. Fung GPC, Yu JX, Lu H (2005) The predicting power of textual information on financial markets. *IEEE Intell Inform Bull* 5(1):1–10
14. Wu D, Fung GPC, Yu JX, Liu Z (2008) Integrating multiple data sources for stock prediction. In: International conference on web information systems engineering, pp 77–89
15. Wu D, Fung GPC, Yu JX, Pan Q (2009) Stock prediction: an event-driven approach based on bursty keywords. *Front Comput Sci China* 3(2):145–157
16. Rezaei H, Faaljou H, Mansourfar G (2021) Stock price prediction using deep learning and frequency decomposition. *Expert Syst Appl* 169:114332
17. Ding G, Qin L (2020) Study on the prediction of stock price based on the associated network model of LSTM. *Int J Mach Learn Cybern* 11(6):1307–1317
18. Li X, Wu P, Wang W (2020) Incorporating stock prices and news sentiments for stock market prediction: a case of Hong Kong. *Inf Process Manag* 57(5):102212
19. Vanipriya CH, Tomar A, Gupta G, Gandotra N, Sheshappa SN, Thammireddy K (2021) Stock market prediction using sequential events. In: 2021 8th International conference on computing for sustainable global development (INDIACom), pp 177–183
20. Seong N, Nam K (2021) Predicting stock movements based on financial news with segmentation. *Expert Syst Appl* 164:113988
21. Hao P-Y, Kung C-F, Chang C-Y, Ou J-B (2021) Predicting stock price trends based on financial news articles and using a novel twin support vector machine with fuzzy hyperplane. *Appl Soft Comput* 98:106806
22. Schumaker RP, Chen H (2009) Textual analysis of stock market prediction using breaking financial news: the AZFin text system. *ACM Trans Inf Syst* 27(2):1–19
23. Kaya MIY, Karsligil ME (2010) Stock price prediction using financial news articles. In: 2010 2nd IEEE international conference on information and financial engineering, pp 478–482
24. Vargas MR, De Lima BSLP, Evsukoff AG (2017) Deep learning for stock market prediction from financial news articles. In: 2017 IEEE international conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA), pp 60–65
25. Lee SY, Oh SY, Lee S, Gim GY (2021) The firm life cycle forecasting model using machine learning based on news articles. *Int J Netw Distrib Comput* 9(1):1–9
26. Hu Z, Zhao Y, Khushi M (2021) A survey of forex and stock price prediction using deep learning. *Appl Syst Innov* 4(1):9
27. Dang M, Duong D (2016) Improvement methods for stock market prediction using financial news articles. In: 2016 3rd National foundation for science and technology development conference on information and computer science (NICS), pp 125–129
28. Nabipour M, Nayyeri P, Jabani H, Mosavi A, Salwana E (2020) Deep learning for stock market prediction. *Entropy* 22(8):840
29. Schumaker R, Chen H (2006) Textual analysis of stock market prediction using financial news articles. In: AMCIS 2006 proceedings, p 185
30. Li X, Wu P (2021) Stock price prediction incorporating market style clustering. *Cognit Comput* 24(4):1–18
31. Duarte JJ, González SM, Cruz JC (2021) Predicting stock price falls using news data: evidence from the Brazilian market. *Comput Econ* 57(1):311–340
32. Kowsari K, Meimandi KJ, Heidarysafa M, Mendu S, Barnes L, Brown D (2019) Text classification algorithms: a survey. *Information* 10(4):150
33. Sun C et al (2019) A deep learning approach with deep contextualized word representations for chemical–protein interaction extraction from biomedical literature. *IEEE Access* 7:151034–151046
34. Peters ME et al (2018) Deep contextualized word representations. *arXiv preprint*. <http://arxiv.org/abs/1802.05365>
35. Šimon P, Hong J-F (2007) Word sense induction using independent component analysis. In: ROCLING 2007 poster papers, pp 423–433
36. Shabat HA, Abbas NA (2020) Independent component analysis based on natural gradient algorithm for text mining. In: 2020 1st Information technology to enhance e-learning and other application (IT-ELA), pp 72–76
37. Jutten C, Herault J (1991) Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture. *Signal Process* 24(1):1–10
38. Rutledge DN (2018) Comparison of principal components analysis, independent components analysis and common components analysis. *J Anal Test* 2(3):235–248
39. Wang W et al (2020) Reinforcement-learning-guided source code summarization via hierarchical attention. *IEEE Trans Softw Eng* 48(1):102–119
40. Kim D-K, Lee B, Kim D, Jeong H (2020) Multi-label classification of historical documents by using hierarchical attention networks. *J Korean Phys Soc* 76(5):368–377
41. Ding X, Zhang Y, Liu T, Duan J (2014) Using structured events to predict stock price movement: an empirical investigation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1415–1425
42. Rish I (2001) An empirical study of the naive Bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence, vol 3, no. 22, pp 41–46
43. Pal M (2005) Random forest classifier for remote sensing classification. *Int J Remote Sens* 26(1):217–222
44. Wright RE (1995) Logistic regression. In: Grimm LG, Yarnold PR (eds) Reading and understanding multivariate statistics. American Psychological Association, Washington, DC, pp 217–244
45. Friedman JH (2002) Stochastic gradient boosting. *Comput Stat Data Anal* 38(4):367–378
46. Jaggi M, Mandal P, Narang S, Naseem U, Khushi M (2021) Text mining of stocktwits data for predicting stock prices. *Appl Syst Innov* 4(1):13
47. Xu Y, Keselj V (2019) Stock prediction using deep learning and sentiment analysis. In: 2019 IEEE international conference on big data (big data), pp 5573–5580
48. Deepika N, Bhat MN (2021) An efficient stock market prediction method based on Kalman Filter. *J Inst Eng (India) Ser B* 102(4):629–644
49. Sadorsky P (2021) A random forests approach to predicting clean energy stock prices. *J Risk Financ Manag* 14(2):48
50. Vignesh CK (2020) Applying machine learning models in stock market prediction. In: EPRA international journal of research and development (IJRD), pp 395–398

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.