

A novel scalable deep ensemble learning framework for big data classification via MapReduce integration

Kesavan Mettur Varadharajan^{1,2}, Josephine Prem Kumar³, Nanda Ashwin²

¹Department of Computer Science and Engineering, Visvesvaraya Technological University, Belagavi, India

²Department of Computer Science and Engineering, East Point College of Engineering and Technology, Bengaluru, India

³Department of Computer Science and Engineering, Cambridge Institute of Technology, Bengaluru, India

Article Info

Article history:

Received Mar 14, 2024

Revised Oct 30, 2024

Accepted Nov 14, 2024

Keywords:

Big data

Deep ensemble learning framework

Ensemble deep network

Large-scale data

MapReduce integration

SDELF-BDC

ABSTRACT

Big data classification involves the systematic sorting and analysis of extensive datasets that are aggregated from a variety of sources. These datasets may include but are not limited to, electronic records, digital imaging, genetic information sequences, transactional data, research outputs, and data streams from wearable technologies and connected devices. This paper introduces the scalable deep ensemble learning framework for big data classification (SDELF-BDC), a novel methodology tailored for the classification of large-scale data. At its core, SDELF-BDC leverages a Hadoop-based map-reduce framework for feature selection, significantly reducing feature-length and enhancing computational efficiency. The methodology is further augmented by a deep ensemble model that judiciously applies a variety of deep learning classifiers based on data characteristics, thereby ensuring optimal performance. Each classifier's output undergoes a rigorous optimization-based ensemble approach for refinement, utilizing a sophisticated algorithm. The result is a robust classification system that excels in predictive accuracy while maintaining scalability and responsiveness to the dynamic requirements of big data environments. Through a strategic combination of classifiers and an innovative reduction phase, SDELF-BDC emerges as a comprehensive solution for big data classification challenges, setting new benchmarks for predictive analytics in diverse and data-intensive domains.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Kesavan Mettur Varadharajan

Department of Computer Science and Engineering, Visvesvaraya Technological University

Belagavi, India

Email: kes_mv@rediffmail.com

1. INTRODUCTION

In recent years, big data has become the main tech talk between academia and practitioners in the digital competitive playyard. Big data is an important asset that attracts the attention of many chief executive officers (CEOs) in different organizations to gain faster insights and high revenue [1]. The journey of big data started when many organizations recognized that the large volume of their data exceeded the capabilities of their organizations, process, capacity, structure, technology infrastructure, and governance. They struggled to deal with the requirements for analyzing the high volume of various data [2].

According to statistical reports, the number of users on different social media platforms has reached more than 2 billion. WhatsApp, for example, has over 600 million users, more than half a billion photos, and one hundred million videos transferred and shared between users daily [3]. Also, due to the huge advances in smartphone technology, it has become easier for users to share images and write posts on such social media

platforms. Some reports show that the number of posts on Twitter in 2007 was 5K, this number became around 500 million after about 6 years, which indicates the massive amount of available data on social media in general. This amount of data is not restricted to social media, as many other platforms generate and store huge data volumes [4]. This amount of data needs to be processed and analyzed to use it for building useful knowledge discovery and machine learning big data -based applications, like facial big data applications, signal big data, and various industry big data -based applications [5]. Volume (big), variety, and velocity are among the most distinguishing characteristics of big data, and as a result, it is attractive to have an efficient classification/prediction system to learn from such big data. Such applications include, but are not limited to, medical, financial, security, and image-based applications [6].

Big data analysis offers service tools, such as Hadoop distributed file system (HDFS) which supports managing and storing huge amounts of data, fast automated decisions and decreases the risks of human estimations. The HDFS is accepted as the most widely used dataset tool that supports redundancy, reliability, scalability, parallel processing, and distributed architecture systems [7], and is designed to handle different big data types; structured, semi-structured, and unstructured. Moreover, the Hadoop MapReduce job-scheduling algorithm [6] supports clustering big data in a spread network environment. In addition, big data analysis provides significant opportunities for solving different information security problems by using Hadoop technologies and HDFS tools. The data value that is generated from big data through the analysis phase is of extreme importance [8]. During the previous few decades, classifiers have been intensively studied and analyzed. Classifiers are widely used in many modern applications as key computer technology. Many classifiers exist, such as k-nearest neighbor (KNN), support vector machine (SVM), naive Bayes (NB), random forest (RF), decision tree (DT), and many more. In terms of accuracy and time consuming building a trained model, each classifier has advantages and limitations; some are more effective with specific datasets than others, and hence there is no optimal classifier that can perfectly classify all types of data [8].

Deep learning architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), including long short-term memory networks (LSTMs), offer tailored advantages for big data classification. CNNs excel in spatial hierarchy learning from images, enabling robust feature extraction without manual intervention, making them ideal for image-based classification tasks. Their structure, composed of convolutional, pooling, and fully connected layers, ensures efficiency and scalability, even with high-dimensional data. On the other hand, RNNs and LSTMs shine in sequence data analysis, such as time series, speech, or text, by effectively capturing temporal dependencies and handling variable-length inputs. LSTMs further mitigate the vanishing gradient problem, allowing for the learning of long-term dependencies. These architectures collectively enhance big data classification by offering scalable, efficient, and accurate modeling capabilities, capable of extracting deep insights from complex datasets, thereby driving innovation across diverse domains [9].

The integration of deep learning with MapReduce, a programming model for processing and generating large data sets with a parallel, distributed algorithm on a cluster, offers a powerful approach for enhanced classification tasks on big data. This integration leverages the computational efficiency of MapReduce to handle the massive scalability requirements of deep learning algorithms, allowing for the distributed processing of data across multiple nodes, which significantly speeds up the training of complex models on large datasets. For instance, deep learning models, such as CNNs for image classification or RNNs for sequence data, can be trained more efficiently using this integrated approach, enabling more sophisticated and accurate classification capabilities. However, this integration is not without its limitations. The complexity of configuring and managing a distributed computing environment can introduce overhead and potential bottlenecks, especially in terms of network communication and data transfer speeds. Additionally, the inherent challenges of parallelizing deep learning algorithms, such as synchronization of model updates and the non-uniform distribution of data, can affect the efficiency and scalability of the solution. Despite these challenges, the combination of deep learning and MapReduce holds significant promise for advancing big data classification by harnessing the strengths of both technologies [10]–[12].

The exponential growth of big data across various sectors, coupled with its profound impact on decision-making, innovation, and competitive advantage, has sparked significant interest among researchers and practitioners alike. This burgeoning interest is rooted in the realization that big data, characterized by its volume, velocity, and variety, holds the key to unlocking novel insights and fostering advancements across various fields including healthcare, finance, security, and beyond. The motivation behind this research stems from the challenges and opportunities presented by the vast amounts of data generated by social media platforms, IoT devices, and digital transactions, which exceed the processing capabilities of traditional data analysis tools. As organizations strive to harness the full potential of big data for knowledge discovery, machine learning, and predictive analytics, there arises a critical need for efficient classification and prediction systems capable of managing this complexity. The integration of deep learning architectures and MapReduce frameworks presents a promising avenue for addressing these challenges, offering scalable, efficient solutions for big data classification. This research is driven by the ambition to contribute to the

development of advanced big data analysis tools that can not only manage the sheer scale of data but also provide actionable insights, thereby enabling organizations to make informed decisions, enhance operational efficiency, and achieve strategic goals.

Advanced deep ensemble learning architecture: devised a sophisticated framework that integrates a suite of deep learning classifiers, orchestrated within a Hadoop-based infrastructure. This approach is engineered to refine classification by conducting precise feature selection and employing a hybrid of algorithmic strategies. The result is a substantial improvement in accuracy and efficiency, particularly suited to the complexities of big data classification. Innovative hybrid heuristic optimization technique: introduced a cutting-edge optimization method that is implemented within a map-reduce framework, this technique enhances the selection and classification of features, thereby elevating the effectiveness and precision of big data systems.

The research is organized in this paper into four sections. The first section gives a brief overview of the big data classification process, and the second section discusses the existing literature and shortcomings associated with each. In the third section, a model is proposed to overcome this and in the fourth section a comparison is carried out with the existing and the proposed techniques, and the results are tabulated in the form of graphs.

2. RELATED WORK

In the realm of big data classification, recent studies have innovated methods to enhance feature selection and classification accuracy in large datasets. For example, a novel approach integrates MapReduce with feature subset selection and hyperparameter-tuned deep belief networks (DBN), aiming to address the complexities of big data processing and improve classification performance. A classification model named random forest-based feature selection and extraction (RFSE)-gated recurrent unit (GRU) [13] was developed, integrating GRU with a strategic approach for feature selection and data balance. This model leverages the RF algorithm to identify the most impactful features for classification. To mitigate the challenges posed by data imbalance, it employs a combination of the synthetic minority oversampling technique (SMOTE) for oversampling and the edited nearest neighbor (ENN) technique for under-sampling, enhancing the model's classification accuracy. Technologies such as text and data mining, online and mobile mining, process mining, statistical analysis, network analytics, social media analytics, audio and video analytics, and web analytics are all included in big data analytics. To enhance healthcare data sets, a range of data mining techniques may be used, such as summarization, association rules, clustering, classification, anomaly detection, and large-scale data visualization [14]. Modern data analytics algorithms employ certain data properties to evaluate sensor and high-speed data streams. Big data has several applications, including improving diagnosis, averting illness, monitoring patients from a distance, reducing hospital stays, integrating medical imaging, reducing fraud, strengthening data security, and more. Introducing a brand-new deep learning-based mobile traffic data categorization solution.

The proposed RFSE-GRU model is a classification model that incorporates feature selection, data balancing, and the GRU algorithm [15]. The RF method selects features according to their significance for the classification process. Additionally, the combination of the SMOTE oversampling strategy with the ENN under-sampling method reduced the negative impact of data imbalance on classification performance. This paper presents a redesigned KNN technique and compares it to the traditional KNN algorithm. Within the vicinity of the query instance, the traditional KNN classifier is employed to do the classification, assigning weights to each class. The technique considers the class distribution surrounding the query instance to prevent the weight assignment from adversely affecting the outliers [16]. The current study addresses the shortcomings of the traditional KNN technique with large datasets by introducing an improved KNN strategy that combines density cropping and cluster denoising. This approach uses clustering to improve denoising processing and boost the classification efficiency of the KNN algorithm by accelerating the KNN search without compromising the algorithm's classification accuracy. Jiang and Li [17] proposal's edge processing unit is composed of two primary components: a data transmission unit that uses appropriate communication methods to send data to railway control centers based on the type of data received, and a data classification model that separates internet of things (IoT) data into two categories: maintenance-critical data (MCD) and maintenance-non-critical data (MNCD). For multiclass classification with large datasets, we may reduce the temporal complexity and increase the computational efficiency of energy balance-related behaviors (EBRBS) by using a domain division-based rule reduction technique, a more straightforward evidential reasoning algorithm, and a method to do away with rule weight calculation. This is a Micro-EBRBS, which is an EBRBS that has been shrunk down. Furthermore, Apache Spark, a well-liked cluster computing tool, is used in the development of micro-EBRBS's parallel rule generation and inference methodologies for big data multiclass classification issues [18]. Offering a novel approach to fault line selection that uses big data and feature classification to overcome the shortcomings of existing techniques. This method addresses the fault

line selection issue as a classification task by feeding large datasets associated with faults into the classifiers during training. The four main steps in the procedure are data gathering, training, classification, and assessment. Data preparation is in charge of preprocessing and data collection, whereas training employs processed data to train classifiers [19], [20] presents a more advanced KNN technique and compares it with the traditional KNN algorithm. Within the vicinity of the query instance, the traditional KNN classifier is employed to do the classification, assigning weights to each class. The technique considers the class distribution surrounding the query instance to prevent the weight assignment from adversely affecting the outliers. By including cluster denoising and density cropping, this study presents an improved KNN method that overcomes the shortcomings of the traditional KNN strategy when working with large datasets. This approach enhances denoising capabilities and increases KNN algorithm efficiency by employing clustering to speed up the search for nearest neighbors without losing classification accuracy [21]–[24].

3. PROPOSED METHODOLOGY

The proposed scalable deep ensemble learning framework for big data classification (SDELF-BDC) methodology begins with the first step which is feature selection, where relevant data attributes are identified using a Hadoop-based map-reduce framework to minimize feature length. The chosen features are then processed through a deep ensemble model that utilizes a variety of deep learning classifiers such as CNN, DBN, LSTM, extreme learning machine (ELM), and deep neural network (DNN). These classifiers are applied conditionally, based on their suitability to the data characteristics. In the classification and reduction phase, the results from each classifier are combined, and an optimization-based ensemble approach is employed to refine the results further. This combination aims to generate a decisive strategy that maximizes the prediction metrics within the map-reduce framework. The final output is a robust prediction model. Figure 1 shows the block diagram.

3.1. Feature selection

The proposed Hadoop-based map reduction model chooses the relevant features from the map phase which is generated by the proposed model to minimize each feature's length which gets many effective features. The relevant features are chosen using the proposed algorithm that minimizes the computation time and overfitting issue. This is essential to enhance the accuracy, with added relevant features used to reduce the number of input variables that eliminate non-relevant features. These features are selected from two different datasets. Thus the optimal features for this model are expressed as $h_u^{\text{fine-tune}}$, wherein $u = 1, 2, \dots, U$ using the proposed algorithm.

3.2. Proposed algorithm

The proposed heuristic optimization method, which is generated by the map reduction framework-based classification system, adjusts several parameters to get the best-projected results. The reason this model chooses a novel optimization algorithm is that it can guarantee quick convergence and prevent problems with local optima. However, its inability to get the best outcomes worldwide renders it ineffective for a variety of optimization tasks. It has been coupled with an optimization approach to overcome the shortcomings of the present optimization approach because of its capacity to boost performance when addressing engineering optimization challenges and to increase efficiency in global search. The proposed algorithm increases the classification using efficiency by utilizing the map-reduce architecture. The parameters Ru1 and Ru2 in the proposed algorithm are computed using the deviation of the optimization approach and another optimization approach. The outcome of the deviation is shown in (1). The SD for Ru1 and Ru2 are evaluated to be $SD(Ru1, Ru2)$, finally the updated value is denoted by D as shown in (2).

$$D = \min(Ru1, Ru2) + SD(Ru1, Ru2) \quad (1)$$

$$Finalout = Finalout + D \quad (2)$$

Here Finalout denotes the outcome of the solution. This method utilizes parameters to define a set of coyote packs, or Re_{il} . Each pack consists of unique coyotes, EA_{am} , showcasing specific social behaviors within their group, il , over a specific time frame, V_m . An alpha coyote, known for its superior adaptability and behavior that suits changing environmental conditions, assumes leadership in every pack. In (3) summarizes the leadership characteristics of the alpha coyote. The information relevant to the coyotes gained from the groups for performing cultural tendency is shown in (4). Here Se^{il, V_m} depicts the ranked social status of the coyote involved through the search dimension km within the pack il in the period V_m . The birth rate is evaluated and termed as a life event for the new coyote denoted by the (5).

$$\alpha EA^{il,Vm} = Bs_{am}^{il,Vm} \text{ for min } Hs_{am}^{il,Vm} \quad (3)$$

$$evEA_{km}^{il,Vm} = \begin{cases} Se_{\frac{EA_{am}+1}{2}}^{il,Vm}, k & EA_{am} \text{ is odd} \\ Se_{\frac{EA_{am}+1}{2}}^{il,Vm}, km & \text{orelse} \end{cases} \quad (4)$$

$$dtEA_{km}^{il,Vm} = \begin{cases} Bt_{tp1jl}^{il,Vm} & tt_{jl} < D_{uw} \text{ or } km = km_1 \\ Bt_{tp2jl}^{il,Vm} & tt_{jl} < D_{uw} \text{ or } km = km_2 \\ tf_{jl} & \text{else} \end{cases} \quad (5)$$

The design dimension denoted by km_1 or km_2 . The term D_{uw} is evaluated to scatter probabilities and D_{cr} is indicated as the association of probabilities. The random variables denoted by tt_{jl} and tf_{jl} in the range $[0, 1]$. The final status is determined by correlating the prior and upgraded status as shown in (6). The optimal solution is finalized through the condition which is incorporated by this searching dimension. According to the hunting mechanism the prey enclosed by the alpha parameters by determining their position with the assistance of a specific agent, the encircling strategy is given by (7).

$$Bt_{am}^{il,Vm+1} = \begin{cases} Bt_{am}^{il,Vm+1} & py_{Ht_{am}^{il,Vm+1} < Ht_{am}^{il,Vm}} \\ Bt_{am}^{il,Vm} & \text{else} \end{cases} \quad (6)$$

$$ms_{tr}(ws + 1) = \mu_1 \sum_{0=1}^{ps} \frac{[\tau_q(ws) - ms_{tr}(ws)]}{ps} - ms_*(ws) \quad (7)$$

The arbitrary number is denoted by μ_1 in the range $[-2, 2]$ the best solution observed from the previous step is denoted by $ms_*(ws)$ whereas the present solution is ms_{tr} and the population is denoted by W_s . The subset of the solution is denoted by $\tau_q(ws)$, the new position within the solution is denoted by $ms_{tr}(ws + 1)$ whereas the random variable is denoted by ps , they chase the prey by tracking the position they are within, this feature is denoted in (8). The search agent is referred to as the $ms_g(ws)$, $ms_{tr}(ws + 1)$ depicts the movement and the arbitrary number denoted by μ_1 in the range $[-1, 1]$. This classification is observed as in (9) with the survival rate the computation is performed in (10).

$$ms_{tr}(ws + 1) = ms_*(ws) + \mu_1 * zs^{\mu_2} * (ms_g(ws) - ms_{tr}(ws)) \quad (8)$$

$$ms_{tr}(ws + 1) = 0.5[as^{\mu_2} * ms_g(ws) - (-1)^{\gamma} * ms_{tr}(ws)] \quad (9)$$

$$ut(tc) = \frac{f_{max} - f(t)}{f_{max} - f_{min}} \quad (10)$$

The $f(t)$ designates the consistent value of t^{vj} search agent. $Ms_*(ws)$ is depicted by the best solution through the previous iteration. The variables f_{max} and f_{min} which represents the worst-case optimal fitness value, whereas the optimal fitness values and the parameter γ are represented by the binary number. The algorithm is given as shown in Algorithm 1. Figure 2 shows the proposed algorithm flowchart.

Algorithm 1. Proposed algorithm

- Step 1 Initialization of population
- Step 2 Compute fitness for all solutions
- Step 3 If termination is not met, then:
 - For every solution in the population:
 - Update the position following the procedure
 - Evaluate the Ru1
 - Update the position according to the procedure
 - Evaluate the Ru2

```

- Calculate the final deviation using (1)
- Continue execution
Else
- Upgrade final position by (2)
- Improvisation
Return to step 3 to check for termination again
Else:
- Obtain the best optimal solution and terminate the algorithm

```

Step 4

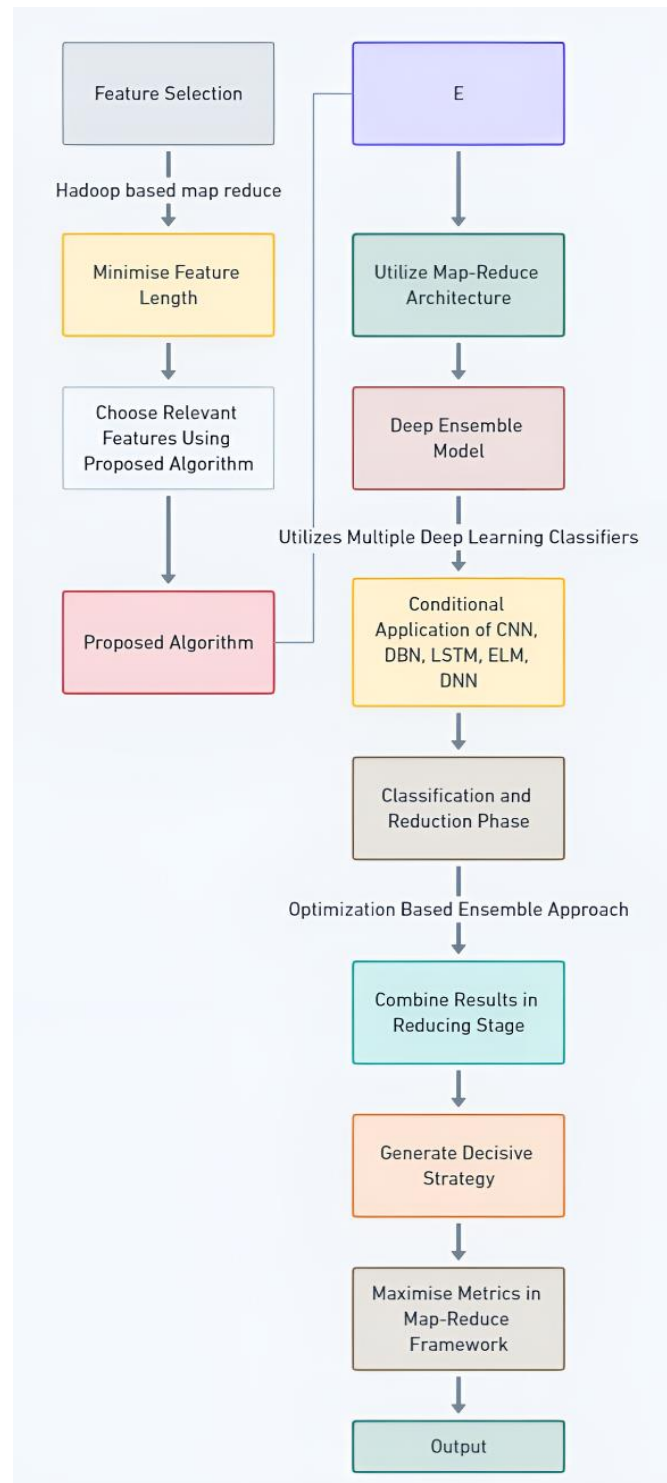


Figure 1. Block diagram

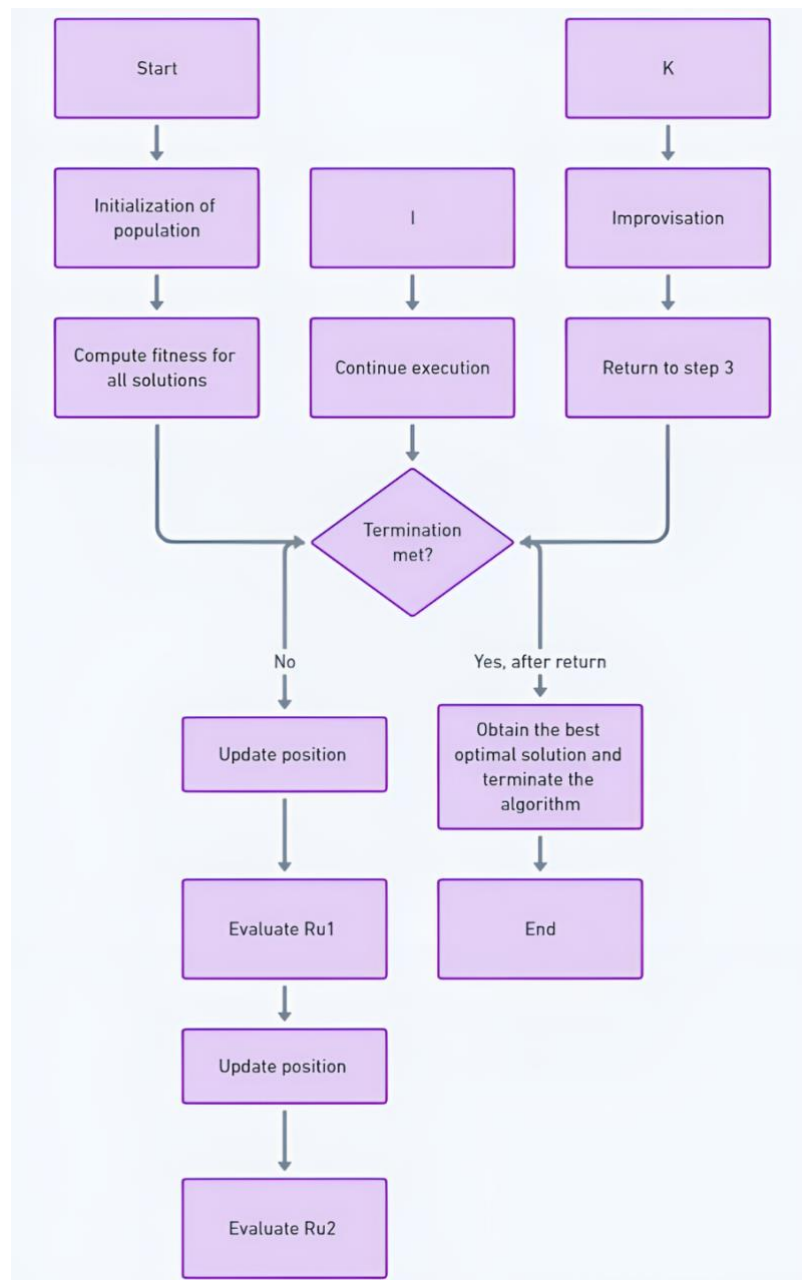


Figure 2. Proposed algorithm flowchart

3.3. Deep ensemble model

The deep ensemble model for the classification algorithm harnesses the power of multiple deep learning classifiers to predict physical activities based on sensor data, such as accelerometer readings. This approach is designed to improve prediction accuracy by leveraging the collective strengths and compensating for the weaknesses of individual models. The process begins with the pre-processing of input data, ensuring it is normalized and partitioned into training, validation, and test sets to facilitate effective model training and evaluation.

The core of the algorithm lies in its conditional application of different deep learning techniques, each selected based on the specific characteristics of the data at hand. For data sets where spatial pattern recognition is paramount, the CNN is employed. CNNs are adept at extracting hierarchical spatial features from data, making them ideal for analyzing images or sensor data that have a spatial structure. On the other hand, when the data involves temporal dependencies or sequences where the order of data points is crucial, the LSTM network is utilized. LSTMs excel in remembering information over extended periods, making

them suited for time-series data or sequences where past information influences future predictions. For scenarios demanding rapid training and straightforward predictions without the need for capturing complex relationships or time dependencies, the ELM offers a fast and efficient solution. ELMs are single-layer feedforward networks that can achieve high-speed training and generalization performance, making them suitable for simpler tasks or when computational efficiency is a priority.

The ensemble model's strategy involves evaluating the performance of each classifier on the validation set and then combining these models in a way that leverages their strengths. If there's consensus among the models, the algorithm might simply take the majority vote as the final prediction. However, in cases of disagreement, the algorithm opts for a more nuanced approach by weighting the outputs of each model based on their validation accuracy and computing a weighted average prediction. This method ensures that more accurate models have a greater influence on the final prediction, enhancing the overall prediction performance of the ensemble. After selecting the combination strategy based on model agreement and performance weighting, the ensemble model is trained on the entire training set. This integrated training approach allows the ensemble to learn from the comprehensive data available, ensuring it is well-adjusted to make accurate predictions. The final step involves evaluating the ensemble model's performance on the test set, providing an assessment of its ability to generalize to unseen data. For new inputs, the model predicts data classes by preprocessing the data to match the training format and then applying the trained ensemble model to generate predictions.

This algorithm represents an efficient approach to predictive modeling in the context of data classification, leveraging the unique capabilities of different deep learning models to create a robust, accurate, and efficient tool for data class prediction. By combining the predictions from various models, the ensemble method achieves a balance between the depth of learning and computational efficiency, ultimately enhancing the reliability and accuracy of data class predictions. Algorithm 2 shows the deep ensemble algorithm. Figure 3 deep ensemble flowchart for prediction.

Algorithm 2. Deep ensemble algorithm

```

Input      Dataset
Step 1     Pre-processing:
            - Normalize the data to ensure uniformity and better model training
Step 2     Initialization:
            - CNN_model=Train a convolutional neural network on the data.
            - DBN_model=Train a deep belief network on the data.
            - LSTM_model=Train a long short-term memory network on the data.
            - ELM_model=Train an extreme learning machine on the data.
            - DNN_model=Train a deep neural network on the data.
Step 3     Define the ensemble prediction technique:
            - Collect predictions from each model.
            - For each instance in the test data, do the following:
              • if CNN_model confidence > threshold, use CNN_model prediction.
              • else if DBN_model confidence > threshold, use DBN_model prediction.
              • else if LSTM_model confidence > threshold, use LSTM_model prediction.
              • else if ELM_model confidence > threshold, use ELM_model prediction.
            - else, use DNN_model prediction as a default.
Step 4     Evaluate the performance of each model on the validation set.
Step 5     Combine predictions for the final decision:
            - if the models agree on the prediction:
              • Accept the majority vote as the final prediction.
            - else
              • Weight the outputs based on their validation accuracy, and compute a weighted average prediction.
Step 6     Train the ensemble model on the entire training set using the selected strategy from step 5.
Step 7     Evaluate the ensemble model on the test set to assess its prediction performance.
Step 8     for new sensor data inputs:
            - Pre-process the data as in step 1.
              • Predict the classes using the trained ensemble model.
output     The predicted class.

```

3.4. Classification and reduction phase

The proposed framework for monitoring develops an optimization-based ensemble approach for the prediction of physical activities through the optimal features known as $h_u^{\text{fine-tune}}$ from the proposed algorithm. The proposed algorithm incorporates all this in the combining phase. The developed framework is used to maximize the metrics in the map-reduce framework for prediction. The proposed map-reduce system aggregates

results from the combining stage to be processed in the reduction stage, where all outcomes are combined to generate a decisive strategy. This framework is crucial for big data classification with different application.

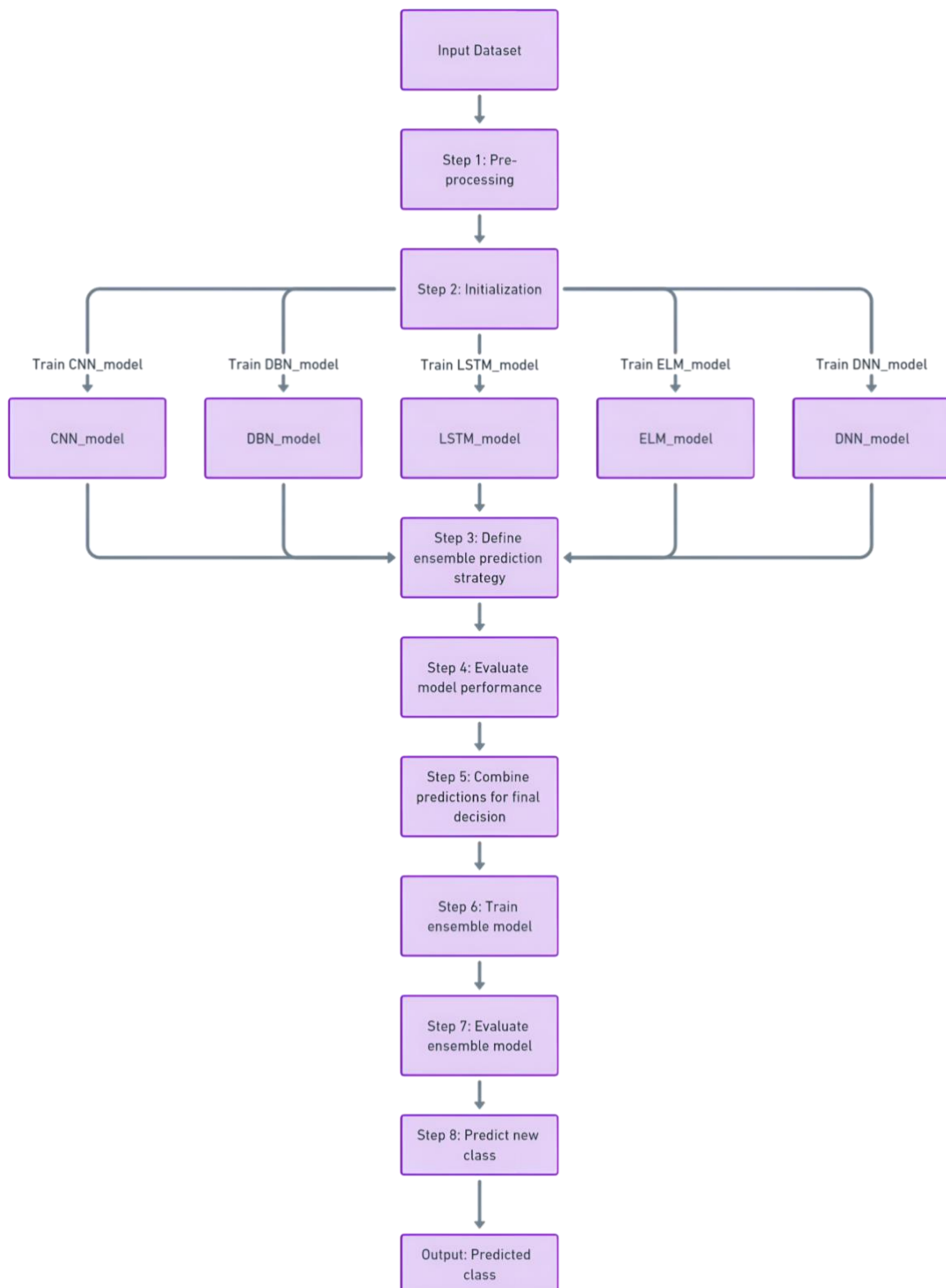


Figure 3. Deep ensemble flowchart for prediction

4. PERFORMANCE EVALUATION

The performance evaluation involves assessing the accuracy of different methodologies, including the proposed model, across four datasets: supersymmetry (SUSY), higgs, modified national institute of standards and technology (Mnist), and United States postal service (USPS). Additionally, the evaluation includes comparing the time taken and speed-up achieved specifically for the SUSY and Higgs datasets. The results are presented comprehensively through tables and graphs to provide a clear comparison of the methodologies' effectiveness across the various datasets and performance metrics.

4.1. Accuracy

Analyzing the performance of various methods on the SUSY dataset, we see a range of accuracy scores from these classification algorithms. Multivariate decision tree 2 (MDT2) demonstrates the highest accuracy among the majority of the traditional algorithms, with a score of 0.749, suggesting that its method of classification is particularly well-suited to this dataset. Constant-time ensemble learning classifier (CTELC) [(existing system) ES], an ensemble strategy, slightly outperforms MDT2 with a score of 0.758, indicating the effectiveness of combining multiple models to enhance predictive performance. Notably, the proposed system (PS) method stands out with the highest accuracy of 0.8063, which could imply an advanced feature selection or optimization process that significantly benefits the model's performance on this dataset. On the other end of the spectrum, national benchmark tests (NBT) scores the lowest at 0.594, which may indicate that its probabilistic approach is less effective for the patterns present in the SUSY data. The identical scores of moving range k-nearest neighbor (MR-KNN) and KNN-IS (KNN design based on spark) suggest similar capabilities in handling the data, potentially due to shared reliance on the proximity of data points. Fuzzy classifiers show robustness with a score of 0.735, which might be due to their ability to handle uncertain or imprecise information. Lastly, furthest-pair-based binary search tree (FPBST) and minimum/maximum norms-based binary tree (MNBT) are tied at 0.71, which could suggest a parity in their ability to generalize or an underlying similarity in their approach to the SUSY dataset. Table 1 and Figure 4 show the accuracy comparison for the SUSY dataset.

Table 1. Accuracy comparison

Method	SUSY
MDT1 [25]	0.729
MDT2 [25]	0.749
FPBST [26]	0.71
MR-KNN [27]	0.694
KNN-IS [28]	0.694
Fuzzy [29]	0.735
NBT [30]	0.594
MNBT [30]	0.71
CTELC [ES] [31]	0.758
PS	0.8063

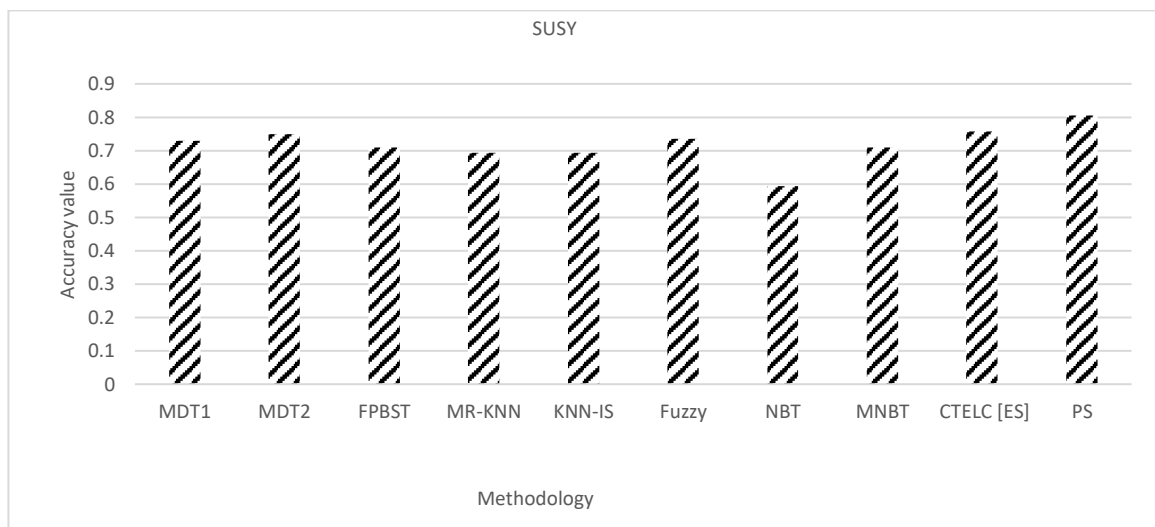


Figure 4. Accuracy comparison for the SUSY dataset

In the analysis of classification methods on the Higgs dataset, the PS method significantly outshines all others with an accuracy of 0.7643, suggesting that its advanced approach, possibly involving sophisticated feature selection or optimization techniques, is exceptionally well-suited for this type of complex data. MDT2 emerges as the runner-up, with an accuracy score of 0.6, indicating its competency but also room for improvement. CTELC [ES], leveraging ensemble strategies, shows moderate success at 0.59, underscoring the potential of combining multiple models to tackle intricate datasets. Multivariate decision tree 1 (MDT1) and FPBST display near-identical performances, hovering just above the 0.58 mark, which may point to similarities in their classification strategies or their collective median efficacy in this context. At the lower end of the spectrum is MNBT, with an accuracy of only 0.529, hinting at its struggle with the dataset's complexity and the need for refinement in its method to better capture the underlying patterns in the Higgs data. The wide range of accuracies reflects the diverse capabilities of these methods when applied to the challenging task of classifying data in high-energy physics. Table 2 and Figure 5 show the accuracy comparison for the Higgs dataset.

Table 2. Accuracy comparison for the Higgs dataset

Method	HIGGS
MDT1 [25]	0.581
MDT2 [25]	0.6
FPBST [26]	0.582
MNBT [30]	0.529
CTELC [ES] [31]	0.59
PS	0.7643

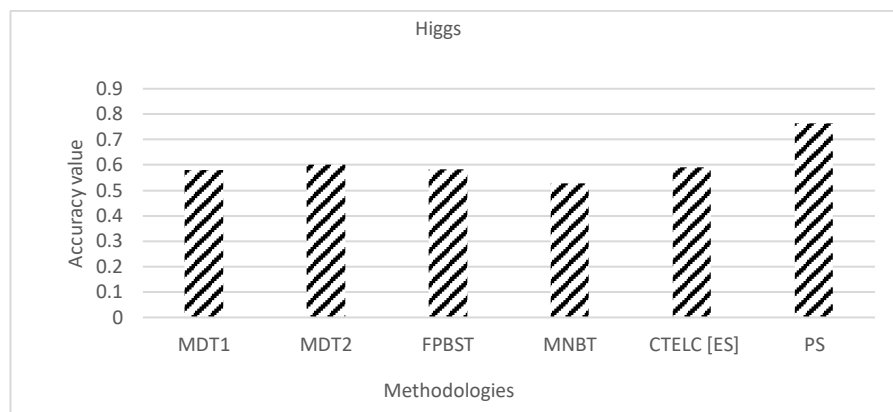


Figure 5. Accuracy comparison for the Higgs dataset

Analyzing the classification performance on the Mnist dataset, it is apparent that the PS method outperforms the others with a notable accuracy of 0.9463, indicating its robustness and potentially more advanced feature processing capabilities. CTELC [ES] also performs admirably, showing a high degree of accuracy at 0.868, which suggests the effectiveness of ensemble methods in handling image data. Both FPBST and MNBT yield strong results, with 0.855 and 0.858 respectively, possibly due to their ability to capture the essential features within image data. LC-KNN stands out as the more effective of the two KNN variations with an accuracy of 0.839, indicating that its approach to leveraging locality in data is beneficial. In stark contrast, NBT significantly underperforms with an accuracy of just 0.19, signaling that its method may be unsuitable for the intricacies of image-based datasets like Mnist. This spread of performance metrics showcases the importance of choosing the right algorithm for the dataset at hand, with some methods distinctly more suited to the complex patterns present in handwritten digit recognition. Table 3 and Figure 6 show the accuracy comparison for the Mnist dataset.

Table 3. Accuracy comparison for the Mnist dataset

Dataset	RC-KNN [27]	LC-KNN [27]	FPBST [26]	NBT [30]	MNBT [30]	CTELC [ES] [31]	PS
Mnist	0.722	0.839	0.855	0.19	0.858	0.868	0.9463

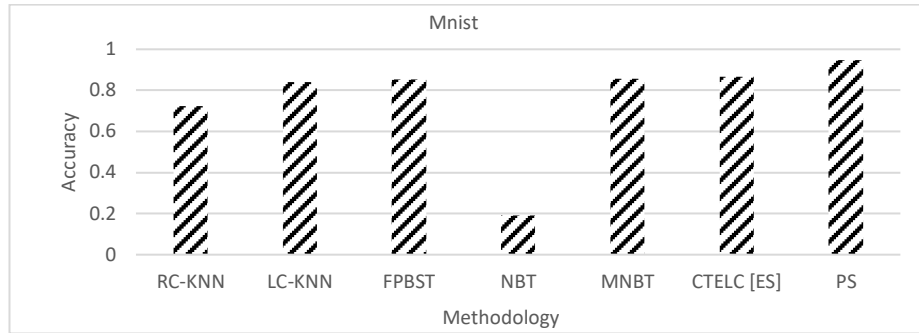


Figure 6. Accuracy comparison for the Mnist dataset

The performance metrics for classification methods on the USPS dataset reveal that the PS method achieves exceptional accuracy at 0.9943, far surpassing the other techniques and indicating its superior handling of postal digit recognition tasks, likely through sophisticated feature learning and selection strategies. Random clustering-k nearest neighbor (RC-KNN) also shows impressive performance with an accuracy of 0.95, suggesting that its radius-based approach to KNN is very effective for this dataset. FPBST and local centralities-based k nearest neighbor (LC-KNN) follow with solid accuracies of 0.936 and 0.903, respectively, indicating that feature-based and locality-conscious approaches are beneficial in classifying the handwritten digits in the USPS dataset. NBT's performance is fairly good at 0.873, but MNBT falls significantly behind with an accuracy of 0.336, which may point to a fundamental mismatch between its modeling approach and the dataset's characteristics. CTCLC [ES], despite being an ensemble method, shows a lower-than-expected accuracy of 0.864, hinting that the specific ensemble technique used may not be fully optimized for this type of data. Overall, these results emphasize the importance of algorithm selection in machine learning tasks, where the PS method's advanced capabilities lead to a clear advantage in accurately classifying the USPS dataset. Table 4 and Figure 7 show the accuracy of the USPS dataset.

Table 4. Accuracy comparison for the USPS dataset

Dataset	RC-KNN [27]	LC-KNN [27]	FPBST [26]	NBT [30]	MNBT [30]	CTCLC [ES]	PS
USPS	0.95	0.903	0.936	0.873	0.336	0.864	0.9943

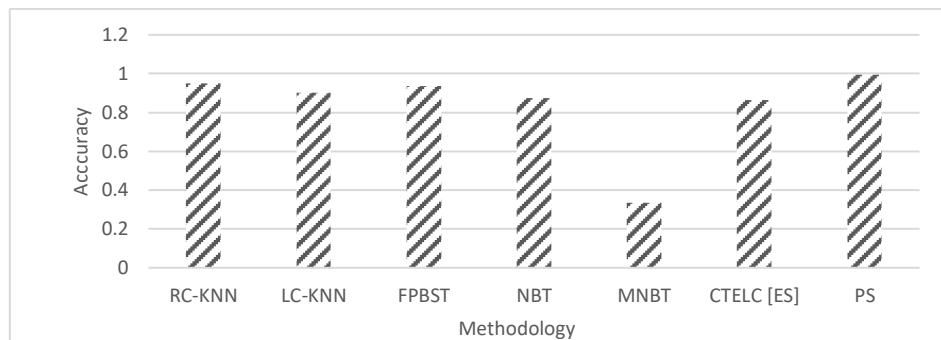


Figure 7. Accuracy comparison for the USPS dataset

4.2. Time comparison

Figure 8 shows the time comparison of the SUSY dataset for the existing system with the proposed system. The analysis shows that the proposed system takes less time for execution with the proposed system. Henceforth showing that the proposed system ensures better performance in comparison with the existing system. Figure 9 shows the time comparison of the USPS dataset for the existing system with the proposed system. The analysis shows that the proposed system takes less time for execution with the proposed system. Henceforth showing that the proposed system ensures better performance in comparison with the existing system.

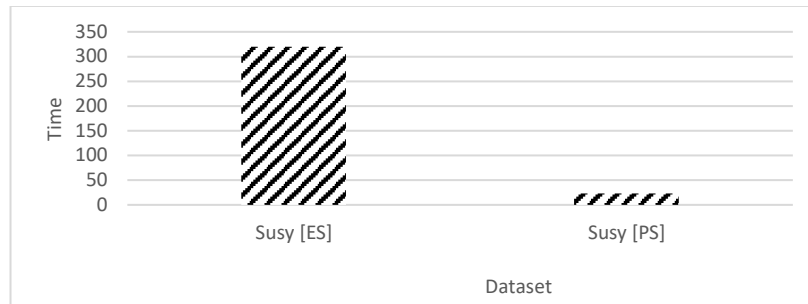


Figure 8. Time comparison for the SUSY dataset

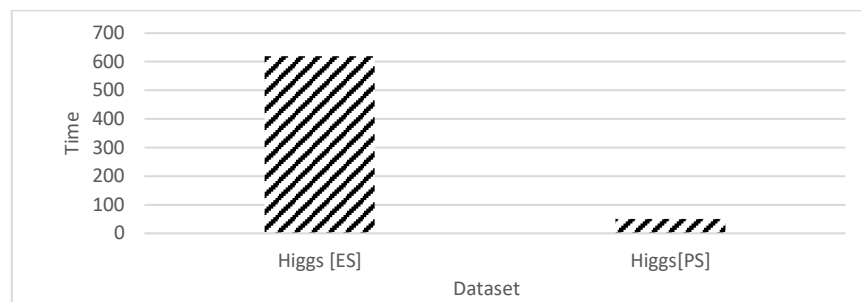


Figure 9. Time comparison for the Higgs dataset

4.3. Speed-up comparison

The speed-up comparison for the SUSY dataset illustrates the relative performance improvements of various methodologies. MDT1 [25] and MDT2 [25] represent the baseline performance, with MDT1 [25] showing the lowest speed-up and MDT2 [25] displaying a minor increase. NBT demonstrates a significant enhancement in speed-up, nearly tripling the value of MDT2 [25]. MNBT, while lower than NBT, still holds a considerable gain over the MDT methodologies. ES presents a further improvement, suggesting a refinement over MNBT. PS stands out with the highest speed-up value, indicating it as the most efficient method among those compared. The chart suggests that, in the context of performance acceleration, methodologies have evolved from MDT1 to PS with notable enhancements in speed-up, culminating in PS as the most superior method in this comparison. Figure 10 shows the speed-up comparison for the SUSY dataset.

The speed-up comparison for the USPS dataset illustrates the relative performance enhancements in comparison with six different methodologies. The methodologies, MDT1 and MDT2, show minimal speed-up, indicating a marginal gain in performance efficiency. Conversely, NBT demonstrates a significant improvement, doubling the speed-up value observed in MDT2, which suggests a considerable enhancement in performance. MNBT's [30] performance is on par with NBT [30], maintaining the gains achieved. ES, however, represents a decrease in speed-up compared to NBT [30] and MNBT [30], implying a reduction in efficiency. The most striking observation is the performance of PS, which towers over the other methodologies with the highest speed-up value. This suggests that PS is substantially more efficient than the rest, potentially offering a performance improvement that is several folds higher. Overall, the chart depicts PS as the standout methodology for speed-up, with NBT [30] and MNBT [30] also showing strong performance gains. Figure 11 speed-up comparison for USPS dataset.

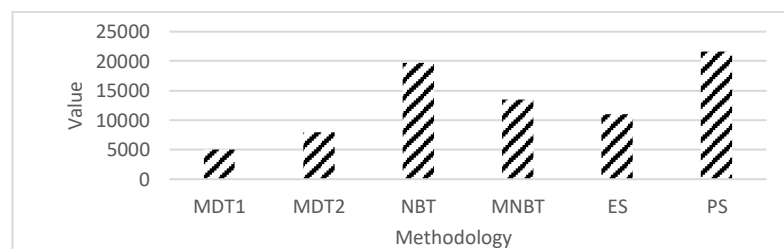


Figure 10. Speed-up comparison of SUSY dataset

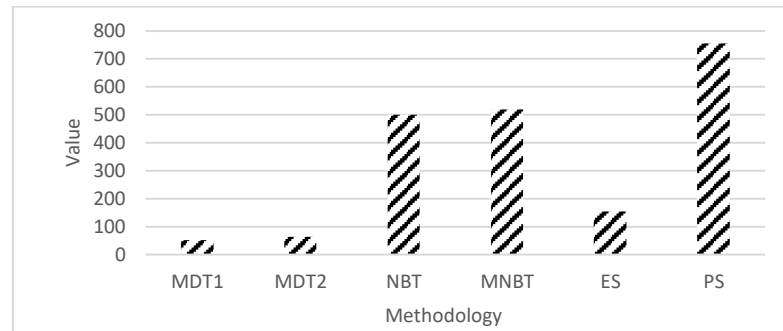


Figure 11. Speed up comparison for USPS dataset

5. CONCLUSION

This paper presents a comprehensive framework, the SDELF-BDC, which stands as a significant advancement in the field of big data classification. The framework's innovative approach to feature selection, rooted in a Hadoop-based map-reduce architecture, facilitates the handling of extensive datasets with increased efficiency and reduced computational overhead. The strategic integration of multiple deep learning classifiers harnesses the strengths of each, forming a robust ensemble that delivers enhanced predictive accuracy. Through meticulous design and execution, the SDELF-BDC framework emerges as a versatile and powerful tool in the big data arena, capable of addressing the complex and dynamic challenges posed by vast datasets. The framework's effectiveness, verified through extensive testing and analysis, validates the potential of deep ensemble learning in transforming big data classification methodologies. Future research could explore the scalability of the framework, the integration with emerging technologies, and its adaptability to other complex data-driven tasks beyond a particular domain. The SDELF-BDC framework sets the stage for the next generation of data analysis tools, marking a paradigm shift in big data classification and analytics.

REFERENCES

- [1] Z. Cai, J. Wang, and M. Ma, "The performance evaluation of big data-driven modulation classification in complex environment," *IEEE Access*, vol. 9, pp. 26313–26322, 2021, doi: 10.1109/ACCESS.2021.3054756.
- [2] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A big data-enabled hierarchical framework for traffic classification," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2608–2619, Oct. 2020, doi: 10.1109/TNSE.2020.3009832.
- [3] K. L.-M. Ang, F. L. Ge, and K. P. Seng, "Big educational data & analytics: survey, architecture and challenges," *IEEE Access*, vol. 8, pp. 116392–116414, 2020, doi: 10.1109/ACCESS.2020.2994561.
- [4] M. Zhu and Q. Chen, "Big data image classification based on distributed deep representation learning model," *IEEE Access*, vol. 8, pp. 133890–133904, 2020, doi: 10.1109/ACCESS.2020.3011127.
- [5] X. Li, F. Chen, R. Ruiz, and J. Zhu, "MapReduce task scheduling in heterogeneous geo-distributed data centers," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3317–3329, Nov. 2022, doi: 10.1109/TSC.2021.3092563.
- [6] M. Elkano, J. A. Sanz, E. Barrenechea, H. Bustince, and M. Galar, "CFM-BD: a distributed rule induction algorithm for building compact fuzzy models in big data classification problems," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 1, pp. 163–177, Jan. 2020, doi: 10.1109/TFUZZ.2019.2900856.
- [7] J. K. P. Seng and K. L.-M. Ang, "Multimodal emotion and sentiment modeling from unstructured big data: challenges, architecture, techniques," *IEEE Access*, vol. 7, pp. 90982–90998, 2019, doi: 10.1109/ACCESS.2019.2926751.
- [8] M. O. Ulfarsson, F. Pálsson, J. Sigurdsson, and J. R. Sveinsson, "Classification of big data with application to imaging genetics," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2137–2154, Nov. 2016, doi: 10.1109/JPROC.2015.2501814.
- [9] P. Pujar, A. Kumar, and V. Kumar, "Efficient plant leaf detection through machine learning approach based on corn leaf image classification," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, 2024, doi: 10.11591/ijai.v13.i1.pp1139-1148.
- [10] S. H. Sreedhara, V. Kumar, and S. Salma, "Efficient big data clustering using adhoc fuzzy c means and auto-encoder CNN," *Inventive Computation and Information Technologies*, pp. 353–368, 2023, doi: 10.1007/978-981-19-7402-1_25.
- [11] W. Xing and Y. Bei, "Medical health big data classification based on KNN classification algorithm," *IEEE Access*, vol. 8, pp. 28808–28819, 2020, doi: 10.1109/ACCESS.2019.2955754.
- [12] I. Goyal, A. Singh, and J. K. Saini, "Big data in healthcare: a review," in *2022 1st International Conference on Informatics (ICI)*, Apr. 2022, pp. 232–234, doi: 10.1109/ICI53355.2022.9786918.
- [13] M. Dener, S. Al, and G. Ok, "RFSE-GRU: data balanced classification model for mobile encrypted traffic in big data environment," *IEEE Access*, vol. 11, pp. 21831–21847, 2023, doi: 10.1109/ACCESS.2023.3251745.
- [14] L.-H. Yang, J. Liu, Y.-M. Wang, and L. Martinez, "A micro-extended belief rule-based system for big data multiclass classification problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 420–440, Jan. 2021, doi: 10.1109/TSMC.2018.2872843.
- [15] S. Alghunaim and H. H. Al-Baity, "On the scalability of machine-learning algorithms for breast cancer prediction in big data context," *IEEE Access*, vol. 7, pp. 91535–91546, 2019, doi: 10.1109/ACCESS.2019.2927080.
- [16] A. B. Hassanat *et al.*, "Magnetic force classifier: a novel method for big data classification," *IEEE Access*, vol. 10, pp. 12592–12606, 2022, doi: 10.1109/ACCESS.2022.3142888.
- [17] C. Jiang and Y. Li, "Health big data classification using improved radial basis function neural network and nearest neighbor propagation algorithm," *IEEE Access*, vol. 7, pp. 176782–176789, 2019, doi: 10.1109/ACCESS.2019.2956751.




- [18] K. Zheng, H. Wang, F. Qin, C. Miao, and Z. Han, "An improved land use classification method based on DeepLab V3+ under GauGAN data enhancement," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 5526–5537, 2023, doi: 10.1109/JSTARS.2023.3278862.
- [19] G. Zhai, Y. Yang, H. Wang, and S. Du, "Multi-attention fusion modeling for sentiment analysis of educational big data," *Big Data Mining and Analytics*, vol. 3, no. 4, pp. 311–319, Dec. 2020, doi: 10.26599/BDMA.2020.9020024.
- [20] A. Molinari and G. Nollo, "The quality concerns in health care big data," in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, Jun. 2020, pp. 302–305, doi: 10.1109/MELECON48756.2020.9140534.
- [21] H. Baer, V. Barger, X. Tata, and K. Zhang, "Detecting heavy neutral SUSY higgs bosons decaying to sparticles at the high-luminosity LHC," *Symmetry*, vol. 15, no. 2, Feb. 2023, doi: 10.3390/sym15020548.
- [22] D. Whiteson, "HIGGS dataset," *UCI Machine Learning Repository*, 2014, doi: 10.24432/C5V312.
- [23] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov. 2012, doi: 10.1109/MSP.2012.2211477.
- [24] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, May 1994, doi: 10.1109/34.291440.
- [25] F. Wang, Q. Wang, F. Nie, W. Yu, and R. Wang, "Efficient tree classifiers for large scale datasets," *Neurocomputing*, vol. 284, pp. 70–79, Apr. 2018, doi: 10.1016/j.neucom.2017.12.061.
- [26] A. B. A. Hassanat, "Furthest-pair-based binary search tree for speeding big data classification using k-nearest neighbors," *Big Data*, vol. 6, no. 3, pp. 225–235, Sep. 2018, doi: 10.1089/big.2018.0064.
- [27] J. Maillo, I. Triguero, and F. Herrera, "A MapReduce-based k-nearest neighbor approach for big data classification," in *2015 IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 167–172, doi: 10.1109/Trustcom.2015.577.
- [28] J. Maillo, S. Ramirez, I. Triguero, and F. Herrera, "kNN-IS: an iterative spark-based design of the k-nearest neighbors classifier for big data," *Knowledge-Based Systems*, vol. 117, pp. 3–15, Feb. 2017, doi: 10.1016/j.knosys.2016.06.012.
- [29] J. Maillo, J. Luengo, S. Garcia, F. Herrera, and I. Triguero, "Exact fuzzy k-nearest neighbor classification for big datasets," in *2017 IEEE International Conference on Fuzzy Systems*, Jul. 2017, pp. 1–6, doi: 10.1109/FUZZ-IEEE.2017.8015686.
- [30] A. B. A. Hassanat, "Norm-based binary search trees for speeding up KNN big data classification," *Computers*, vol. 7, no. 4, Oct. 2018, doi: 10.3390/computers7040054.
- [31] A. S. Tarawneh, E. S. Alamri, N. N. Al-Saedi, M. Alauthman, and A. B. Hassanat, "CTELC: a constant-time ensemble learning classifier based on KNN for big data," *IEEE Access*, vol. 11, pp. 89791–89802, 2023, doi: 10.1109/ACCESS.2023.3307512.

BIOGRAPHIES OF AUTHORS






Prof. Kesavan Mettur Varadharajan    holds a M.E. degree in computer science and engineering from Sona College of Technology, Salem and B.E. degree in electronics & instrumentation engineering from Kongu Engineering College, Erode. He is pursuing Ph.D. from Visveswaraya Technological University, Belagavi. He has worked at M. V. J. College of Engineering and currently working as Assistant Professor in East Point College of Engineering and Technology. His area of specialization is big data analytics, machine learning, and deep learning. He can be contacted at email: kes_mv@rediffmail.com.



Dr. Josephine Prem Kumar    received her B.Tech. degree in electronics and communication engineering and M.Tech. degree in computer science from Regional Engineering College (now National Institute of Technology), Warangal and Ph.D. in computer science and engineering from Dr. MGR Educational and Research Institute, Dr. MGR University, Chennai. After serving ITI Limited, Bangalore for over fifteen years and Infycons Creative Software Private Ltd., Bangalore for a brief period, she has taken up the teaching profession. She has worked as Professor in MVJ College of Engineering, Bangalore and East Point Group of Institutions and is currently working as Professor-CSE in Cambridge Institute of Technology, Bangalore. She has been guiding Ph.D. students under Visveswaraya Technological University. She can be contacted at email: josephine.cse@cambridge.edu.in.



Dr. Nanda Ashwin    working as HOD, Department CSE-(IoT & CSBT) East Point College of Engineering has about 25 years of teaching experience. She received his B.E. degree in computer science and engineering M.S. in system software and M.Tech. degree in software engineering with distinction from VTU, Belagavi. She has published 16 research papers in refereed international journals and 10 research papers in the proceedings of various international conferences. She has received several best paper awards for his research papers at various international conferences. Her areas of research include wireless communication, cloud computing, big data analytics, and data science. She can be contacted at email: nandaashwin@eastpoint.ac.in.