International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-12 Issue-5, June 2023

An Aggregator Framework for Transforming Big Data in Real-Time using PT-INDRNN

Sowmya R, Suneetha K R

Abstract*: The prediction of stock market prices based on the financial text sentiment classification using Machine Learning (ML) and Deep Learning (DL) models is becoming popular among researchers in the era of Big Data (BD). Nevertheless, owing to the lack of extensive analysis, most of the developed ML and DL models failed to achieve better classification results. Thus, for the real-time prediction of the stock price polarity, a Probability Tanh-Independently Recurrent Neural Network (PT-IndRNN)-based classification of the sentiment in financial text data from Twitter is proposed to address this problem. Primarily, by utilising the corresponding API, real-time financial data and Twitter data are extracted and stored in a MongoDB database using Apache Flume. This stored data, along with the historical big datasets, is taken and pre-processed. Next, by deploying the proposed Hadoop Distributed File System (HDFS) clustering, the pre-processed stock market data and Twitter data are combined in real-time, as well as the historical dataset, separately. After that, the features are extracted from the clustered sentences. Then, by utilizing the SentiWordNet, the sentences chosen using Linear Scaling-Dwarf Mongoose Optimization Algorithm (LS-DMOA) are converted to negative and positive scores. In the end, the sentiment of the financial texts is classified by the PTh-IndRNN, which is proven by obtaining reliable result values.

Keywords*: Hadoop Distributed File System (HDFS), Probability Tanh-Independently RNN (PTh-IndRNN), Application Programming Interface (API), Linear Scaling-Dwarf Mongoose Optimization Algorithm (LS-DMOA), financial text sentiment analysis.

I. INTRODUCTION

The act of forecasting the future value of a company's stock or other financial instruments traded on an exchange is referred to as Stock Market Prediction (SMP). If the SMP is performed effectively, it could be paramount for the investors. Investors might be guided in making suitable decisions on whether to buy or sell shares by the effectual SMP (Nousi& Tjortjis, 2021, [1]). According to the Efficient Market Hypothesis (EMH), it is not possible to predict the market with 100% accuracy. However, the researchers have not stopped their efforts to enhance methodologies for surpassing the stock price random walk pattern (R. Gupta & Chen, 2020, [2]). For ages, an active area of research has been designing robust and accurate predictive models for stock price prediction (Mehtab& Sen, 2020, [3]). ML and text data mining implemented on data extracted from social networks are famous techniques to detect the future value of the stock market.

Helpful information, such as public government policies, decisions, political events, and investors' opinions, is often found in the web content of social networks, including Facebook, Twitter, Instagram, and online reviews. (Kesavan et al., 2020, [4]). The performance of forecasting stock prices could be effectively improved by considering fundamental analysis data, such as financial news on the web or posts on social networks. (Lin et al., 2022,[5]). Social networks are characterized by the 5Vs of BD like velocity, volume, value, variety, and veracity. Thus, BD analytic techniques and frameworks in Social Networks are commonly exploited (Bazzaz Abkenar et al., 2021, [6]).

Sentiment analysis grounded on financial texts, which has a significant effect on perceiving the emotions of the crowd and predicting the financial or other economic situation trends, is termed the usual BD analytic technique (Shao & Chen, 2022, [7]). When it comes to the interpretation and analysis of sentences, words, phrases, and statements, data mining requires all kinds of text-mining tools, as the information is represented in textual form (Hassani et al., 2020, [8]). In text mining of extensive datasets, Apache Spark and Hadoop MapReduce have played a vital role in business analytics. Spark applications could be written rapidly, and Hadoop has flexible scalability (Y. K. Gupta & Sharma, 2020, [9]). Next, the financial text sentiment can be classified as positive, negative, or neutral with the features of the extracted text. A few methodologies like Support Vector Machine (SVM), Generative Adversarial Network (GAN), and Domain Adaptation have been developed for the classification of financial text sentiment (S. Dong & Liu, 2021, [10]). Nevertheless, due to the unstructured nature of the data, these techniques have some limitations. Thus, this paper proposes a reliable DL-based PTh-IndRNN for financial text sentiment classification, enabling real-time BD analysis on Twitter.

A. Problem Statement

There are certain limitations due to the following problems, despite the development of several ML models for predicting market trends based on sentiment analysis.

Manuscript received on 02 May 2023 | Revised Manuscript received on 09 May 2023 | Manuscript Accepted on 15 June 2023 | Manuscript published on 30 June 2023.

*Correspondence Author(s)
Sowmya R*, Research Scholar, Department of Computer & Engineering, Bangalore Institute of Technology, Bengaluru, Visvesvaraya Technological University, Belagavi (Karnataka), India. Email: sowpu29@gmail.com
ORCID ID: https://orcid.org/0000-0002-4527-8060
Dr. Suneetha K R, Professor Department of Computer & Engineering, Bangalore Institute of Technology, Bengaluru, Visvesvaraya Technological University, Belagavi (Karnataka), India. Email: suneetha.bit@gmail.com

- The efficient representation, access, and analysis of unstructured or semi-structured data are still challenging, considering the variety of datasets in BD.
- A dataset with high dimensionality may not be minable, causing ML algorithms to overfit data or generate incomprehensible rules.
- To perform mapping, shuffling, and reducing tasks, MapReduce in HDFS requires a lot of time, thereby increasing latency.

By analyzing these problems, the contribution of the proposed framework is given as,

- A novel HDFS MapReduce model is proposed for solving unstructured data problems and time complexity problems in existing HDFS.
- Feature Selection (FS) based on the LS-DMOA algorithm is proposed to solve the high dimensionality in the datasets.

The remaining part is arranged as follows: the related works of the proposed framework are elucidated in section 2; a detailed explanation of the approaches leveraged in the proposed framework is described in section 3; the experimental results of the proposed framework are expounded in section 4; the paper is winded up with the future work in section 5.

II. RELATED WORKS

(Xiaofeng et al., 2021, [11]) Done research on sentiment classification of future predictive texts. Here, grounded in Bi-directional Encoder Representations from Transformers (BERT), sentiment classification was performed. According to the results, the accuracy of the sentiment measurement was enhanced. Nevertheless, since the stock's closing price was neglected, the prediction was not reliable.

(Jaggi et al., 2021, [12]) Constructed a system centred on historical stock prices and financial-related messages for detecting stock movement. The system was presented with Fin-A Lite BERT (FinALBERT). For training the FinALBERT, the dataset was labelled with three methodologies. The percentage change methodology with '2' labels depicted the superior outcomes among the labelling methods. Nevertheless, a significant amount of time was spent on training FinALBERT; moreover, the system's performance showed no considerable enhancement.

(Achyutha et al., 2022, [13]) Exhibited a user classification and stock market-centric recommendation engine. For the sentimental analysis of tweets, the engine was developed with Naïve Bayes (NB). The final prediction recommended which company was nice to invest in. But the recommendation outcomes might deteriorate with unstructured data.

(Rodrigues & Chiplunkar, 2022, [14]) Examined a BD system for topic classification and sentiment analysis of Twitter data. Next, a sentiment analysis model with the Hybrid Lexicon-NB Classifier (HL-NBC) was presented. The sentimental analysis was done by the HL-NBC technique with superior accuracy. However, performance deterioration could be caused by unstructured data constructed by Hive on Hadoop.

(Demirbaga, 2021, [15]) Developed HTwitt for the analysis of the streamed Twitter data. On top of the Hadoop ecosystem and a set of ML methodologies embedded within a BD analytics platform, the HTwitt was constructed. According to the results, high-quality outcomes were achieved. Nevertheless, the sentiment of text in continuous values could not be detected accurately with the NB.

(J. Dong, 2020, [16]) Recommended a financial investor sentiment analysis grounded on Field Programmable Gate Array. A multi-window Convolutional Neural Network (CNN)-LSTM and a multi-window CNN were combined. According to the results, the CNNs' accuracy was better. Still, the accuracy of CNN could not be reliable without the regularization of missing values.

(Bourezk et al., 2020, [17]) Employed '2' techniques of sentiment classification centred on online stock market news articles for detecting the movements of stock prices. Lexicon-centric and NB classifier-centric sentiment analysis were the two deployed techniques. According to the results, a negative mood had a significant impact on fluctuations in stock market indices. However, only the stock prices of some companies were evaluated, which made the system prone to overfitting problems.

(Dubey et al., 2021, [18]) Developed an Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO)-centric system for data pre-processing and data classification in the BD environment. According to the results, the PSO-SAW demonstrated efficacy in terms of accuracy level. However, an update procedure was required as the content information of attributes varied; thus, the system was limited.

(Khan et al., 2022, [19]) Built an SMP grounded on ML classifiers and social media news. For the stock market prediction, twelve ML classifiers were evaluated. In the performance of the individual classifiers, enhancements were depicted by the ensemble of forecasts. However, it was challenging to account for the time overhead in the prediction model due to the multiple analyses of the ML classifiers.

(Deepika & Nirupama Bhat, 2021, [20]) Presented an efficient SMP centred on the Kalman filter. The Kalman filter was used to predict the stock market value and smooth out noise, while the Accelerated Gradient LSTM (AG-LSTM) was employed. The AGLSTM attained 90.42% accuracy with the Kalman filter. The model was unable to improve prediction accuracy with outliers or missing data, even when using the Kalman filter.

(Aasi et al., 2021, [21]) Designed a sentiment and public engagement analysis for stock price prediction. For offering a 1-week prediction on the stock close value for the technology company "Apple Inc", a Multivariate Multistep output LSTM (MMLSTM) was presented. The MMLSTM surpassed the LSTM models. However, the system was designed to detect only the stock value of a single company.

(Nti et al., 2021, [22]) Recommended a multi-source information-fusion system for accuracy enhancement in stock market prediction. The fusion was presented with CNN and LSTM. According to the results, enhanced prediction accuracy was achieved on the collected dataset. Still, for the fusion model, it was challenging to afford the essential memory footprint.

(Yasir et al., 2020, [23]) Developed a DL-centric system for detecting interest rates using Twitter sentiment. Twitter sentiments of the '6' mega-events were deployed as input. According to the results, when event sentiment was integrated, the system's error was minimised. However, the utilised DL model takes more time to train its parameters due to the large dimensionality of the features.

(Zhao et al., 2021, [24]) Presented a sentimental analysis and key entity detection approach for online financial texts developed centred on the BERT. To enhance the performance of the given system, ensemble learning was leveraged. As per the experiments, the BERT-centric systems' performance was superior. However, the system's performance may deviate if no significance is given to neutral, positive, and sarcastic sentences.

(Xu et al., 2020, [25]) Employed a Stock Movement Prediction Network (SMPN) grounded on the tweet and historical prices. An incorporative attention system that merged the local and contextual attention schemes for offering information was employed by the network. According to the results, the system demonstrated superior performance on both the tweet and the historical price datasets. Nevertheless, the accuracy level might be degraded since the system's low learning efficiency was utilised for extracting contextual information.

III. PROPOSED FINANCIAL TEXT SENTIMENT CLASSIFICATION METHODOLOGIES

The fluctuations in the stock market are influenced since the financial texts are more readily available as BD. Thus, the investor is aided by the prediction of the economic text polarity to invest money in the share market. Thus, for classifying the polarity of Twitter texts to predict stock prices, a DL-based PTh-IndRNN with LS-DMOA is proposed to overcome this limitation. Figure 1 illustrates the proposed framework.

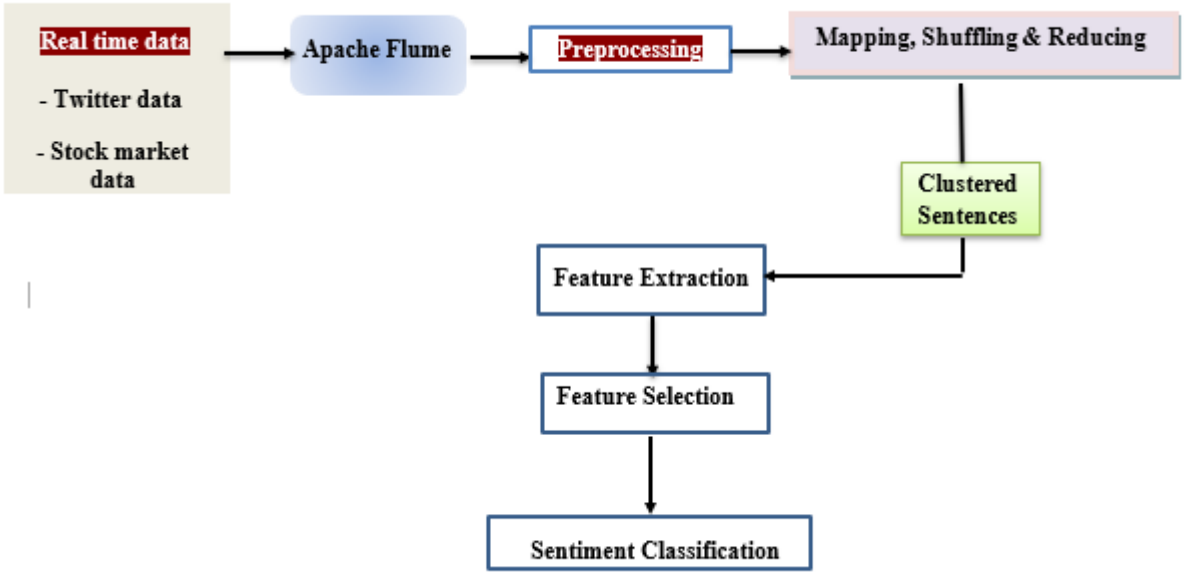


Figure 1: Architecture of the proposed framework

A. Iinput Data

Primarily, the proposed model utilises real-time and historical big data from Twitter and the stock market.

Real-time data: Real-time Twitter data, including financial tweets, is extracted using the Twitter API. By employing the External API, the real-time stock market data is extracted from external websites.

Storing Real-time Data: In MongoDB, the extracted real-time data is stored as a database using Apache Flume. Apache Flume is an open-source, robust, reliable, and flexible system used to collect, aggregate, and move large amounts of unstructured data from multiple data sources into MongoDB, a NoSQL database. The stored database (D) is denoted as,

$$D = [d_T, d_S] \quad (1)$$

Here, the Twitter and the stock market datasets are signified as d_T, d_S.

Datasets: For enhancing the system's accuracy, the publicly available datasets of Twitter tweets and the nifty-50-stock-market datasets are taken with D. The historical dataset set (H) is equated as,

$$H = [h_T, h_S] \quad (2)$$

Here, the historical Twitter and stock market dataset is depicted as h_T, h_S.

B. Pre-processing

To facilitate a more effective classification process, pre-processing is performed to restructure the data. The Twitter and the stock-market datasets are pre-processed separately in the proposed model.

1) Twitter Data Set

The financial texts in d_T, h_T Are pre-processed separately with the following steps:

Tokenisation: During tokenisation, financial texts are split into individual words.

URL removal: Here, the URL present in the financial texts is removed.

Stop word removal: In this step, the unnecessary words (is, was) are removed from the text.

Word Stripping: The elongated words (ngggg, …..) and Short words (gd,gn) are reframed.

Stemming: Next, the words are reduced by neglecting prefixes, suffixes, and affixes, such as "ed," "ing," and "en," etc.

The pre-processed d_T, h_T is denoted as a_T, b_T.

2) Stock-Market Data Set

The stock-market datasets d_S, h_S are pre-processed with the following steps,

Attribute extraction: The attributes like date, symbol, series, previous close, open, high, low, last, close, VWAP, volume, turnover, trades, deliverable volume, and % deliverable are extracted individually with their equivalent data from the d_S, h_S.

Numeralization: The strings in d_S, h_S They are converted into numbers as the system can only process numerical data.

Missing value imputation:

Here, it will be interpolated with the average value of the previous and next time-series values of the corresponding missing data by the interpolation technique if a missing value is detected in d_S, h_S. The pre-processed d_S, h_S are denoted as z_S, y_S.

3) Clustering

Next, based on the company name, the pre-processed data a_T, z_S The Dataset is clustered. To obtain effective sentence clusters, the mapping, shuffling, and reducing processes of existing HDFS and Apache Spark are modified.

a) Mapping.

Here, the words are stored in distributed files, where words are mapped based on keywords by deploying the proposed Hamming-Correlated K-Means (H-CKMeans) algorithm. As K-means can group unlabeled dataset instances into clusters based on similar attributes, it is selected. However, it does not guarantee that it will find the global optimum solution for optimal clustering. Thus, the Pearson Correlation Coefficient (PCC) and Hamming distance are utilised in the existing K-Means algorithm to address these issues. In the H-CKMeans algorithm, the c centroids for c Clusters are estimated using the PCC technique as:

$$C_n = \frac{\sum(\alpha_i - \bar{a}_T)(\beta_i - \bar{z}_T)}{\sqrt{\sum(\alpha_i - \bar{a}_T)(\beta_i - \bar{z}_T)^2}} \quad (3)$$

The n^{th} The cluster centroid is depicted as C_n, $n = 1,2,...,c$, $\alpha_i, \beta_i \in \omega_i$ denotes the i^{th} data in a_T, z_S and $\bar{a}_T, \bar{z}_T$ signifies the mean data values of a_T, z_S. The distance (similarity) between C_n and data points of a_T, z_S are estimated using the Hamming distance (d) after the c Centroids are estimated.

$$d_i = \sum_{i=1}^{m} |C_n - \omega_i| \quad (4)$$

The data point in a_T *or* z_S is mentioned as $\omega_i \in a_T, z_S$. The data points are mapped in the same cluster (file) for a smaller d_i. The mapped c files (k_c) are denoted as,

$$M = \{k_1, k_2, ..., k_c\} or\, k_n \quad (5)$$

The mapped file set is depicted as M.

b) Shuffling

Then, the words in k_n They are shuffled, where the same word is stored in the same file. The Fisher-Yates Shuffling Algorithm (FYSA) is used for shuffling.

Step 1: Generate a temporary array $(Tmp[\ ])$ And store the words of M in $Tmp[\ ]$ And give the key value of $(1,2,.....,nn)$ To each word. The exact words have the same value.

Step 2: Choose the random value v from $(1,2,.....,nn)$ And store the corresponding word from $Tmp[\ ]$ In a new array $(N[\ ])$ And delete the word from the $Tmp[\ ]$.

Step 3: Until all the words in $Tmp[\ ]$ It is deleted. Repeat step 2. The words with the same key value are stored in the same array by doing so, and the shuffled set is depicted as (SS),

$$SS = \lfloor s_1, s_2, .., s_Q \rfloor or\, s_z, z = 1,2,.....,Q \quad (6)$$

Where, the Q^{th} Shuffled file, which contains the same keyword, is signified as s_Q.

c) Reducing

Significant keywords are selected in the reducer using the Tournament Selection-based Sand Cat Optimization Algorithm (TS-SCOA) from the shuffled data. As the Sand Cat Optimisation Algorithm (SCOA) has high convergence accuracy and the ability to escape from locally optimal solutions, it is chosen for the proposed model. However, the SCOA has a drawback of random wheel selection. Thus, to develop the

proposed TS-SCOA algorithm, the Tournament selection approach is leveraged in the SCOA algorithm to overcome that drawback.

Initialization: The initial position of Sand Cats (SCs), which is the shuffled keywords, is initialized in the TS-SCOA and is represented within the dimension $\Re$ as,

$$P=\begin{bmatrix}\rho_{1,1} & \rho_{1,2} & \cdots & \rho_{1,\Re}\\ \rho_{2,1} & \rho_{2,2} & \cdots & \rho_{2,\Re}\\ \vdots & \vdots & \ddots & \vdots\\ \rho_{Q,1} & \rho_{Q,2} & \cdots & \rho_{Q,\Re}\end{bmatrix} or \left[\rho_{Z,Y}\right], Y=1,2,...,\Re \quad (7)$$

Where is the position of the SC Z in $\Re$ is depicted as $\rho_{Z,Y}$.

Exploration Phase (prey search): The prey equation of SC is given as,

$$\vec{\rho}_Z(I+1)=\vec{x}.\left(\vec{\rho}^*(I)-R.\vec{\rho}_z(I)\right) \quad (8)$$

Here, the position of the SC Z In the iteration I is signified as $\vec{\rho}_Z(I)$ The random number within the range of $(0,1)$ is depicted as R, and the best position of SC, which is obtained by estimating the fitness of the position, is mentioned as $\vec{\rho}^*(I)$. The fitness is equated as,

$$f=\uparrow acc(cl) \quad (9)$$

Where fitness is considered as the enhanced clustering accuracy. The range of sensitivity of SCs is depicted as $\vec{x}$, which is equated as,

$$\vec{x}=\vec{x}_g\times R \quad (10)$$

Here, the general sensitivity range of SC, which gets linearly decreased from 2 to 0, is signified as $\vec{x}_g$. This can be equated to,

$$\vec{x}_g=\delta_q-\frac{\delta_q\times I}{I_{mxm}} \quad (11)$$

Where, I_{mxm} implies the maximum iteration, and δ_q It is set to 2 as the SC can sense below 2 kHz.

Exploitation phase (attacking prey): The random distance (σ) The distance between the prey position and the SC is estimated using tournament selection to perform a prey attack.

$$\sigma=Q\times\left|R\times\rho^*(I)-\rho_Z(I)\right| \quad (12)$$

The sensitivity range of the SC is assumed to be a circle. For selecting a random angle (ϕ) The direction of movement of the SC employs the Roulette wheel selection algorithm. Likewise, the SC moves in a different circumferential direction in the search space and attacks the prey using the formula,

$$\rho_Z(I+1)=\rho^*(I)-\vec{x}.\vec{\rho}_{random}.\cos(\phi) \quad (13)$$

Here, the random position is generated between $\rho^*(I)$ and $\rho_Z(I)$ is depicted as $\vec{\rho}_{random}$.

Balancing exploration and exploitation: In the TS-SCOA, for balancing the exploration and exploitation phase, an adaptive factor (γ) is utilized. The (γ) is represented as,

$$\gamma=\vec{x}_g.(2\times R-1) \quad (14)$$

The SC updates the position using equation (8) if $|\gamma|\le 1$, else the position is updated using equation (13). By doing this, the optimal position of sc (ρ^*) (that is, optimal reduced keywords) can be obtained. The selected keywords (company names) are depicted as,

$$K=\{k_1,k_2,.....,k_\aleph\} or\ k_l, l=1,2,...,\aleph \quad (15)$$

Here, the reduced file set is depicted as K, and the $\aleph^{th}$ The reduced file is mentioned as $k_\aleph$.

The pseudocode for the proposed TS-SCOA

Input: $\left[s_1,s_2,..,s_Q\right] or\ s_Z$

Output: Selected keywords

Begin

Initialize SC population, $\Re,\gamma,\vec{x},\vec{x}_g$ and maximum iteration I_{mxm}

Set $I=1$

Calculate Fitness

While $(I\le I_{mxm})$

For each SC, **do**

Estimate distance using TS

Obtain angle $\phi\ (0^o\le\phi\le 360^o)$

If $(|\gamma|\le 1)\{$

Update position using $\rho^*(I)-\vec{x}.\vec{\rho}_{random}.\cos(\phi)$

$\}$ **Else if** $(|\gamma|>1)\{$

Perform $\vec{\rho}_Z(I+1)=\vec{x}.\left(\vec{\rho}^*(I)-R.\vec{\rho}_z(I)\right)$

$\}$

End If

End For

Update (ρ^*)

End While

Return the Best solution.

$I=I+1$

End

(iv) Final Output: The final cluster of Twitter data and stock market data under the same company name is obtained from the reducers, which is equated as,

$$E=\{e_1,e_2,....,e_\chi\} or\ e_r, r=1,2,...,\chi \quad (16)$$

Here, the reduced file set is depicted as E, and the cluster of the χ^{th} The Company is signified

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

as e_{χ}. Here, the unstructured data is converted to structured data.

4) Feature Extraction

Features such as n-grams (sequences of n words), bigrams (sequences of 2 words), emoticons (symbols representing emotions), Parts of Speech (PoS), and others are extracted from the clustered data. The extracted feature set (A) is equated as,

$$A=\{\varepsilon_1,\varepsilon_2,....,\varepsilon_t\} or\ \varepsilon_{\wp}, \wp=1,2,...,t \quad (17)$$

Where the extracted t^{th} The Feature is depicted as ε_t.

5) Feature Selection

Significant features are optimally selected from the extracted feature set (A) To reduce the training time of the classifier. LS-DMOA is proposed for FS. Due to the optimal FS in high-dimensional datasets, the Dwarf Mongoose Optimisation Algorithm (DMOA) is considered reliable. However, DMOA has the drawback of random position initialisation. Thus, Linear Scaling (LS) is introduced in the DMOA and is referred to as the LS-DMOA technique.

Initialization: The position of the Dwarf Monkey (DM) population is considered to be the $\{\varepsilon_1,\varepsilon_2,....,\varepsilon_t\}$ and $\{\varepsilon_1,\varepsilon_2,....,\varepsilon_t\}$ Within the problem dimension $(\Im)$ in the LS-DMOA and is equated as,

$$A=\begin{bmatrix} \varepsilon_{1,1}\ \varepsilon_{1,2}\cdots\varepsilon_{1,\Im-1}\ \varepsilon_{1,\Im} \\ \varepsilon_{2,1}\ \varepsilon_{2,2}\cdots\varepsilon_{2,\Im-1}\ \varepsilon_{2,\Im} \\ \vdots\ \vdots\ \ddots\ \vdots\ \vdots \\ \varepsilon_{t,1}\ \varepsilon_{t,2}\cdots\varepsilon_{t,\Im-1}\ \varepsilon_{t,\Im} \end{bmatrix} or\left[\varepsilon_{\wp,\eta}\right], \eta=1,2,..,\Im \quad (18)$$

Here, the DM population is depicted as A The size of the population is signified as t, and the position of DM (t) in dimension $\Im$ is mentioned as $\varepsilon_{t,\Im}$. The distribution of $\left[\varepsilon_{\wp,\eta}\right]$ is given using the LS technique as,

$$\varepsilon_{\wp,\eta}=\lambda\left(\frac{\Im-\Im_{lb}}{\Im_{ub}-\Im_{lb}}\right) \quad (19)$$

Here, a random number, which is randomly distributed, is depicted as λ.

Alpha group: The alpha female (μ) Is the controller of the family unit in the DM population, which can be determined using the fitness function $F(\ )$,

$$\mu=\frac{F(\wp)}{\sum_{\wp=1}^{t}F(\wp)} \quad (20)$$

The $F(\ )$ In the proposed LS-DMOA is equated as,

$$F=\max(acc) \quad (21)$$

Fitness is the maximum accuracy. In the μ Group, the number of scouting mongooses is obtained as $(t-\hbar)$, and $\hbar$ Depicts the number of babysitters.

Exploitation stage: To give a signal to the path family unit on the track, μ creates a peep sound (ℓ). The sleeping mount, which can be denoted using a randomly distributed number (ζ) It is determined based on the candidate food solution. $(\varepsilon_{\wp+1})$.

$$\varepsilon_{\wp+1}=\varepsilon_{\wp}+\zeta\times\ell \quad (22)$$

The fitness of $\varepsilon_{\wp+1}$ is evaluated and $\varepsilon_{\wp+1}$ The best fitness is selected as the sleeping mat. The sleeping Mount $(M_{\wp})$ is evaluated after each iteration as,

$$M_{\wp}=\frac{F(\wp+1)-F(\wp)}{\max(|F(\wp+1),F(\wp)|)} \quad (23)$$

The average value of $M_{\wp}$ (ξ) It is obtained after the sleeping mount is identified.

$$\xi_{\wp}=\frac{\sum_{\wp=1}^{t}M_{\wp}}{t} \quad (24)$$

Exploration (or) scouting: The baby exchange criteria $(ct\geq pr)$ are achieved after ξ It is estimated that scouting is performed using the DM Scouter group. The time counter and babysitter exchange parameters are depicted as ct, pr. The movement vector $(\vec{V})$ Of DM to the sleeping mount is given as,

$$\vec{V}=\sum_{\wp=1}^{t}\frac{\varepsilon_{\wp}.M_{\wp}}{\varepsilon_{\wp}} \quad (25)$$

Next, the scout mongoose's next position is given with (V) as,

$$\varepsilon_{\wp+1}=\begin{cases}\varepsilon_{\wp}-vm*\vartheta*\left[\varepsilon_{\wp}-\vec{V}\right] & if\ \xi_{\wp+1}>\xi_{\wp}\ (Exploration)\\ \varepsilon_{\wp}+vm*\vartheta*\left[\varepsilon_{\wp}-\vec{V}\right] & else\qquad\qquad (Exploitation)\end{cases} \quad (26)$$

Where a random number is depicted as ϑ, and the collective-volatile movement of the mongoose group that linearly decreases during iterations is signified as vm.

$$vm=\left[1-\frac{It}{It_{\max}}\right]^{\left(\frac{2.It}{It_{\max}}\right)} \quad (27)$$

Here, the current and maximum iteration is delineated as $It, It_{\max}$. The feature $\varepsilon_{\wp+1}$ is selected if $F(\varepsilon_{\wp+1})>F(\varepsilon_{\wp})$.

The Pseudocode of LS-DMOA:

Input: Extracted features
Output: Selected features

Begin
Initialize the population $\Im, \lambda, It_{\max}, \ell, pr$, and number of babysitters $(\hbar)$
Determine the distribution of DM using the LS
Set $It = 1$, population size $t = t - \hbar$
While $(It \le It_{\max})$ **do**
Calculate F
Set ct
Determine alpha (μ)
Generate candidate food position $\varepsilon_{\wp+1}$
Evaluate $F(\varepsilon_{\wp+1})$
Perform evaluation of $M_{\wp}$ and determine $\xi_{\wp}$
If $(\xi_{\wp+1} > \xi_{\wp})${
Update scouter position using $\varepsilon_{\wp} - vm * \vartheta * [\varepsilon_{\wp} - \vec{V}]$
} **Else** {
Update $(\varepsilon_{\wp+1})$ using $\varepsilon_{\wp} + vm * \vartheta * [\varepsilon_{\wp} - \vec{V}]$
}
End If
Update best $\varepsilon_{\wp+1}$
End While
Return selected features.
End

6) Word Embedding

The words within the feature $(\varepsilon_{\wp+1})$They are converted to numbers utilising the SentiWordNet dictionary after the features are selected. To extract the synset of a word (i.e meaning, definition, etc.,), $word.synset(\ )$Is used. Then, synset scoring (Sc) of the word $(\upsilon)$is performed using,

$$Sc = Ps - Ns \tag{28}$$

Where, Ps, Ns Implies positive and negative scores. Next, as per the usage ranks, the synsets are weighted with the final score of each word in $(\varepsilon_{\wp+1})$and is given as,

$$\Omega = \sum_{\upsilon=1}^{mm} Sc(\upsilon)/\upsilon \tag{29}$$

Here, the score predicted for a word is depicted as Ω, and the synset rank is signified as υ. The polarity will be considered as positive/negative scores if $\Omega > Th$. The desired threshold score value is depicted as Th. Hence, the SentiWordNet dictionary (Dy) With negative and positive scores corresponding to the text.

7) Sentiment Classification

It is given to the proposed PTh-IndRNN, which classifies the polarity of texts after the scores of texts are predicted; moreover, it is mainly used to evaluate the conversion process of unstructured data into structured data. Since Independently Recurrent Neural Network (IndRNN) prevents gradient explosion problems, it is selected. However, the back-propagation time is maximized by the IndRNN. Hence, Probability Tanh (PTh) activation is embedded in the IndRNN to solve this problem. In Figure 2, the PTh-IndRNN architecture is depicted.

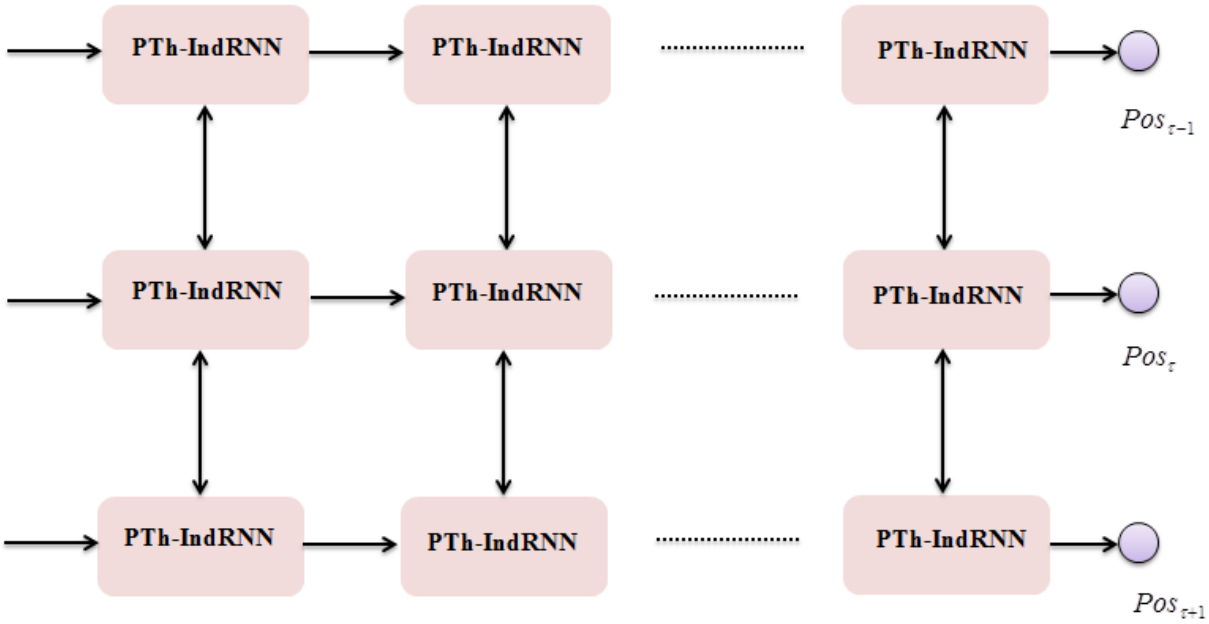


Figure 2: Proposed PTh-IndRNN architecture

Hidden states: The PTh-Ind RNN processes recurrent input weight (Γ) With the Hadamard product $(\circ)$ is equated as,

$$\varsigma_{\tau} = \kappa\left(o.Dy_{\tau} + \Gamma \circ \varsigma_{\tau-1} + B\right) \quad (29)$$

Where is the hidden state at the time step? (τ) is depicted as ς_{τ} The activation function is signified as κ, the weight matrix of Dy_{τ} is mentioned as o, and the bias value is delineated as B.

The hidden state of the neuron u (h_u, τ) is independent of the other hidden neurons in PTh-IndRNN and can be described as,

$$\varsigma_{u,\tau} = \kappa\left(o_u Dy_{\tau} + \Gamma_u \varsigma_{u.\tau-1} + B_u\right) \quad (30)$$

Here, the input weight, recurrent weight, and bias value for the neuron u are depicted as o_u, Γ_u, b_n.

Output: The output predicted polarity (Pol_{τ}) can be obtained using the PTh activation function from the hidden states as,

$$Pol_{\tau} = \left[\left(\left(\frac{\exp(\varsigma_{u,\tau}) - \exp(-\varsigma_{u,\tau})}{\exp(\varsigma_{u,\tau}) + \exp(-\varsigma_{u,\tau})}\right) + 1\right) \Big/ 2\right]^{v} \quad (31)$$

Where the constant value 3.0 is depicted as v. This is the result obtained for the a_T, z_S Dataset. Likewise, processes such as proposed HDFS, feature extraction, FS, and word embedding are performed with the b_T, y_s Datasets are given to the PTh-IndRNN for training and testing. Hence, in the result section, the obtained results from the pre-processed real-time datasets and historical datasets b_T, y_s Are evaluated. The stock market investor can decide whether to invest in the share market of the corresponding company or not based on the polarity predicted. Thus, the real-time BD analysis is performed.

IV. RESULTS AND DISCUSSIONS

The proposed system's performance is evaluated in comparison to existing works. The experiment was conducted on the Java development platform.

A. Data Set Description

Big-data datasets are created with real-time Twitter data and stock market data in the proposed model. Moreover, to demonstrate the proposed technique's performance, publicly available historical datasets are utilised.

Financial Tweets: This dataset includes over 28,000 publicly traded companies (with Company names and tickers) that are tagged with the company they are tweeted about, the company symbols, and other attributes.

NIFTY-50 Stock Market Data (2000 - 2021): It encompasses the stock price data of '50' stocks in the NIFTY-50 index from the National Stock Exchange (NSE) India. The datasets are at the day level, including pricing and trading values.

B. Performance Analysis.

Proposed HDFS, FS, and sentiment classification are the 3 phases in which the proposed framework's performance is evaluated.

1) Performance analysis of the proposed HDFS

Here, regarding fitness, the proposed reducer TS-SCOA's performance is compared to that of the prevailing SCOA, Artificial Algae Algorithm (AAA), BAT, and Crow Search Optimisation Algorithm (CSOA). Next, regarding clustering time, the proposed H-CKMeans clustering is analogized to the prevailing K-Means, Clustering Large Applications (CLARA), Partition Around Medoids (PAM), and Fuzzy C-Means (FCM).

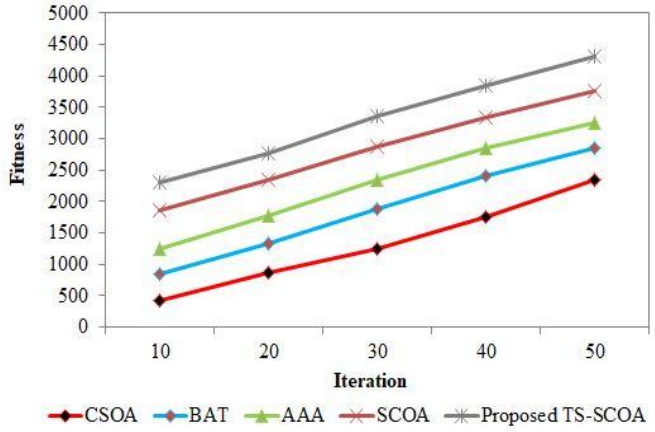


Figure 3: Fitness vs iteration

The graphical analysis of fitness values for proposed and existing algorithms, over 10 to 50 iterations, is depicted in Figure 3. The fitness is evaluated at the 40th iteration, in which the existing CSOA, BAT, and proposed algorithms select the optimal keywords with fitness values of 1757, 2415, and 3847, respectively. Thus, when analogized to the prevailing methodologies, the TS-SCOA achieved the best fitness, which improves the accuracy of the proposed HDFS clustering results.

Table 1: Clustering time analysis

Algorithms	Clustering time (ms)
FCM	77164
PAM	72659
CLARA	68482
K-Means	63548
H-CKMeans	58475

The proposed H-C-K Means algorithm achieves clustering in less time, at 5073 ms, compared to 18,689 ms when compared to the prevailing K-Means and FCM algorithms. Owing to the embedding of the Hamming distance and PCC function in the K-Means algorithm, superior outcomes are acquired.

2) Performance analysis of feature selection

Here, centred on the fitness value, the proposed FS is compared with the prevailing DMOA, Grasshopper Optimisation Algorithm (GOA), Harmony Search Optimisation Algorithm (HSOA), and Grey Wolf Optimisation Algorithm (GWOA).

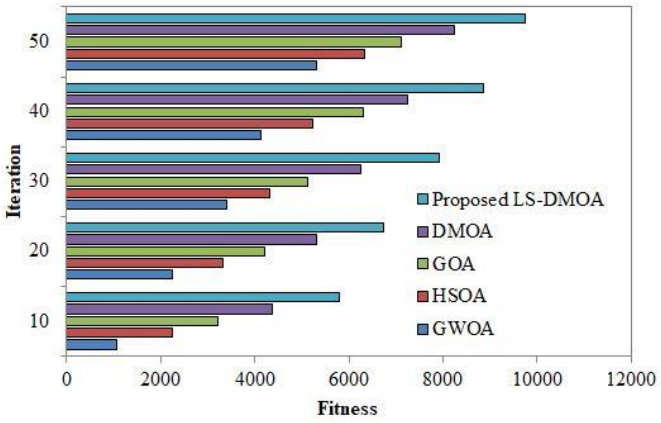
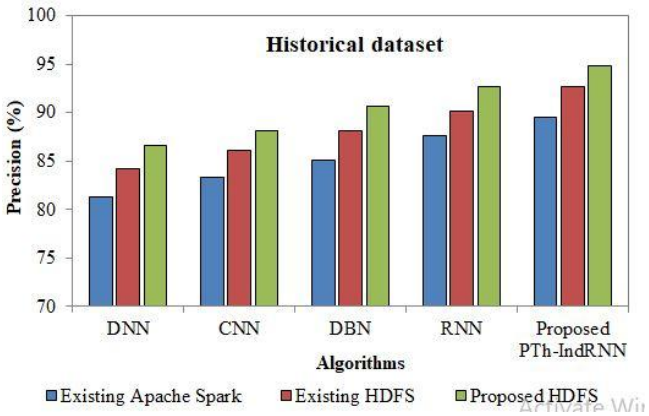


Figure 4: Fitness value analysis

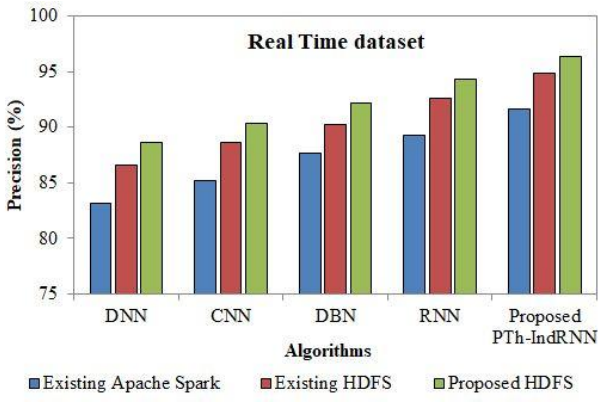
In Figure 4, the fitness obtained by the LS-DMOA and the existing algorithms for 10 to 50 iterations is pictorially depicted. At the 50th iteration, the proposed LS-DMOA selects the optimal feature with a fitness level of 9756, which surpasses that of the existing DMOA, GOA, HSOA, and GWOA. This enhancement is associated with the use of the LS technique in the DMOA approach.

3) Performance analysis of sentiment classification.

Grounded on the accuracy, training time, precision, recall, F-Measure, sensitivity, and specificity, the proposed sentiment classification PTh-IndRNN on real-time and historical datasets is evaluated with the prevailing HDFS, Apache Spark, and proposed HDFS techniques. The PTh-IndRNN is also compared with existing models, including the RNN, Deep Belief Network (DBN), CNN, and Deep Neural Network (DNN), to demonstrate its efficacy.



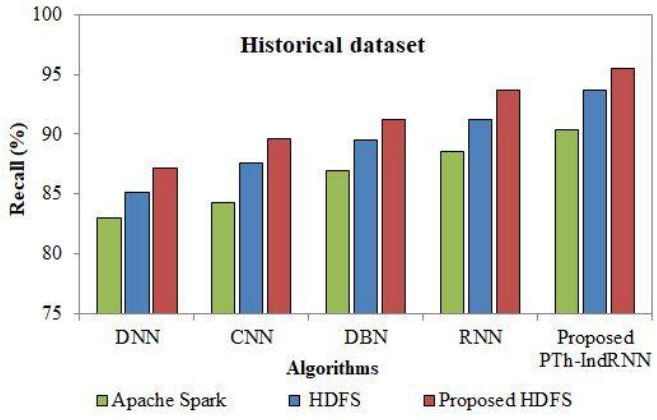
(a)



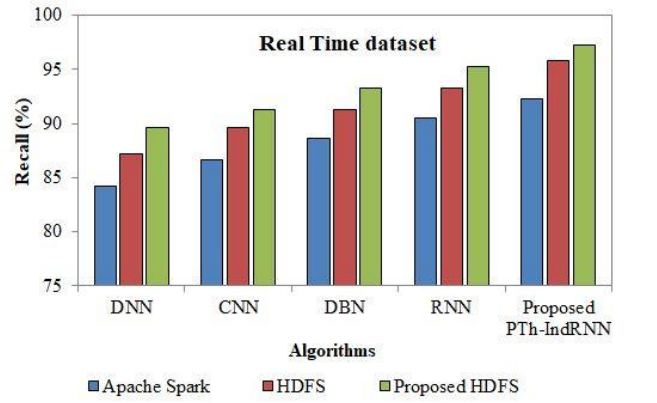
(b)

Figure 5: Precision evaluation on (a) historical dataset, (b) real-time dataset

The PTh-IndRNN achieved a precision level of 94.84% on the historical dataset using the proposed HDFS, which is 2.36%, 4.62%, and 9.49% higher than the prevailing RNN, DBN, and DNN, respectively. The PTh-IndRNN achieved a higher precision level with the proposed HDFS than with the existing HDFS and Apache Spark systems, as evident in Figures 5(a) and 5(b). The precision of PTh-IndRNN with the proposed HDFS is 92.32% on the real-time dataset, which is higher than that of the existing RNN, DBN, CNN, and DNN approaches. Thus, when compared to other classifiers, the proposed model makes fewer incorrect predictions with PTh-IndRNN.



(a)



(b)

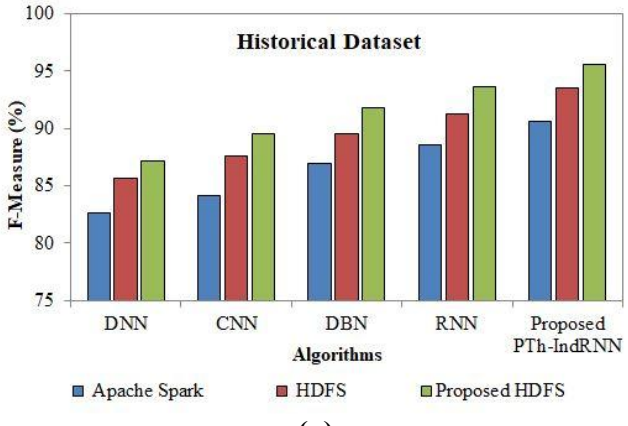
Figure 6: Recall results of classifiers

The recall levels obtained by the proposed PTh-IndRNN using traditional HDFS and Apache Spark are depicted in Figures 6(a) and 6(b). The proposed HDFS is higher than that of the existing classifier algorithms. Nevertheless, when classification is performed with the proposed HDFS system, the classifiers achieved better recall where the PTh-IndRNN perceived a higher level (95.48%, 97.23%) than the RNN (93.65%, 95.21%), DBN (91.18%, 93.21%) and CNN (87.12%, 91.32%) on both datasets. Thus, the proposed classification is suitable for sentiment classification in predicting stock values.

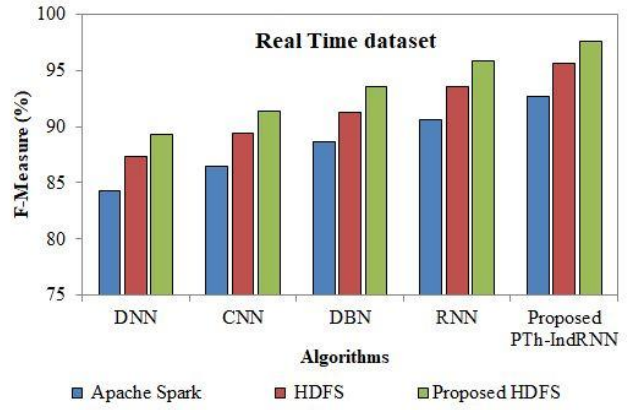
Table 2: Accuracy level (%) on the historical and real-time datasets

Algorithms	Historical dataset			Real-time dataset		
	HDFS	Apache Spark	Proposed HDFS	HDFS	Apache Spark	Proposed HDFS
DNN	85.6545	82.6298	87.2145	87.3247	84.2478	89.3584
CNN	87.6248	84.1254	89.5448	89.4578	865178	91.3658
DBN	89.5412	86.9575	91.8475	91.3256	88.6545	93.5471
RNN	91.2365	88.5148	93.6517	93.5148	90.6545	95.8475
Proposed PTh-IndRNN	93.5478	90.6548	95.6287	95.6589	92.6589	97.5478

The level of PTh-IndRNN is higher than that of existing algorithms, as shown in Table 2. When compared to HDFS and Apache Spark, the accuracy of 2.32% and 6.06% is improved with the proposed HDFS on the real-time dataset. Thus, the sentiment of the texts is classified more accurately with the PTh-IndRNN than with traditional classifiers.



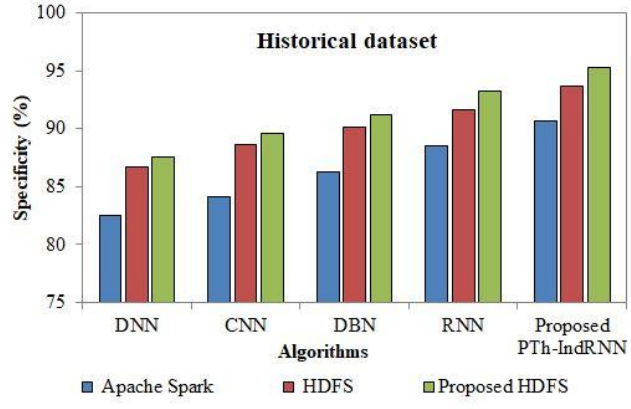
(a)



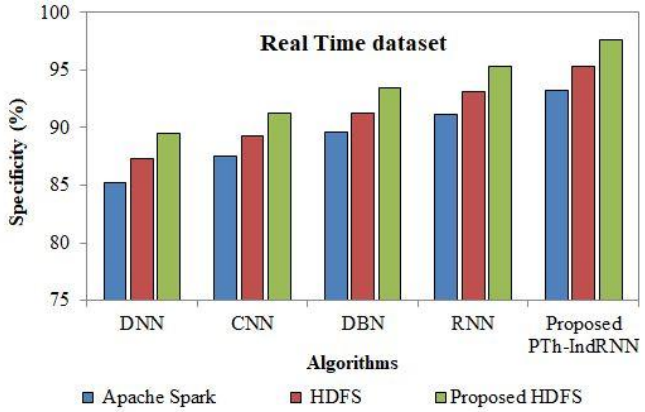
(b)

Figure 7: F-Measure values on (a) Historical dataset, (b) real-time dataset

Regarding precision and recall values, the f-measure is calculated. When compared to other classifiers, the F-Measure of PTh-IndRNN is superior, as shown in Figures 7(b) and 7(a). Even with the proposed HDFS, the DNN attained unreliable results (87.21%, 89.35%) on both datasets. However, the proposed model achieved a higher F-measure level, indicating the efficacy of the PTh-IndRNN in both datasets for text sentiment classification.



(a)



(b)

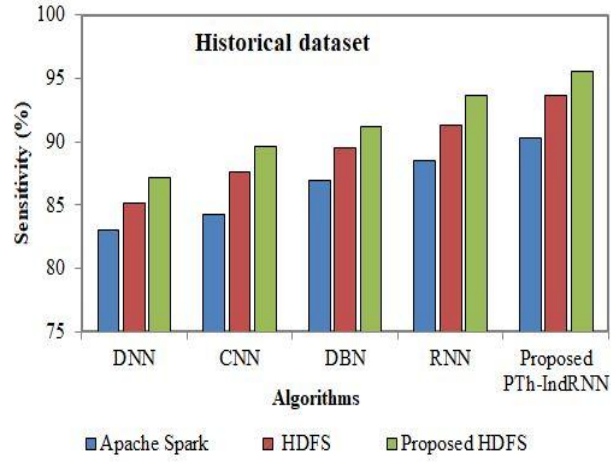
Figure 8: Specificity levels on (a) historical dataset, (b) Real-time dataset

In Figures 8(a) and 8(b), the specificity levels of the proposed and traditional classifiers, using HDFS, Apache Spark, and the proposed HDFS on historical and real-time data, are pictorially represented. More specificity levels (91.18%, 93.15%, and 95.32%) could be achieved by the RNN with the involvement of the proposed HDFS on the real-time datasets. Nevertheless, the PTh-IndRNN achieved a higher specificity level of 93.26%, 95.32%, and 97.65%, which is a result of incorporating PTh activation into the IndRNN.

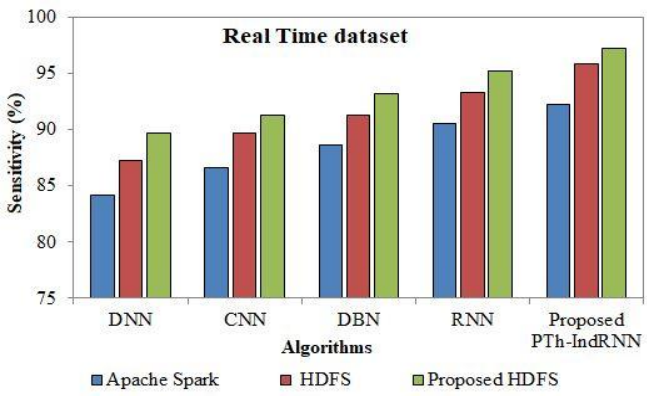
Table 3: Training time

Algorithms	Training time (ms)	
	Historical dataset	Real-Time dataset
DNN	119542	116521
CNN	114478	111457
DBN	109635	106356
RNN	104547	101478
Proposed PTh-IndRNN	99457	95658

The training time of DNN is higher (119542 ms and 116521 ms) on both datasets, followed by CNN and DBN, as evident in Table 3. However, when analogized to the prevailing systems, the RNN takes less time. However, the PTh-IndRNN exhibits a higher training time of 5090ms and 5820ms for the RNN, which demonstrates the PTh-IndRNN 's time efficiency.



(a)



(b)

Figure 9: Sensitivity analysis on (a) historical dataset, (b) real-time dataset

More sensitivity level is attained by the proposed and existing classifiers with the proposed HDFS on the historical datasets (95.48%, 91.18%, and 87.12%) and on the real-time datasets (97.23%, 93.21%, and 89.65%), which is depicted in Figure 9(a) and 9(b). The PTh-IndRNN performs better with the existing HDFS and Apache Spark system. Thus, on both datasets, more polarities could be identified perfectly with the PTh-IndRNN.

C. Comparative Analysis

The proposed PTh-IndRNN 's text sentiment classification accuracy is analogized to the surveyed works BERT-LSTM (Xiaofeng et. al., 2021, [11]), HL-NBC(Rodrigues & Chiplunkar, 2022, [14]), and AG-LSTM.

Table 4: Comparative analysis of the existing works

Algorithms	Accuracy (%)
BERT-LSTM (Xiaofeng et al., 2021)	84.55
HL-NBC (Rodrigues & Chiplunkar, 2022)	82
AG-LSTM (Deepika & Nirupama Bhat, 2021)	90.42
Proposed PTh-IndRNN (Historical dataset)	95.6287
Proposed PTh-IndRNN (Real-Time dataset)	97.5478

The comparative analysis of the proposed text sentiment classification accuracy on real-time and historical datasets, alongside the existing BERT-LSTM, HL-NBC, and AG-LSTM models, is presented in Table 4. For accuracy, the existing BERT-LSTM and AG-LSTM achieved 84.55% and 90.42%. HL-NBC was unable to outperform the proposed approach, despite utilising a combination of the MapReduce programming model and Hadoop. Thus, the stock market polarity can be predicted effectively using the PTh-IndRNN combined with the proposed HDFS system.

V. CONCLUSION

This paper proposes a novel HDFS-based PTh-IndRNN system for real-time big-data analysis. The mapping, shuffling, and reduction are performed using H-CKMeans, FYSA, and TS-SCOA in the proposed HDFS system.

The performance of the proposed techniques is evaluated on real-time and historical big data datasets. The TS-SCOA and H-C KMeans exhibited enhanced outcomes, which enriched the proposed HDFS system. The proposed LS-DMOA achieved higher fitness values of 5,784, 6,748, 7,912, 8,852, and 9,756, which verified better convergence. Next, the proposed technique acquired higher accuracy, recall, sensitivity, specificity, precision, and F-measure. The PTh-IndRNN required less training time, at 119,542 ms and 116,521 ms, on both datasets, which demonstrated the time efficiency of the proposed framework. Ultimately, the superiority of the proposed Twitter financial text sentiment classification is shown through a comparative analysis. Score values are included in the SentiWordNet dictionary for general words only. Thus, the financially trained dictionary can be used to improve classification accuracy in the future.

DECLARATION

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval or consent to participate, as it presents evidence that is not subject to interpretation.
Availability of Data and Material/ Data Access Statement	Data extracted from Twitter AP and AlphaVantage – Data provider
Authors Contributions	Sowmya R Main Author, Dr. Suneetha K R. Reviewer.

REFERENCES

1. Nousi, C., & Tjortjis, C. (2021). A Methodology for Stock Movement Prediction Using Sentiment Analysis on Twitter and Stock Twits Data. 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM 2021. 1-7 [CrossRef]
2. Gupta, R., & Chen, M. (2020). Sentiment Analysis for Stock Price Prediction. *Proceedings - 3rd International Conference on Multimedia Information Processing and Retrieval, MIPR 2020*, 213–218. [CrossRef]
3. Mehtab, S., & Sen, J. (2020). Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models. *2020 International Conference on Decision Aid Sciences and Application, DASA 2020*, 447–453. [CrossRef]
4. Kesavan, M., Karthiraman, J., Ebenezer Rajadurai, T., & Adhithyan, S. (2020). Stock Market Prediction with Historical Time Series Data and Sentimental Analysis of Social Media Data. *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, 477–482. [CrossRef]
5. Lin, Y.-L., Lai, C.-J., & Pai, P.-F. (2022). Using Deep Learning Techniques in Forecasting Stock Markets by Hybrid Data with Multilingual Sentiment Analysis. *Electronics*, *11*(21), 1-19. [CrossRef]
6. Bazzaz Abkenar, S., Haghi Kashani, M., Mahdipour, E., & Jameii, S. M. (2021). Big data analytics meets social media: A systematic review of techniques, open issues, and future directions. *Telematics and Informatics*, *57*, 1-38. [CrossRef]
7. Shao, C., & Chen, X. (2022). Deep-Learning-Based Financial Message Sentiment Classification in Business Management. *Computational Intelligence and Neuroscience*, *2022*. 1-9. [CrossRef]
8. Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing*, *4*(1), 1–34. [CrossRef]
9. Gupta, Y. K., & Sharma, N. (2020). Propositional aspect comparison between Apache Spark and Hadoop MapReduce for stock market data. *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, 479–483. [CrossRef]
10. Dong, S., & Liu, C. (2021). Sentiment Classification for Financial Texts Based on Deep Learning. *Computational Intelligence and Neuroscience*, 1-9. [CrossRef]
11. Xiaofeng, W., Jinghua, Z., Chenxi, J., & Yiying, J. (2021). Research on sentiment classification of futures predictive texts based on BERT. *Computing*, 1-18. [CrossRef]
12. Jaggi, M., Mandal, P., Narang, S., Naseem, U., & Khushi, M. (2021). Text mining of StockTwits data for predicting stock prices. *Applied System Innovation*, *4*(1), 1–22. [CrossRef]
13. Achyutha, P. N., Chaudhury, S., Bose, S. C., Kler, R., Surve, J., & Kaliyaperumal, K. (2022). User Classification and Stock Market-Based Recommendation Engine Based on Machine Learning and Twitter Analysis. *Mathematical Problems in Engineering*, 1-9. [CrossRef]
14. Rodrigues, A. P., & Chiplunkar, N. N. (2022). A new big data approach for topic classification and sentiment analysis of Twitter data. *Evolutionary Intelligence*, *15*(2), 877–887. [CrossRef]
15. Demirbaga, U. (2021). HTwitt: A Hadoop-based platform for analysing and visualising streaming Twitter data. *Neural Computing and Applications*, 1-16. [CrossRef]
16. Dong, J. (2020). Financial investor sentiment analysis using FPGA and convolutional neural networks. *Microprocessors and Microsystems*, 1-6. [CrossRef]
17. Bourezk, H., Raji, A., Acha, N., & Barka, H. (2020). Analysing the Moroccan Stock Market using Machine Learning and Sentiment Analysis. *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2020*.1-5. [CrossRef]
18. Dubey, A. K., Kumar, A., & Agrawal, R. (2021). An efficient ACO-PSO-based framework for data classification and preprocessing in big data. *Evolutionary Intelligence*, *14*(2), 909–922. [CrossRef]
19. Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2022). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, *13*(7), 3433–3456. [CrossRef]
20. Deepika, N., & Nirupama Bhat, M. (2021). An Efficient Stock Market Prediction Method Based on Kalman Filter. *Journal of The Institution of Engineers (India): Series B*, *102*(4), 629–644. [CrossRef]
21. Aasi, B., Imtiaz, S. A., Qadeer, H. A., Singarajah, M., & Kashef, R. (2021). Stock Price Prediction Using a Multivariate Multistep LSTM: A Sentiment and Public Engagement Analysis Model. *2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings*. 1-9. [CrossRef]
22. Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2021). A novel multi-source information-fusion predictive framework based on deep neural networks for accuracy enhancement in stock market prediction. *Journal of Big Data*, *8*(1), 1-28. [CrossRef]
23. Yasir, M., Afzal, S., Latif, K., Chaudhary, G. M., Malik, N. Y., Shahzad, F., & Song, O. Y. (2020). An efficient deep learning based model to predict interest rate using Twitter sentiment. *Sustainability (Switzerland)*, *12*(4). 1-16. [CrossRef]
24. Zhao, L., Li, L., Zheng, X., & Zhang, J. (2021). A BERT-based Sentiment Analysis and Key Entity Detection Approach for Online Financial Texts. *Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2021*, 1233–1238. [CrossRef]
25. Xu, H., Chai, L., Luo, Z., & Li, S. (2020). Stock movement predictive network via incorporative attention mechanisms based on tweets and historical prices. *Neurocomputing*, *418*, 326–339. [CrossRef]

AUTHORS PROFILE

Sowmya R, Research Scholar Education: pursuing Ph.D. program in Computer Science and Engineering. The publication includes "Data Mining with Big Data" – DOI: 10.1109/ISCO.2017.7855990, Analysis and Verification of Video Summarisation using Shot Boundary Detection, American International Journal of Research in Science, Technology, Engineering & Mathematics, ISSN (Print): 2328-3491, ISSN (Online): 2328-3580.

Dr. Suneetha K R, Professor, Education: Ph.D. in Computer Science and Engineering. The publication includes "Web Log Mining using Improved Version of Apriori Algorithm", International Journal of Computer Applications, 'Classification of Web Log Data to Identify Interested Users Using Decision Trees', International Journal of Ubiquitous and Communication Journal, "Performance evaluation of compact prediction tree algorithm for webpage prediction", IEEE International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE'20) VIT, Vellore, India, "Applications of Association Rule Mining Algorithms in Deep Learning", Accepted in IEEE International conference on Electronics and Sustainable Communication Systems (ICESCS 2020) Hindustan Institute of Technology, CITATIONS-344, HI INDEX-6, I10 INDEX 5