Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 4, August 2025, pp. 3061~3074
ISSN: 2302-9285, DOI: 10.11591/eei.v14i4.9595

HBRFE: an enhanced recursive feature elimination model for big data classification

Kesavan Mettur Varadharajan[1], Josephine Prem Kumar[2], Nanda Ashwin[1]
[1]Department of Computer Science and Engineering, East Point College of Engineering and Technology, Bengaluru, India
[2]Department of Computer Science and Engineering, Cambridge Institute of Technology, Bengaluru, India

Article Info

Article history:

Received Nov 22, 2024
Revised Jun 30, 2025
Accepted Jul 5, 2025

Keywords:

Big data
Classification
Ensemble learning
Feature selection
Hadoop framework
Recursive feature elimination

ABSTRACT

The process of classification in big data is a tedious task due to the large number of volumes, veracity, and variety of the data. Classification of big data pave the path to organize the data and improve the classifier performance. This research article proposed a Hadoop framework based recursive feature elimination-based model called HBFRE for extract significant features from the big data by integrating map and reduce frame work. HBFRE extract the significant features by removing the least and irrelevant features from the dataset by using refined recursive feature elimination (RFE) with map and reduce framework. This method takes the mean of each attribute and find the variance in each instance. The proposed model is evaluated and analyzed by the accuracy performance and time complexity. This research utilized various classifier like artificial neural network (ANN), support vector machine (SVM), random forest (RF), k-nearest neighbors (KNN), and AdaBoost to measure the classification performance on the big data. Proposed HBRFE model is compared with different feature selection like RFE, relief, backwards feature elimination, maximum relevance k-nearest neighbors (MR-KNN), and scalable deep ensemble framework big data classification (SDELF-BDC).

Corresponding Author:

Kesavan Mettur Varadharajan
Department of Computer Science and Engineering, East Point College of Engineering and Technology
Bengaluru, Karnataka 560049, India
Email: kesavan.mv@gmail.com

1. INTRODUCTION

Several researchers have been rigorously investigating and evaluating methodologies and resources pertaining to big data for an extensive period. The substantial volume of information disseminated and archived by social media participants, healthcare institutions, educational establishments, and other organizations is the impetus behind this scholarly interest. In recent years, big data has surfaced as the predominant term of choice within the digital competitive landscape for both researchers and practitioners. Big data represents a crucial asset that numerous executives across diverse industries are keen to leverage in order to derive rapid insights and enhance profitability [1]. The advent of big data and analytics began when numerous organizations realized that the volume of data, they were handling outstripped their processes, capacities, structures, technology infrastructure, and governance. They found it difficult to meet the demands of assessing the vast amount of diverse data [2]. Big data classification is turning into a crucial task in many different industries, including marketing, social media, and biology. The amount of data we must manage is growing uncontrollably because of recent improvements in data collection in several of these disciplines. Big data's large quantities, diversity, and complexity could make it more difficult to analyze and extract insights from it. In this case, traditional big data classification models must be modified or reworked to handle this

data [3]. Big data often comes with high dimensionality, meaning there are many features to consider during classification. Handling high-dimensional data efficiently without overfitting or losing important information is a significant challenge. The notable problem in big data is classifying big data. Accurate classification depends on selecting the most pertinent features from a vast set of available features [4]. The goal of feature selection is to find informative features and eliminate redundant or unnecessary ones. Identifying big data analysis provides methods for managing large amounts of data, storing it, making quick automated choices, and reducing the errors associated with human predictions. One such tool is the Hadoop distributed file system (HDFS). The HDFS is acknowledged as the most popular dataset tool. It is made to handle many big data kinds, including unstructured, semi-structured, and structured, and it enables distributed architectural systems, parallel processing, redundancy, and scalability [5]. As the landscape of big data continues to evolve, integrating advanced methodologies such as parallel processing becomes increasingly vital for enhancing feature elimination techniques. The implementation of distributed frameworks like MapReduce allows for the efficient handling of massive datasets by breaking down tasks into smaller, manageable chunks that can be processed simultaneously across a cluster of machines [6] furthermore, big data analysis offers tremendous potential for employing HDFS [7] tools and Hadoop technology to address various information security issues. The data value that is produced at the analysis stage of big data is extremely significant.

To the best of our knowledge, the amount of time needed for training and/or testing popular big data categorization techniques varies with the size of the big data set due to the presence of large number of redundant and irrelevant features. It would be tempting to create a new classifier that takes a fixed amount of time, regardless of the size of the big dataset, given the enormous volume and variety of big data. The implementation of a Hadoop-based framework is expected to enhance the efficiency of feature selection processes. By leveraging parallel processing capabilities, the framework can manage large datasets more effectively, leading to faster feature selection times. The research question arises when dealing in big data classification would a smaller number of significant features is sufficient to train the dataset for classification tasks. This manuscript endeavours to address the inquiry by presenting a Hadoop framework based recursive feature elimination (HBRFE) model aimed at eliminating redundant and minimally significant features from the dataset. The primary aim of the research endeavor is delineated as:

- To formulate a dynamic model for feature selection utilizing Hadoop technology to eliminate the least significant, redundant, and irrelevant features from large datasets, thereby enhancing the efficacy of classification performance.
- To extract significant feature from big data and create a constant number of feature subset with less computation time.
- Combining MapReduce and Hadoop framework with pre-defined threshold decision help to improve the efficiency of HBRFE algorithm when dealt with large dimension datasets.
- The proposed framework allows for the simultaneous execution of multiple feature selection algorithms. This feature enables to conduct direct feature extraction from the dataset with different selectors, helping to identify which algorithms perform best under various conditions and datasets.

2. LITERATURE REVIEW

Big data classification is significant because it can draw useful conclusions, trends, and information from large, intricate datasets. Big data classification is essential for turning unstructured data into actionable insights, facilitating well-informed decision-making, increasing productivity, and opening fresh doors for development and innovation in a variety of fields [8]. Recent research has developed novel techniques to improve feature selection and classification accuracy in large datasets within the field of big data classification. For instance, a cutting-edge method combines feature subset selection and hyper parameter-tuned deep belief networks with MapReduce to tackle the challenges of huge data processing and enhance classification performance [9]. A classification model called random forest-based feature selection (RFSE)-gated recurrent units (GRU) was created by combining a data balance and feature selection strategy with GRU. The random forest (RF) technique is used by this model to determine which features have the greatest influence on categorization. It uses a mix of the edited nearest neighbor (ENN) technique [10] for under sampling and the synthetic minority oversampling technique (SMOTE) for oversampling to reduce the difficulties caused by data imbalance and improve the classification accuracy of the model. Big data analytics includes a variety of technologies, including text and data mining, online and mobile mining, process mining, statistical analysis, network analytics, social media analytics, audio and video analytics, and web analytics [11]. Big data feature selection poses special difficulties because of the volume and high dimensionality of the data. Feature selection can be classified into filter method, embedded method, and wrapper method. The importance of features is assessed using filter methods other from the classifier. Mutual information [11], correlation analysis, and statistical tests such as ANOVA are common methods. Although these techniques

are effective in terms of computation and fit for sizable datasets, they might not consider feature interactions. Embedded techniques include feature selection in the process of creating the model [12]. During training, ensemble techniques such as gradient boosting and RF, as well as decision trees (DT), automatically choose features according to their significance. Large datasets can be handled by these techniques with ease; however, they might not necessarily yield the optimum feature subset. To assess feature subsets, wrapper approaches train and test a classifier on various feature subsets [13]. This group includes methods such as recursive feature elimination (RFE), forward selection, and backward elimination. Although they can capture feature interactions, wrapper approaches can be computationally costly for large datasets [14].

A binary search tree-based classification model [15] called feature selection using binary search tree-based partitioning and subset tree (FBPST) has proposed to handle big data classification problem efficiently [16]. FBPST is evolved from the concept of binary search tree construction to speed up the classification process. The goal of the furthest pair issue is to determine which two points in a set are the furthest apart. This issue can occur in several settings, including network design and computational geometry. Combining these ideas, the term "furthest-pair-based binary search tree" may be used to describe a specific binary search tree structure or technique intended to effectively resolve issues pertaining to locating the furthest pairs inside the data that the tree represents. The data examples that are, the examples with the maximum distance in the dataset are added into the tree based on how far away they are from the furthest pair. Approximately twenty datasets were used to test the approach, and the classification results were good. This method's primary flaw [17] is that it takes a long time to generate the models on the Higgs dataset with 11 million records, which it took almost 50 minutes. However, during the testing phase, the model takes a logarithmic amount of time if we ignore the model construction time. The training component within the FPBST generates a BST, thus accelerates the process, especially when classifying large data sets in comparison to the unsatisfactory amount of time the k-nearest neighbors (KNN) [18] classifier takes to find a test sample. The instances that are more like P2 than P1 are ordered to the right of the same host node in the same higher level, while the examples that are more like P1 are sorted to the left of their host node in a higher level [19], [20].

According to Garg *et al*. [21], norm-based binary tree and "minimum/maximum norms-based binary tree" are proposed to store the data in tree structure to speed up the classification process. This method probably describes a structure for a binary tree in which the nodes are arranged according to either their minimum or maximum norms. The data are stored recursively in binary search tree norms. The arrangement of nodes is based on which vector in each node's subtree has the smallest norm. This might be helpful in applications like closest neighbor searches when it is crucial to minimize the distance between nodes or vectors. In minimum the arrangement of nodes is based on which vector in each node's subtree has the biggest norm. This can be helpful in applications like clustering algorithms or outlier detection when it is crucial to maximize the distance between nodes or vectors.

A methodology referred to as fuzzy-KNN represents an enhancement of the traditional KNN algorithm that incorporates principles of fuzzy logic. The procedure consists of two distinct phases. The initial phase involves transformation, during which the training dataset is augmented to incorporate the degrees of class membership. Subsequently, the classification is executed in the second phase by utilizing the class membership information pertinent to the test subset. They used Poker Hand, supersymmetry (SUSY), and Higgs, three well-known large datasets, to test their approach. Their reported findings demonstrated the effectiveness of their approach. On the other hand, the Higgs dataset required about three days to run, which is a lot longer than the approximate algorithms required run times to obtain almost identical accuracy. MR-KNN, or MapReduce-based KNN, is a method that effectively uses the MapReduce programming model to carry out KNN searches on big datasets [22]. A programming concept and related implementation called MapReduce is used to process and generate massive datasets in parallel over a distributed cluster [23]. The incoming data is separated into smaller groups known as splits, usually expressed as key-value pairs. A map function processes each split separately, extracting pertinent information, and producing intermediate key-value pairs. In KNN, a data point's identifier could be represented by the key, and the feature vector by the value. The map function shuffles and sorts the intermediate key-value pairs by key throughout the cluster. The MR-KNN algorithm aims to cluster data points that may be the closest neighbors. MR-KNN's scalable and parallelizable approach to k-nearest neighbor search on large datasets. MR-KNN can be benefit in different applications including in data mining, machine learning, and information system. However, except for scenarios involving massive features, most of the time a linear speed increase is realized. As previously said, the maximum number of concurrently performing map tasks is surpassed in this scenario. Additionally, certain superliner speed increases that can be mistaken for sequential version memory-consumption issues. Mehta *et al*. [24], developed a dynamic big data classification model called SDELF-BDC. The proposed method incorporated the core concept of map and reduce framework to select the significant features. The suggested map reduction model, selects pertinent features from the map phase that it generates to minimize the length of each feature and obtain many significant features based on Hadoop framework [24]. The

suggested algorithm's parameters Ru1 and Ru2 are calculated using two different optimization approaches and measures the variance between two different optimization model. After that, a deep ensemble model that makes use of several deep learning classifiers processes the selected features. By using clustering to expedite the search for nearest neighbors without sacrificing classification accuracy, this method improves denoising capabilities and boosts the KNN method's performance. This proposed model performed well in classifying big dataset like Higgs [24], SUSY [25], modified national institute of standards and technology (MNIST) [26] and united states postal service (USPS) [27]. According to the published results, the technique took an average of roughly 5K seconds to process a dataset like Higgs, 6K seconds for SUSY, and 7K seconds and 8K seconds respectively, for MNIST and USPS resulting expensive time.

In machine learning, RFE is a feature selection method [28] that is frequently used to find the most significant features in a dataset for predictive modeling. RFE seeks to determine the feature subset that produces the greatest model performance [29] by repeatedly eliminating the least significant features. It is especially helpful in big datasets when there are many more features than samples because it reduces the chance of overfitting and enhances the interpretability of the model [30]. The general overview of RFE is expressed in Figure 1. Depict from the Figure 1, the process of getting data from a database or data source is called data fetching. Applications must be able to retrieve the required data quickly for them to carry out their intended tasks. This requires efficient data fetching. Preparing unprocessed data for analysis or modeling is referred to as pre-processing. It is an essential step in pipelines for machine learning and data analysis since it enhances the usefulness and quality of the data. Data cleansing, handling missing values, data normalization, data splitting, data transformation, and other activities are all included in pre-processing. Initial model training is a process that a model is first trained on the complete collection of features in RFE.

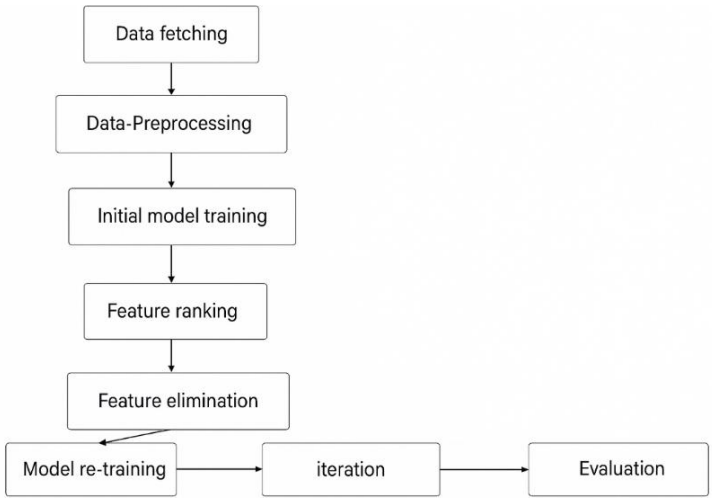

Figure 1. Workflow of RFE

The significance of every characteristic is established in feature ranking. For linear models, the importance of feature ranking is often depending on coefficients of feature ranking or tree-based models. Feature elimination will aid to remove the least significant features from the dataset. Model re-training is a process of training the reduced dataset. Iteration is a repeated process from step 2-4 till a certain number of features has been attained or till performance measurements (such as accuracy or error) stop improving. Eventually, the model's performance with the chosen features is assessed using cross-validation or a set of validations. RFE is not so much a mathematical formula as it is an algorithmic procedure. Nonetheless, RFE is frequently expressed in terms of mathematical steps.

a. Let consider data set D contain N features with M instances.
b. Choose big data model F and the desired feature selected K.
c. Train the model on entire data: F(X)=y, where X→ M*N feature matrix, and y is a target output.
d. Estimate feature importance score x_i {where, i=1,2,3…. N}.
e. Identifying the least significant feature x_j.
f. Remove the least significant features from the dataset: $X^{|}=X/\{x_j\}$.
g. Repeat from step c-step f until the desired number of features is obtained {($|X^{|}$)=K}.
h. Evaluate the performance of the model on selected features through cross validation.

The important score x_i can be obtained using any approach unique to the machine learning model of choice, such as the coefficients of a linear model or the feature importance of a tree-based model.

3. PROPOSED METHOD

Classifying big data is a crucial task due to the enormous features presents in the dataset. A dynamic model is required to extract significant feature from the big data to improve the classification task efficiently. The first stage of the proposed method is called RFE, in which relevant attributes are found using a map-reduce framework in order to minimize big datasets. Subsequently, a distinct classification model employing a range of classifiers, including KNN, support vector machine (SVM), DT, artificial neural network (ANN), RF, and AdaBoost, to process the selected features. These classifiers are used conditionally, according to how well they fit the features of the data. The implementation of a Hadoop-based framework is expected to enhance the efficiency of feature selection processes. By leveraging parallel processing capabilities, the framework can manage large datasets more effectively, leading to faster feature selection times. The outcomes of each classifier are pooled in the classification and reduction phase, and an optimization-based ensemble approach is used to further refine the results. With this combination, the goal of the proposed model by integrating MapReduce architecture is to produce a decisive strategy that maximizes the prediction metrics. The work flow of the proposed model is divided into multiple phases as defined in Figure 2.

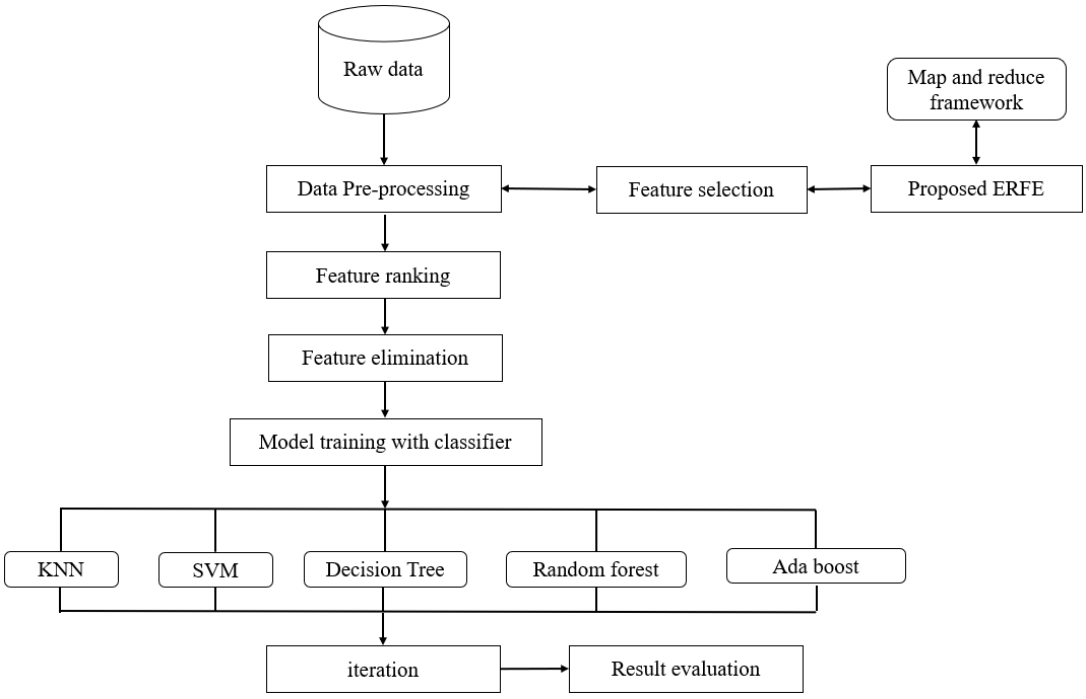

Figure 2. Proposed method workflow

The definition of the workflow as follows:

- Raw data: the term "raw data" describes any unprocessed, unstructured, and disorganized information that has not been altered, examined, or interpreted. This type of data is the most fundamental and is frequently gathered directly from several observations. In addition to lacking context, raw data can have mistaken or inconsistencies which can scrutinized in pre-processing phase.
- Pre-processing: to ensure that the analyses and modelling efforts that follow produce accurate, dependable, and useful insights, pre-processing is essential to the preparation of raw data for analysis. However, many attributes in the dataset contained different mean and standard deviation value. We used Z-normalization [31] to scale the values. Missing at random methods were used to handle the missing data [32].
- Feature selection: important features were extracted from the dataset based on HBFRE score. The detailed description of the proposed method is described in section 3.1.

- Feature ranking: the aim of feature ranking is to rank the extracted features by HBRFE and identify the subset of key features that has the greatest impact on model performance or forecast accuracy.
- Feature elimination: least score and redundant feature will be eliminated from the feature subset.
- Model training: the performance of the proposed HBRFE will be evaluated by different classifiers like KNN, SVM, DT, RF, and AdaBoost.
- Iteration: selection of significant features will be stopped once the desired set of features are selected.
- Result evaluation: eventually, the results are proven the proposed is robust in big data classification.

3.1. Proposed Hadoop framework based recursive feature elimination

The proposed algorithm-HBRFE, combines the technique of RFE with distributed computing capabilities of Hadoop MapReduce framework. This integrated approach encourages effective and expandable feature selection by allocating the computation of feature importance across multiple nodes, consequently decreasing the computational overhead. The iterative algorithm-HBREF computes each feature contribution by applying statistical measures like mean and standard deviation which eliminates features. This guarantees the selection of only those features that are both very important and consistent which improves the classification performance in big data infrastructures.

The HBRFE algorithm starts with collecting training dataset and thoroughly examining the features in the datasets. Using MapReduce Paradigm, the algorithm distributed the computation of features scores. The Map Phase calculates the distinct feature contributions, while the Reduce Phase combines and refine these scores. The statistical parameters such as mean (μ) and standard deviation (σ) are evaluated for each features. The displayed high mean scores and low variance are considered as important and are retained for the features. This dual-criteria evaluation ensures the stability and reliability of selected features. Subsequently, the algorithm constructs a reduced feature subset, which is used to train the classification model. The testing phase also leverages only these selected features, ensuring enhanced accuracy and reduced complexity. The final classification output is generated by evaluating the trained model against the test data using the refined feature set. The detailed HBRFE algorithm with all metrics given as shown in Algorithm 1.

Algorithm 1. HBRFE

Input: Training data set, finding least score features
Output: Extracted significant features
for all features in dataset D **do**
  for all X in features f1,f2,f3…fn **do**
    calculate the feature score $X^{|}=X/\{x_j\}$
    m:map(X(f1), X(f2),X(f3)….X(f2) && r:reduce(y(g(f1,f2,f3,)f4,fn)]
    for all features f(m, r) **do**
      calculate mean: μ of {f1,f2,f3} in D_i {i=f1,f2,f3…fn}
standard deviation: σ in D_i {i=f1,f2,f3…fn} wrt 'μ '
    end for
    f(m, r)=|{x, y}| where, {x, y} positive feature score
      x→ array of future with mean score
      y→ array of future with minimum variance
  end for
    $F(X_i) = \frac{\sum_{i=1}^{n} D_i, f\{m,r\}}{|\mu,\sigma(D_i)|} + \mathrm{var}(X, X_i)$
    X_i→generated subset of significant features from the dataset
    var(f1,f2…fn) feature training to find min variance score
    $\mathrm{train}(X_i) \subset \mathrm{var}(X, X_i)$
end for
    $\mathrm{cls}[C_x]$= test $(X_i) \subset \mathrm{var}(X, X_i)$
    return $\mathrm{cls}[C_x]$
    $\mathrm{cls}[C_x]$ → classification phase splitting least significant features
return $F(X_i)$

The HBRFE is implemented with objective function $F(X_j)$, where X_j is a subset of significant feature that extracted from the data set D_i (where i=f1, f2, f3…. fn features). The mathematical formula can be expressed as (1):

$$F(X_i) = \frac{\sum_{i=1}^{n} D_i, f\{m,r\}}{|\mu,\sigma(D_i)|} + var(X, X_i) \quad (1)$$

where;
D_i=data set contains i-features where (i=f1, f2, f3…fn},
f{m,r}=function of m: map and r: reduce,
μ=mean value of each feature,
σ=standard deviation on each feature set wrt mean value,
X_i=generated significant feature subset.

$$var(X, X_i) = \sum_{i=1}^{n} (|X_i - X^{|}{}_i|)^{\frac{1}{\mu}}$$

Definition: the map function creates a new set of changed items by separately applying a given operation to each element in a collection of data elements. The map operation can be expressed mathematically as: map (X, [f1, f2, f3…fn])=[X (f1), X (f2), X (f3) ….X (f2) where X is the function applied on each individual features. The reduction function applies a specified binary operation repeatedly to a collection of data items, combining them into a single result. It works by "reducing" each item in the list one at a time to a single value. The reduction operation can be expressed mathematically as reduction functions''→**reduce** (y ([f1, f2, f3, f4…. fn])=[g (f1, f2, f3,) f4, fn)] where y is the binary operation used to group two different features.

3.2. Parameter

The aim of HBRFE is to select the significant features by removing least significant score features. The objective function $F(X_i)$, resulting to select significant feature by integrating RFE and map and reduce framework. Parameter map(x[F]) is designed to map all the input data features into key and value pairs. Key represent the symbolic address of the feature variable and value to store the actual of the feature variable. On each of these input key-value pairs, the map () function will run in its internal memory warehouse and produce a placeholder key-value pair that serves as an input parameter for the reducer or reduce () function. The linear combination of input features f1, f2, f3…. fn and their corresponding difference of D_i, *f(m,r)* with the addition value of variance of the selected features X and mean X_i. The reduce () function receives the intermediate key-value pairs that are sorted and shuffled before being used as input to the reduce (). *cls[*C_x*]*, classification phase to remove least significant feature by default parameter 'none', feature which an integer the default parameter none are considered absolute number of features. Whereas, if float value 0 to 1 are considered the fraction score of features. $var(X, X_i)$ measure the feature score variance with mean value of each attribute. $F(X_i)$ control and store the optimal significant features as output and overrides the default feature importance score.

4. RESULTS

4.1. Dataset description

The evaluation of performance is conducted using four datasets. Higgs, MNIST, SUSY, and USPS. In addition to comparing each methodology with the suggested model and evaluating correctness for each dataset, we also compare time and speed for the Susy and Higgs datasets. The comparison results of the proposed method are visualized by chart and tables.

4.2. Performance evaluation

4.2.1. Accuracy comparison

We examine how different approaches perform on the Higgs dataset and find that these classification algorithms produce a variety of accuracy scores. The ultimate goal of the HBRFE framework is to identify the most effective feature selector and the relevant features to select from large datasets. This outcome is crucial for enhancing the performance of machine learning models, as selecting the right features can greatly impact model accuracy and efficiency. With an accuracy score of 90%, HBFRE outperforms most conventional algorithms, indicating that its classification approach is especially well-suited to this dataset. Still, with scores of 88% and 87%, SDELF-BDC and MR-KNN also marginally outperform. The FPBST and fuzzy-KNN approach, achieved a dependable accuracy of 84%, which may indicate a sophisticated feature selection or optimization procedure that greatly enhances the model's performance on this dataset in future. Conversely, MNBT has the lowest score (83%), which might mean that the patterns in the Higgs data do not work as well for its probabilistic technique. All the results are tabulated in Table 1 and visualized in Figure 3.

Table 1. Accuracy comparison on Higgs dataset

Different classification accuracy comparison on Higgs dataset	KNN	SVM	ANN	DT	RF	AdaBoost
FPBST	84	80	78	75	86	89
MNBT	83	79	79	79	83	94
MR-KNN	87	82	77	88	84	91
SDELF-BDC	88	81	81	86	79	90
Fuzzy-KNN	84	88	87	88	81	93
Proposed HBFRE	90	89	88	87	88	92

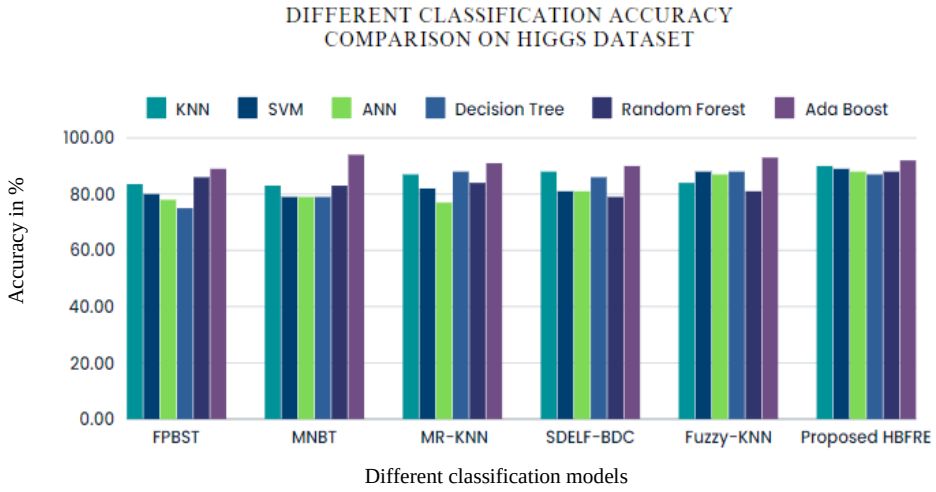

Figure 3. Accuracy comparison on Higgs dataset

The suggested method performs significantly better than other models in the analysis of classification methods SVM, KNN, ANN, and RF on the SUSY dataset, with accuracy of 89%, 90%, 88%, and 88%. The integration of a binary associative-memory MapReduce within the Hadoop framework provides flexibility in adapting to various types of datasets and feature selection needs. This adaptability is essential for while working with diverse data sources. This suggests that the advanced approach, which may involve complex feature selection or optimization techniques, is highly appropriate for this kind of complex data. Nevertheless, MR-KNN outperformed with 88% accuracy on MNBT by AdaBoost classifier and DT classifier. SDELF-BDC obtained good accuracy than MNBT, FBPST, and fuzzy-KNN with KNN classifier. Fuzzy-KNN performed better than SDELF-BDC, MR-KNN, MNBT and FPBST with SVM classifier. MR-KNN achieved great classification accuracy on SUSY dataset with DT classifier and MNBT achieved 94% i.e., average 4% higher than all the other models with AdaBoost shown in Table 2 and Figure 4. Eventually, all the models performed competitively better with each other's. Comparatively HBRFE achieved better classification performance on the data than other models.

Table 2. Accuracy comparison on SUSY dataset

Different classification accuracy comparison on SUSY dataset Higgs	KNN	SVM	ANN	DT	RF	AdaBoost
FPBST	84	80	78	75	86	89
MNBT	83	79	79	79	83	94
MR-KNN	87	82	77	88	84	91
SDELF-BDC	88	81	81	86	79	90
Fuzzy-KNN	84	88	87	88	81	93
Proposed HBFRE	90	89	88	87	88	92

The classification performance on MNIST dataset have been conducted with various classifier. The classification result shows that the proposed model out performs using SVM, DT, RF, and AdaBoost. Using SVM classifier model obtained nearly 85% accuracy which is 2% higher than fuzzy-KNN and 6% higher than FPBST and 9%, 5%, higher than MNBT and MR-KNN. The experimental results highlight the effectiveness of the RF algorithm and AdaBoost within the Hadoop MapReduce framework, the benefits of

incremental processing, and the importance of energy efficiency and scheduling in optimizing performance. These findings contribute valuable insights into improving classification process on large clusters [33]. SDELF-BDC achieved 2% higher than HBRFE with KNN and MNBT obtained 0.5%. However, the overall performance of the proposed is better on SVM, DT, RF, and AdaBoost. The comparison result is shown in detail in Table 3 and Figure 4. The results also discuss the implementation of priority-based scheduling, which allocates classification process based on task requirements and utilization. This parameter map(x[F]) and reduce () approach contributes to the reduction of map tasks, further enhancing the system's energy efficiency and performance.

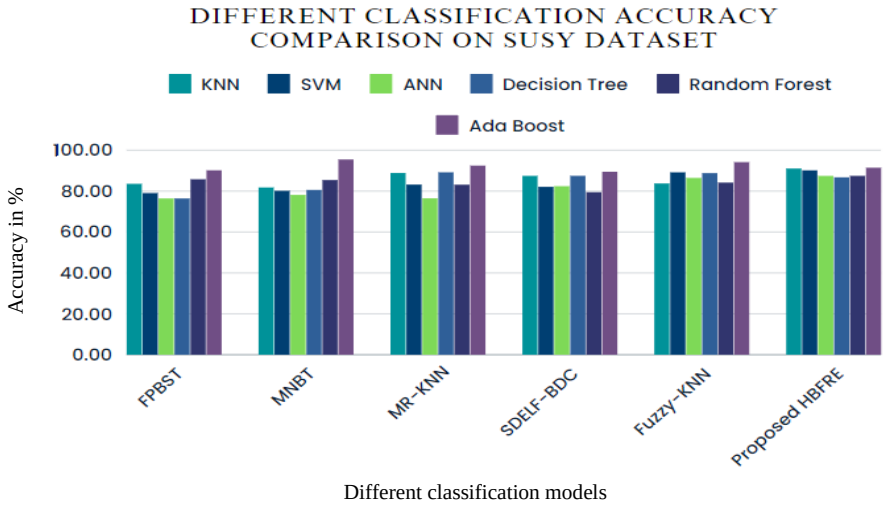

Figure 4. Accuracy comparison on SUSY dataset

Table 3. Accuracy comparison on MNIST dataset

Different classification accuracy comparison on MNIST dataset						
	KNN	SVM	ANN	DT	RF	AdaBoost
FPBST	74.56	79.46	77.49	81.47	82.59	81.89
MNBT	78.45	76.45	79.56	83.45	83.75	82.46
MR-KNN	80.75	80.19	78.79	84.79	84.49	81.79
SDELF-BDC	81.78	82.47	76.45	86.45	85.45	84.73
Fuzzy-KNN	79.45	83.45	78.49	84.46	84.79	81.47
Proposed HBFRE	80.45	84.57	79.09	87.89	86.19	84.84

The proposed method outperforms existing models in the examination of classification methods SVM, ANN, DT and AdaBoost on the USPS dataset. This implies that the advanced method is very suitable for this type of complicated data and may entail intricate feature selection or optimization procedures. However, RF performed at 91% accuracy, whereas SDELF-BDC performed at 88%. Using a KNN classifier, SDELF-BDC outperformed MNBT, FBPST, and fuzzy-KNN in terms of accuracy. MR-KNN outperformed with RF classifier in comparison to SDELF-BDC, MR-KNN, MNBT, and FPBST as shown in Table 4 and Figure 5.

Table 4. Accuracy comparison on USPS dataset

Different classification accuracy comparison on USPS dataset						
	KNN	SVM	ANN	DT	RF	AdaBoost
FPBST	89.45	90.14	92	88.74	87.88	64.48
MNBT	88.74	90.12	89.79	90.01	89.76	71.46
MR-KNN	86.47	90.46	92.56	91.46	91.45	66.59
SDELF-BDC	87.91	91.28	91	86.45	79.84	88.46
Fuzzy-KNN	87.81	94.45	88.46	84.23	77.48	79.46
Proposed HBFRE	86.47	95.79	94.43	92.45	84.23	86.43

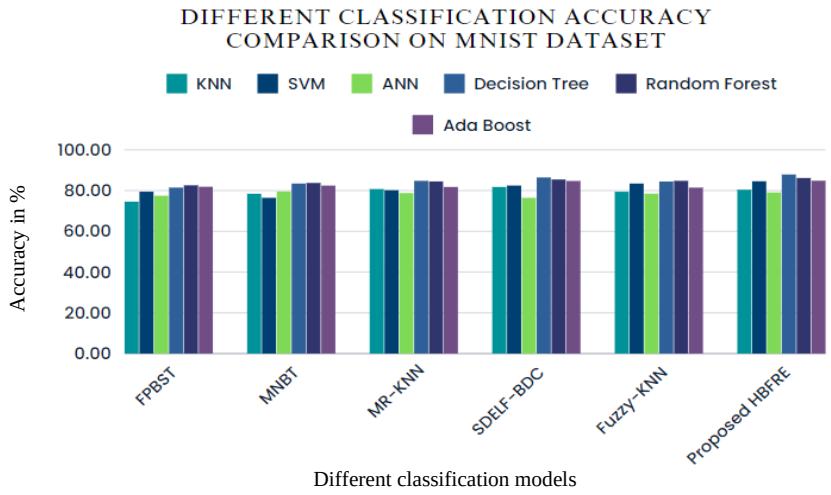

Figure 5. Accuracy comparison on MNIST datasets

SDELF-BDC achieved 2% higher than HBRFE with KNN and MNBT obtained 0.5% on USPS dataset. However, the overall performance of the proposed is better on SVM, DT, RF, and AdaBoost. The comparison result is shown in detail in Table 4 and Figure 6. The results also discuss the implementation of priority-based scheduling, which allocates classification process based on task requirements and utilization. This parameter map(x[F]) and reduce () approach contributes to the reduction of map tasks, further enhancing the system's energy efficiency and performance.

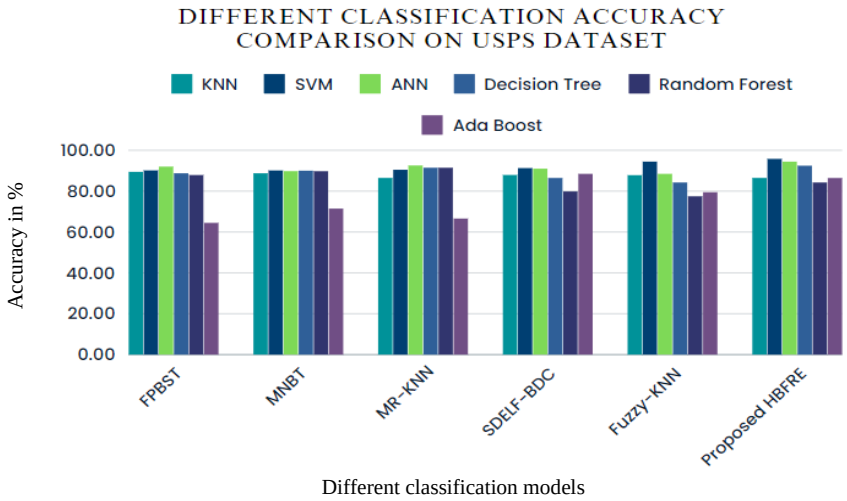

Figure 6. Accuracy comparison on USPS datasets

4.2.2. Computation runtime comparison

All algorithms are implemented in Python using the scikit-learn library for SVM, KNN, DT, RF, and TensorFlow backend for MLP and ANN. Experiments are run on a desktop with an Intel Core i5-7th generation with 8 GB of RAM. Since estimating execution time is a well-known problem, many strategies have already been put forth. The task is divided into its most fundamental component processes and elements using the analytical method. Standard timings are applied to these items if they are accessible from a data source. The execution times are estimated based on the dataset in the local machine. The execution time is varied based on the environment and system specification. All the mentioned and introduced models use past data to estimate execution time in seconds. This can be compared to the big O notation's algorithm complexity formulas. The computation time on Higgs dataset of FBPST is 10.6 hrs larger than other models. MNBT and MR-KNN both models achieved 9.1 hrs slightly better than FBPST. SDELF-BDC achieved

6.8 hrs and fuzzy-KNN is 6.4 hrs which is better than FPBST, MNBT, and MR-KNN. The proposed HBRFE achieved 5.2 hrs which is better than the other models. The computation time on SUSY dataset for MNBT is 7.53 hrs which is equal to 452 seconds maximum than other models. FPBST obtained 6.7 hrs slightly better than FPBST. MR-KNN is 6.61 hrs better than MNBT and FPBST. SDELF-BDC consumed 6.41 hrs far better than MR-KNN, FPBST, and MNBT. Fuzzy-KNN taken 6.8 hrs to perform the classification task which is better than MNBT but costlier than MR-KNN and SDELF-BDC. HBRFE attained 5.2 hrs equal to 323 second which much better than the other models [34]. In MNIST big data fuzzy-KNN consumed 13.31 hrs higher complexity than all the other models. FBPST taken 13.15 hrs which slightly slower than fuzzy-KNN. Eventually MNBT, MR-KNN, and SDELF-BDC are consumed 12.65 hrs, 12.8 hrs, and 12.48 hrs less time than FBPST and fuzzy-KNN. However proposed HBRFE attained least computation time then all the above models due to its 'none' and 'var. The highest computation time consumed on USPS dataset is fuzzy-KNN 5.25 hrs and SDELF-BDC is 5 hrs. Other models like [35] FBPST, MNBT, MR-KNN are consumed linear run time like 4.5 hrs, 4.08 hrs, and 4.81 hrs. However, the proposed HBRFE attained least computation complexity than another model as shown in Table 5 and visualized in Figure 7.

Table 5. Comparison of computation complexity

Computation time comparison in seconds				
	Higgs	SUSY	MNIST	USPS
FPBST	640	402	789	274
MNBT	548	452	759	245
MR-KNN	546	397	768	289
SDELF-BDC	412	385	749	300
Fuzzy-KNN	385	412	799	315
Proposed HBFRE	312	323	745	241

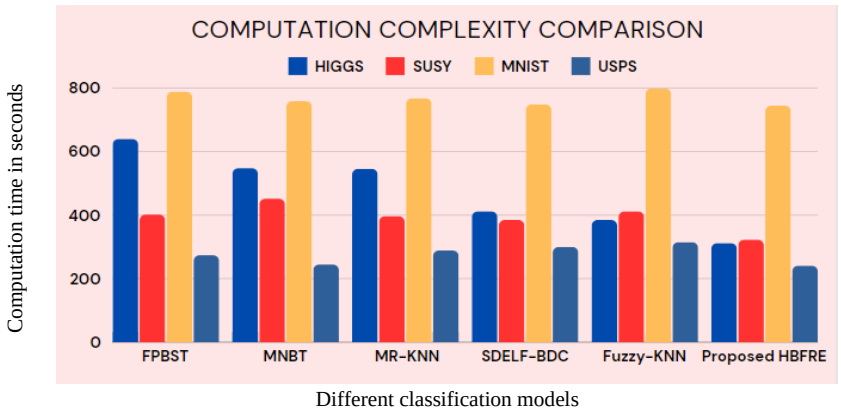

Figure 7. Comparison of computation complexity

5. CONCLUSION

The study emphasizes the significance of Hadoop framework and RFE model in large dataset classification. It highlights that traditional methods often overlook the features of unknown patterns, focusing primarily on known patterns. The proposed model addresses this gap by incorporating HBFRE which represents a breakthrough in the big data classification space. This model utilizes a MapReduce process for unknown patterns based on the KNN RF and various classifications models. This method allows for a more nuanced understanding of the feature significance, particularly when they overlap, leading to improved classification accuracy. By carefully combining the map and reduce functions, the strengths of each parameter specified in the objective function can be leveraged to create a strong feature selection model that improves classification accuracy. The paper discusses the creation of a matrix during the map and reduce process. This matrix plays a crucial role in representing the features of unknown patterns, thereby facilitating better classification outcomes. The proposed model demonstrates impressive classification accuracy, achieving 84.86% on a 50% training dataset and 89.35% on an 80% training dataset. This indicates that the model is not only effective but also efficient, as it learns well with a relatively small amount of training data. The results of the proposed model were compared with various other models, including the FPBST, MNBT, MR-KNN, SDELF-BDC, and fuzzy–KNN. The findings suggest that the new model outperforms these existing methods, showcasing its potential for practical applications in big data classification. The classifier's

ability to learn effectively from limited training data points to its efficiency and speed. This characteristic makes it a promising tool for real-world applications where data may be scarce. The paper concludes that the proposed HRFE model significantly enhances big data classification by effectively utilizing Hadoop and MapReduce, leading to high accuracy and efficiency in processing of significant feature selection from the big data.

ACKNOWLEDGMENTS

We would like to express my sincere gratitude to East Point College of Engineering and Technology, Bengaluru for providing the necessary infrastructure and academic support to carry out this research work.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kesavan Mettur Varadharajan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
Josephine Prem Kumar	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	
Nanda Ashwin		✓		✓		✓	✓		✓		✓	✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

The research related to human use has been complied with all the relevant national regulations and institutional policies in accordance with the tenets of the Helsinki Declaration and has been approved by the authors' institutional review board or equivalent committee.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [initials: KMV], upon reasonable request.

REFERENCES

[1] M. V. Kesavan, J. P. Kumar, and A. Manimegalai, "Survey on MapReduce Scheduler Algorithms in Hadoop Framework," *International Journal of Innovative Research in Information Security*, vol. 10, no. 4, pp. 314–319, May 2024, doi: 10.26562/ijiris.2024.v1004.38.

[2] R. Sahal, J. G. Breslin, and M. I. Ali, "Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case," *Journal of Manufacturing Systems*, vol. 54, pp. 138–151, Jan. 2020, doi: 10.1016/j.jmsy.2019.11.004.

[3] R. S. A. Daulay, S. Efendi, and Suherman, "Review of Literature on Improving the KNN Algorithm," *Transactions on Machine Learning and Artificial Intelligence*, vol. 11, no. 3, pp. 63–72, Jun. 2023, doi: 10.14738/tecs.113.14768.

[4] S. Hedayati, N. Maleki, T. Olsson, F. Ahlgren, M. Seyednezhad, and K. Berahmand, "MapReduce scheduling algorithms in Hadoop: a systematic study," *Journal of Cloud Computing*, vol. 12, no. 1, p. 143, Oct. 2023, doi: 10.1186/s13677-023-00520-9.

[5] 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.

[6] 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.

[7] D. M. D. Raj and R. Mohanasundaram, "An Efficient Filter-Based Feature Selection Model to Identify Significant Features from High-Dimensional Microarray Data," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2619–2630, Apr. 2020, doi: 10.1007/s13369-020-04380-2.

[8] M. S. Mahmood *et al.*, "Enhancing compressive strength prediction in self-compacting concrete using machine learning and deep learning techniques with incorporation of rice husk ash and marble powder," *Case Studies in Construction Materials*, vol. 19, p. e02557, Dec. 2023, doi: 10.1016/j.cscm.2023.e02557.

[9] A. Zafra and E. Gibaja, "Nearest neighbor-based approaches for multi-instance multi-label classification," *Expert Systems with Applications*, vol. 232, p. 120876, Dec. 2023, doi: 10.1016/j.eswa.2023.120876.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/bf00994018.

[11] B. Kamiński, M. Jakubczyk, and P. Szufel, "A framework for sensitivity analysis of decision trees," *Central European Journal of Operations Research*, vol. 26, no. 1, pp. 135–159, Mar. 2018, doi: 10.1007/s10100-017-0479-6.

[12] X. Wang, X. Wang, and M. Wilkes, "A Nearest Neighbor Classifier-Based Automated On-Line Novel Visual Percept Detection Method," in *New Developments in Unsupervised Outlier Detection*, Singapore: Springer Singapore, pp. 223–255, 2021, doi: 10.1007/978-981-15-9519-6_9.

[13] D. de Rigo, A. E. Rizzoli, R. S. Sessa, E. Weber, and P. Zenesi, "Neuro-dynamic programming for the efficient management of reservoir networks," *Proceedings of MODSIM 2001, International Congress on Modelling and Simulation*, vol. 4, pp. 1949–1954, 2001, doi: 10.5281/ZENODO.7481.

[14] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998, doi: 10.1109/34.709601.

[15] L. Chen, C. Wang, J. Chen, Z. Xiang, and X. Hu, "Voice Disorder Identification by using Hilbert-Huang Transform (HHT) and K Nearest Neighbor (KNN)," *Journal of Voice*, vol. 35, no. 6, pp. 932.e1-932.e11, Nov. 2021, doi: 10.1016/j.jvoice.2020.03.009.

[16] 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.

[17] D. M. D. Raj and R. Mohanasundaram, "An Efficient Filter-Based Feature Selection Model to Identify Significant Features from High-Dimensional Microarray Data," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2619–2630, Apr. 2020, doi: 10.1007/s13369-020-04380-2.

[18] A. B. A. Hassanat, "Furthest-pair-based decision trees: Experimental results on big data classification," *Information (Switzerland)*, vol. 9, no. 11, pp. 1-22, Nov. 2018, doi: 10.3390/info9110284.

[19] 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, 2018, doi: 10.1089/big.2018.0064.

[20] D. R. Munirathinam and M. Ranganadhan, "A new improved filter-based feature selection model for high-dimensional data," *Journal of Supercomputing*, vol. 76, no. 8, pp. 5745–5762, Aug. 2020, doi: 10.1007/s11227-019-02975-7.

[21] A. Garg, B. Sharma, and R. Khan, "Heart disease prediction using machine learning techniques," *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, pp. 1-10, Jan. 2021, doi: 10.1088/1757-899X/1022/1/012046.

[22] J. Maillo, I. Triguero, and F. Herrera, "A MapReduce-Based k-Nearest Neighbor Approach for Big Data Classification," in *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom*, Aug. 2015, vol. 2, pp. 167–172, doi: 10.1109/Trustcom.2015.577.

[23] A. Mehta, D. Goyal, A. Choudhary, B. S. Pabla, and S. Belghith, "Machine Learning-Based Fault Diagnosis of Self-Aligning Bearings for Rotating Machinery Using Infrared Thermography," *Mathematical Problems in Engineering*, vol. 2021, 2021, doi: 10.1155/2021/9947300.

[24] K. M. Varadharajan, J. P. Kumar, and N. Ashwin, "A novel scalable deep ensemble learning framework for Big Data Classification via MapReduce integration*" International Journal of Artificial Intelligence*, vol. 14, no 2, pp. 1386-1400, Apr. 2025, doi: 10.11591/ijai.v14.i2.pp1386-1400.

[25] 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, pp. 1-22, Feb. 2023, doi: 10.3390/sym15020548.

[26] D. M. D. Raj and R. Mohanasundaram, "An Efficient Filter-Based Feature Selection Model to Identify Significant Features from High-Dimensional Microarray Data," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2619–2630, Apr. 2020, doi: 10.1007/s13369-020-04380-2.

[27] D. Whiteson, "HIGGS-UCI Machine Learning Repository," *UC Irvine Machine Learning*, 2014.

[28] S. Li *et al.*, "Development and validation of nomograms predicting postoperative survival in patients with chromophobe renal cell carcinoma," *Frontiers in Oncology*, vol. 12, Nov. 2022, doi: 10.3389/fonc.2022.982833.

[29] C. Z. Huang *et al.*, "A Web-Based Prediction Model for Cancer-Specific Survival of Elderly Patients With Clear Cell Renal Cell Carcinoma: A Population-Based Study," *Frontiers in Public Health*, vol. 9, Mar. 2022, doi: 10.3389/fpubh.2021.833970.

[30] M. V. Kesavan, "Predicting Indian GDP with Machine Learning: A Comparison of Regression Models," *International Journal of Innovative Research in Information Security*, vol. 10, no. 04, pp. 150–155, May. 2024, doi: 10.26562/ijiris.2024.v1004.06.

[31] C. Gopalakrishnan and M. Iyapparaja, "Multilevel thresholding-based follicle detection and classification of polycystic ovary syndrome from the ultrasound images using machine learning," *International Journal of Systems Assurance Engineering and Management*, Aug. 2021, doi: 10.1007/s13198-021-01203-x.

[32] M. L. Piccinelli *et al.*, "Critical Appraisal of Leibovich 2018 and GRANT Models for Prediction of Cancer-Specific Survival in Non-Metastatic Chromophobe Renal Cell Carcinoma," *Cancers*, vol. 15, no. 7, pp. 1-13, Apr. 2023, doi: 10.3390/cancers15072155.

[33] A. Shah, S. Ahirrao, S. Pandya, K. Kotecha, and S. Rathod, "Smart Cardiac Framework for an Early Detection of Cardiac Arrest Condition and Risk," *Frontiers in Public Health*, vol. 9, Oct. 2021, doi: 10.3389/fpubh.2021.762303.

[34] C. Gopalakrishnan and M. Iyapparaja, "Active contour with modified Otsu method for automatic detection of polycystic ovary syndrome from ultrasound image of ovary," *Multimedia Tools and Applications*, vol. 79, no. 23–24, pp. 17169–17192, Jun. 2020, doi: 10.1007/s11042-019-07762-3.

[35] S. Suresh, T. R. Kumar, M. Nagalakshmi, J. B. Fernandes, and S. Kavitha, "Hadoop Map Reduce Techniques: Simplified Data Processing on Large Clusters with Data Mining," in *6th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2022 - Proceedings*, Nov. 2022, pp. 420–423, doi: 10.1109/I-SMAC55078.2022.9986501.

BIOGRAPHIES OF AUTHORS

Prof. Kesavan Mettur Varadharajan holds a M.E. degree in Computer Science and Engineering from Sona College of Technology, Salem, Tamil Nadu Anna University, Chennai. B.E. degree in Electronics and Instrumentation Engineering from Kongu Engineering College, Erode, Tamil Nadu Bharathiyar University, Coimbatore and Pursuing Ph.D. in Computer Science and Engineering from VTU University, Belagavi, Karnataka. His area of specialization is big data analytics, machine learning, and deep learning. He has worked at institutions such as MVJ College of Engineering, Bengaluru, Sakthi Institute of Technology, Tamil Nadu. A resourceful and dedicated Faculty in Department of CSE, with 21 years of experience in academic teaching and various administration roles for UG & PG Students and has published many publications in Peer reviewed and refereed Journals and conferences. He has handled Subjects such as big data analytics, data mining and data warehousing, analog and digital electronics, computer organization, advance computer architecture, operating system, microprocessors and microcontrollers, embedded system, programming in C++, Java, and J2EE. He can be contacted at email: kesavan.mv@gmail.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 (National Institute of Technology) Warangal. Ph.D. in Computer Science and Engineering from Dr. MGR Educational and Research Institute, Dr. MGR University, Chennai. She has completed NPTEL Domain certification in artificial intelligence. 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 Visvesvaraya Technological University. She can be contacted at email: d_prem_k@yahoo.com.

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 MS 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.