International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

IFSFSPIS: Incremental Feature Selection and Feature Sensitive Progressive Instance Selection Models for BigData Analytics

Subhash Kamble [1*], Arunalatha J. S.[2], Venugopal K. R.[3]

Submitted: 09/05/2023 **Revised**: 14/07/2023 **Accepted**: 08/08/2023

Abstract: In this paper, a highly robust and first-of-its-kind Incremental Feature Selection (IFS) framework is designed for BigData Analytics, which considers both feature reduction as well as feature sensitive instance selection as a viable solution towards BigData Analytics. Unlike classical threshold based feature selection methods, this work is designed for an IFS model encompassing Chi-Squared IFS with Feature Sensitive Progressive Instance Selection (FSPIS). This concept intends to meet Volume, Variety, Velocity, and Veracity aspects of the BigData simultaneously. FSPIS model executed K-Means clustering over the selected features and performed incremental stratified instance selection. The proposed FSPIS model initiated feature selection with minimum volume as 20% (and maximum as 80%), which was continuously updated by appending new (ranked) features and corresponding instances to achieve expected accuracy performance. To ensure generalizability of the solution, FSPIS model can be applied to an ensemble learning model encompassing Bagging, AdaBoost, k-NN, Random Forest and Extended-Tree Classifiers as foundational-classifiers to perform consensus-based classification. Simulation results over the different datasets confirmed that the proposed FSPIS model selects minimum features while retaining higher statistical performance (i.e., Accuracy), and minimum computational time than other state-of-art techniques.

Keywords: *BigData Analytics, Feature Sensitive Progressive Instance Selection, Incremental Feature Selection, Select-k-Best.*

1. Introduction

The high pace increase in advanced software technologies, internet, and affordable hardware solutions have expanded the opportunities for global human society to exploit aforesaid technologies for making efficient, and timely decisions. The above mentioned technologies have enabled the different decentralized computing environment to exploit the large set of input data to perform real time data mining, and decision making. Exponentially developing technologies, and allied up-surge in demands from the different socio-industrial verticals such as industrial communication, business intelligence and analytics, e-Healthcare, surveillance, civic administration, and allied query-driven data support, science and technology, social media, e-Commerce, and digital education, *etc.*, have given rise to a new computing world called BigData. BigData, often commonly defined using 4Vs stating Volume, Variety, Velocity, and Veracity demands a state-of-art robust computing environment to process significantly large data elements to identify the optimal set of cues for accurate decision making. Despite the roaring significance in the contemporary (analytics) world [1], [2], [3] BigData need to address numerous challenges including large heterogeneous data, unstructured data, and high-dimensional humongous data. BigData analytics possesses a large set of spatio-temporal features having impact on certain target cues or decision-making variables. However, processing a significantly large set of features over humongous (heterogeneous and high dimensional) inputs make major BigData analytics models confined, especially due to large features, convergence and local minima, and delay issues. Such limitations confine the efficacy of the BigData analytics solutions to meet Velocity and Veracity demands. To alleviate such issues, learning over most significant features can be of great significance. Moreover, the analytics model requires the ability to process humongous data swiftly and learn over the maximum possible, and significant features to make final prediction accurately [4], [5].

In sync with the above stated 4V challenges, the classical analytics solutions or data mining methods which are typically based on the full-batch-mode learning concept turn into confined solutions, and fail in addressing 4V expectations. This as a result makes analytics solutions inferior to contemporary real-time decision systems. The exploitation of the complete set of original data is visualized to be the key reason behind such limited performance [6], [7]. On the contrary, there can be a set of minimum features which could give the same or relatively similar performance even with significantly reduced data size. As a result, it can help not only alleviate the issues like local minima and convergence but can provide higher accuracy with low latency, and achieves Accuracy (i.e., Veracity), Time-efficient computation (i.e., Velocity), even with large (i.e., Voluminous) and high-dimensional *i.e.*, Variety) input data. With respect to BigData attributes, the dimensionality

[1* 2,3]*Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, Karnataka – 560001, India*
ORCID ID: 0009-0007-3108-535X
** Corresponding Author Email: subhash.kambli@gmail.com*

reduction methods can be of vital significance so as to minimize the dimensions of the original data, while enabling better learning efficiency [8].

BigData analytics typically consider analytics problems as three key tasks, clustering, regression and classification. In these processes, the primary objective of feature selection is to retain a subset of the most significant features to construct an optimal prediction model, by dropping irrelevant or less-significant features [9]. It can help in improving the performance of the prediction systems by minimizing the challenges of high dimensionality, performance generality, and accelerating the computation [2], [10]. It also helps in improving the interpretability of the model. BigData analytics employs different techniques like mining, machine learning etc., to perform feature learning, and allied decision making [11]. However, the classical data mining, and allied feature selection methods undergo limited performance, and demand more computational time and memory. On the contrary, in a real-world environment, the data volume keeps increasing over time, and hence makes most of the existing methods limited under dynamic data. Despite being explored extensively, the majority of the at-hand solutions are capable of learning data in batch-wise processing, and inculcates features selection as an offline process. Moreover, in this mechanism, the features of training instances are provided in advance. Unfortunately, such hypotheses may not be applicable to all the real-time BigData environments, where the input data is dynamically fed into the computing system, and the analytics model might have to do online feature learning to make decisions. Moreover, the majority of the existing feature selection methods merely focus on dimensional reduction while ignoring the presence of redundant instances in the selected feature set.

Motivation: In the last few years, a few efforts have been made by applying Rough Set methods towards feature selection [12], [13], [14]; however, these methods failed in addressing dynamic data, and solved feature selection as an offline problem. Moreover, these approaches merely focused on dimension condensation, even at the cost of feature (instance)-insensitiveness, and uncertainty [15]. Despite of the fact a few methods like applied Fuzzy Rough Set concept [4] is used to improve feature selection online, but they failed in addressing data redundancy, and hence underwent the compromised performance. Though, the concept of Incremental Feature Selection method has gained widespread attention, due to its capacity to study data online, and estimate the features effectiveness dynamically to keep the most important, while dropping the redundant one. The Incremental Feature Selection methods have considered different computing paradigms such as, learning old data first, and then appending learnt new data with the old selected features. However, a paradigm is executed with the minimum feature set, and sample size to increase both feature, and sample dynamically to attain the expected maximum 4V performance.

Contributions:

1. To achieve 4V-centric BigData analytics, the FSPS model applies Incremental Feature Selection followed by FSPIS method is used, that helps not only to retain reduced feature counts but also sample size to attain computationally efficient analytics.
2. Since, the FSPIS method applies both feature selection as well as sample selection as the cumulative solution for Incremental Feature Selection; it avoids the need of a separate sample selection method. It enhances the computational efficiency of the BigData analytics solution.
3. Unlike classical approaches, FSPIS model contributed a first of its kind feature sensitive progressive sampling concept that applied K-means clustering over each selected features and performed stratified sample selection method to ensure that even with the minimum sample size, the proposed features deliver better performance.

Organization: The manuscript is structured into five main sections. The related work presented in section II, followed by Section III which outlines the problem definition and objectives. The system model presented in section IV, while section V highlights results and subsequent discussion. Lastly, section VI represents conclusions and inferences.

2. Related Work

Realizing the efficacy of rough-set methods, significant efforts have been dedicated in recent years where the key emphasis was made on feature variations, values and instances. In sync with these key aspects, Wang *et al.*, [16] focused merely on feature variation by applying classical rough-set algorithms. In this method, feature entropy information was applied as a decision variable to perform dynamic feature selection in an incremental manner. Unlike [16], Shu and Shen [17] developed an incremental feature selection model which performed iterative feature addition and feature deletion concept to achieve higher accuracy. However, this approach was computationally exhaustive and even limited for large input data size with high dimensional features. As an enhanced solution, Qian *et al.*, [18] proposed a simultaneous addition and deletion-based feature set selection model by executing knowledge granulation updates within systems dealing with sets of information. Jing *et al.*, [19] assessed the efficacy of IFS by applying knowledge granularity information. This method, initially evaluated a granular feature matrix, which were subsequently utilized it incrementally to identify the most suitable set of features.

To cope up with dynamic data, [20] have proposed streaming feature selection methods. Javidi and Eskandari [20] began by estimating the significance level of each feature, and later utilized a rough set algorithm to conduct feature selection on a per stream basis. An improved solution was suggested by Liu *et al.,* [10] developed a method for real time multi-label streaming feature selection. Unlike [21], in [10] applied neighborhood rough sets algorithm to perform incremental feature selection. Zhou *et al.,* [22] contributed a real time dynamic feature selection model on the basis of proximity rough sets algorithm, which was especially designed to operate with an imbalanced dataset. Towards feature value variation cases, authors employed rough set-based IFS. Wang *et al.,* [23] designed an IFS model by applying information entropy estimates. Exploiting the variances in feature values over one to another instance, Shu *et al.,* [24] applied two IFS methods for incremental values associated with positive region.

Initially the instance variation information of dynamic instance is applied for only one instance. For the newcomer instance, Liu *et al.,* [25] intended to identify the minimum values within an information system devoid of decision labels to perform incremental feature selection. Chen *et al.,* [26] developed an IFS concept by using rough sets variable precision. Yet these techniques were inadequate in addressing BigData analytics challenges, especially high dimensionality and large instances over multiple features.

Unlike above stated approaches, an efforts have been made towards increasing instance sets dynamically. Das *et al.,* [27] focused on increasing instance sets rather than reducing the feature sets. For instance, Liang *et al.,* [28] proposed for group instance addition to achieve better performance. A group incremental feature selection concept by applying three entropy measures using rough sets. Here, rough-sets helped in deciding the group of instances to be added iteratively to achieve better performance. Zeng *et al.,* [29] developed single addition and deletion (of feature) model for incremental feature selection. Gaussian kernelized fuzzy rough sets is used to estimate the dependency amongst feature for incremental feature selection. Yang *et al.,* [30] applied a fuzzy rough set algorithm towards incremental feature selection. To update the relative discernibility relations with the old instances and the incoming one instance to perform continuous instance increment. In subsequent work, Yang *et al.,* [31] developed two feature selection models using fuzzy rough sets. This approach, initially they chop the input data into multiple chunks and estimating the incremental relative discernibility associations to iteratively refine the feature subsets. Noticeably, estimating the relative discernibility relationship involves a computationally exhaustive process, as it necessitates n x n comparative discernibility relation matrices for every individual feature. The n denotes the amount of instances associated with each feature. In such conditions, [30] and [31] can be limited due to large instances in each feature. It indicates the scope of both features as well as associated instance selection, simultaneously. Though researchers claimed that information entropy does not consume more memory and computation in comparison to the relative discernibility relations the iterative fuzzy rough-set for entropy estimation and feature update makes it more complex and exhausting. Although, entropy information can aid in achieving feature selection [32], it fails to consider the combined aspects of both features and instances simultaneously.

Moreover, the above stated methods do not address the problem when a large instance including both old data as well as newcomer one possesses the same level of significance or values. Addressing this problem can lead to improvements in both computational cost and accuracy. The utilization of active incremental feature selection [33] emerges as a promising solution, enabling the dynamic update of features based on representative instances.

3. Problem Statement and Objectives

3.1 Problem Statement

To develop a highly robust improved Chi-Square driven Select-k-Best algorithm with feature sensitive incremental instance selection model.

3.2 Objectives

1. To minimizing the suitable set of features.
2. To maximizing the output accuracy, and
3. To reduce the computational time.

4. System Model

The key intend of FSPIS model is to design a robust feature selection method which could ensure superior performance with low-redundant computation and higher accuracy. The matter of fact is that the optimality of a feature selection method depends on the fact that how significant feature it retains, and with what optimal learning environment it classifies the data to ensure better performance. In other words, a feature selection algorithm can be effective only when it is armoured with suitable feature extraction, sample selection, and classification methods. Considering this fact, FSPIS model intends to improve feature selection, sample selection, and classification functions. To analyse efficacy of the FSPIS model under various data conditions, FSPIS model is employed on different benchmark data for BigData analytics, such as Breast Cancer, Sonar, Lung Cancer, Parkinsons, WDBC, Ionosphere, KC1, Page Blocks, PC1 and Scene. Noticeably, these datasets possess the different features along with the different data sizes, and therefore satisfactory performance over the different data can enable the FSPIS model to be generalized for BigData analytics. Noticeably, the aforesaid data is considered in this study and did not require more sophisticated pre-processing or feature

extraction algorithms, and hence, focused more on feature selection, sample selection and classification problems. In this reference, the FSPIS model introduces a ground-breaking incremental feature selection method that retains minimum possible features while ensuring minimal (corresponding) samples from the selected features so as to ensure optimal performance. Undeniably, in the efficacy or limitation of a feature selection method or resulting selected features. Over the same selected features, the different classifiers can perform distinctly and hence generalizing suitability of a feature selection method and allied feature sets can be challenging.

To alleviate this problem, in the FSPIS model a maximum voting ensemble driven consensus model is designed to perform classification. Unlike traditional standalone classifier-driven prediction, the proposed model employs k-NN, Bagging, AdaBoost, Random Forest and Extended Tree Classifier as foundational classifiers. These basic classifiers perform distinct classification and label each instance with corresponding class-label. Subsequently, the proposed ensemble model estimates consensus for each data element by employing labels given by each base-classifier. Thus, a data with the higher class-label is predicted as that (corresponding) class. In this manner, being consensus-driven classification, the FSPIS model provides more accurate and reliable performance. Thus, aligned with the aforementioned research implementation modality, FSPIS model encircled with the subsequent key phases:

1. Data Acquisition and Preprocessing
2. Incremental Feature Selection
3. Feature Sensitive Incremental Instance Selection
4. Ensemble Learning based Classification.

Fig. 1 illustrate the overall proposed model and in the subsequent section in depth discussion of the model is provided.

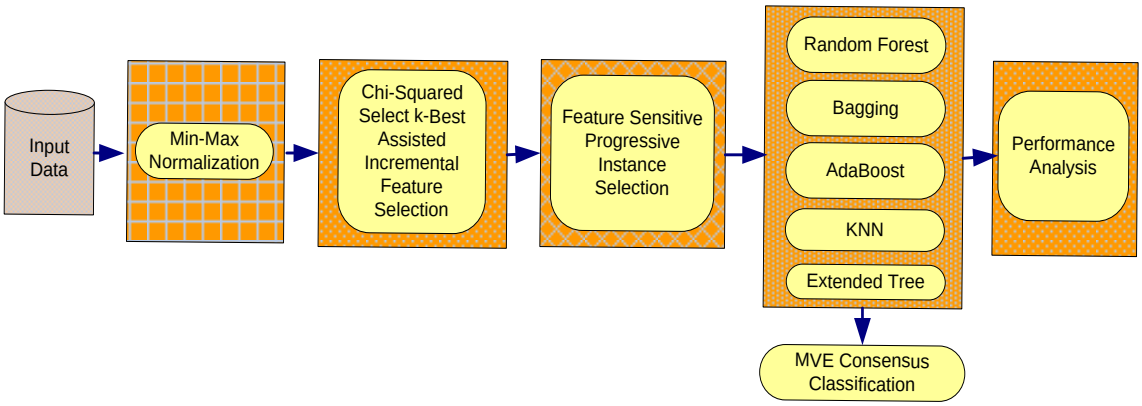


Fig. 1: Overall Proposed BigData Analytics Model with Incremental Feature Selection

4.1 Data Acquisition and Preprocessing

To evaluate the FSPIS model performance, consider the various benchmark datasets possessing the varied number of features, classes and instance sizes. Here, the key motive was to evaluate whether the FSPIS model can perform efficiently over the different data conditions. Obtained a total of 10 benchmark datasets from UCI ML Repository, accessed on https://www.openml.org/search platform. A snippet of the dataset is considered and their corresponding features are given in Table I.

The Table I signifies the diversity of the datasets and hence an efficient performance by the proposed feature selection model over these datasets can indicate its suitability towards real-time BigData analytics tasks. In sync with the original data and the different non-linear ranges, to alleviate any possibility of over-fitting during training, before executing the feature selection, processed for Min-Max normalization.

Table I. Dataset Collection

Datasets	Instances	Features	Classes
Breast Cancer	0699	009	2
Sonar	0208	060	2
Lung Cancer	0226	024	2
Parkinson	0195	022	2
WDBC	0569	030	2
Ionosphere	0351	034	2
KC1	2110	021	2
Page Blocks	5473	010	2
PC1	1109	021	2
Scene	2407	299	2

The proposed normalization model mapped the input data into the range of 0 to 1, and thus alleviated any possibility of convergence and over-fitting due to data non-linearity. Functionally, in the proposed method, each data element p_i of the selected features P is assigned a normalized value p_i, ensuring it falls within the range of [0, 1]. An equation (1) is used to calculate normalized value(s) of the data input p_i.

$$Norm(p_i) = \frac{p_i - min\ (P)}{max(P) - \ min\ (P)} \tag{1}$$

4.2 Incremental Feature Selection (IFS)

As stated above, this work primarily focused on designing a powerful and efficient incremental feature selection model followed by incremental instance selection. The key motive is to maintain the important features while processing the minimum possible instances for the selected features to make the computation more efficient. To achieve it, developed a state-of-art a novel and robust incremental feature selection method using improved dual-objective driven Chi-Squared concept. This approach uses the SkB method to evaluate significant rank of the features. To achieve it, applied dual objects driven Chi-Squared concept which aid in evaluating a set of suitable important features (say, top-k features). These top-k features have been later processed for instance selection using Feature Sensitive Progressive Sampling (FSPS) method.

4.2.1 Single Variable Chi-Squared Test

The Chi-Squared feature selection employs the χ^2 statistics to estimate the significance level of a feature by comparing it to the target class. This method assesses each feature individually to understand the level of significance or the strength of its relationship with the variable target. Functionally, it acts as a vital semi-parametric evaluation method especially applied to contrast more than two variables within arbitrarily selected datasets. Sometimes, it is also called the independence-test approach, as it enables finding the independence in between two arbitrary values or variables. Thus, it estimates a value on the basis of the associations in between the instance and the class it should belong to. In case of 0-value, it signifies the absence of any association between the instance and the class. In contrast, higher value, signifies the stronger relationship between the instance and corresponding class. In the FSPIS model, applied Chi-Squared method from the scikit-learn library to conduct initial feature estimation. SkB method which selected the k-highest scoring features from the complete feature space. Functionally, Chi-squares statistics estimation acts in sync with the information-theoretic feature selection method, where it intends to retrieve the intuition that the best terms t_k for certain class c_i exhibit distinctive distribution among both positive and negative examples of the class c_i. In this reference, Chi-squares assessment is performed using the equation (2).

$$Chi - Sqr(t_k, c_i) = \frac{N(AD - CB)^2}{(A + C)(B + D)(C + D)} \tag{2}$$

The equation (2) signifies the Chi-Square estimation, which assigns a score to each feature within each class. The description of the symbols is provided in the Table II.

Table II. Notations

Symbol	Description
N	The total numbers of data or documents in the corpus
A	The data within class c_i that contain the term t_k
B	The count of documents containing the term t_k in different classes
C	The count of documents within class c_i which do not encompass any term of t_k
D	The count of documents which do not encompass any term $t_k$in different classes

Subsequently, the individual scores were combined into a final composite score as in equation (3).

$$MAX\big(Chi - Sqr(t_k, c_i)\big) \quad (3)$$

Now, unlike classical Chi-Squared methods [34], [35], [36] where merely the highest score of the feature is considered as the decision variable for feature selection, and the dual objective driven score estimation. In other words, FSPIS model aims to identify or retain a feature subset capable of achieving improved accuracy, when dealing with minimal features. In the FSPIS model, this fitness estimation as in equation (4) is obtained for each feature.

$$Fitness = \alpha A + \beta B \quad (4)$$

The parameters *A* and *B* represents the amount of features and the minimal expected accuracy, in equation (4).

Noticeably, being an incremental feature method, the values of α and β are dynamically fine-tuned to align with the desired objective function (4). In the FSPIS model, the α and β are assigned to lower threshold. A majority of the classical feature selection estimates feature rank (Example, Chi-Squared rank, or Pearson Correlation Coefficient) and retains those features having higher rank, irrespective of the fact that such selected features might impact the overall accuracy over run-time execution. To alleviate this problem, FSPIS model is utilized as a dual-objective driven fitness function. It aims to minimize the value of *X* while simultaneously maximizing accuracy (*Y*).

Thus, in order to meet the requirements for higher accuracy, assigned a weight of 20% (α) to the features and a weight of 80% (β) to the accuracy. The parameters assigned α = 0.2, while β = 0.8. As stated, in the FSPIS model, the value of α is dynamically tuned to achieve the desired fitness value. Mathematically, updated the value of feature sets weight α using equation (5).

$$A = 1 - \left(\frac{Number\ of\ Selected\ Features}{Total\ number\ of\ Features}\right) \quad (5)$$

Thus, the fitness function is updated by incorporating the tuned number of features in equation (5), and obtained Fitness as in equation (6).

$$Fitness = \left[1 - \left(\frac{Number\ of\ Selected\ Features}{Total\ number\ of\ Features}\right) + 0.8 * Accuracy\right] \quad (6)$$

Thus, applying above stated fitness value, FSPIS Incremental Feature Selection method gradually increases the features from 20% to 80%, with 20% (0.2) serving as the lower bound and 0.8 defining the upper bound. To be noted, to achieve higher accuracy with minimum features, FSPIS method increases the number of features (*i.e.,* the nominator in the first component of equation (6)) by the fraction of 5% (*i.e.,* here, assigned increment factor as 0.5). Using this approach, features are progressively added to the existing feature sets until the desired accuracy level or region is attained. Noticeably, here the new feature is appended to (6) by picking the top-k feature obtained by means of Chi-Square selected feature set, where the use of SkB method features are retained in the order or decreasing score. Thus, the FSPIS model achieves a minimal feature sets that maximizes possible accuracy to perform generalizable classification. IFS and progressive sampling is performed after selecting the features in the FSPIS model.

4.3 Feature Sensitive Progressive Instance Selection (FSPIS)

It is evident that utilizing a reduced feature set can have better computational efficiency, and hence in this conjunction the proposed Incremental Feature Selection method that even considers accuracy as well as selection criteria can yield superior performance. However, in typical BigData analytics problems, where each feature can have

gigantically huge data size or the instance size. Similar to the redundant feature processing problem, the likelihood of redundant instance processing cannot be ignored. In other words, amongst the gigantically large sample size, there can be the set of minimum samples which can help the analytics model to yield the same level of accuracy, as is expected from the complete data size. Therefore, dropping such redundant samples or instances can help an analytics model to perform superior. To achieve it, recently a new technology called progressive sampling has been proposed [37], [38] that intends to retain minimum possible samples from each selected feature to perform prediction or classification, without compromising the accuracy. Considering it as motivation, in this work in addition to the above discussed Incremental Feature Selection method, we developed a state-of-art novel and robust clustering driven feature sensitive instance selection (FSPIS) or FSPS model. Unlike traditional resampling techniques such as up-sampling, down-sampling, random sampling or Synthetic Minority Over Sampling (SMOTE) methods, which often undergo class-imbalance, this study introduces a model called feature sensitive progressive sampling and allied instance selection model (FSPIS), which is the first of its kind.

As stated above the random instance selection from a feature set might cause class-imbalance where the selected instances might skew the overall feature pattern or the data pattern.

$$\begin{matrix} F_1 & Fs_{11} & Fs_{12} & \dots & Fs_{1n} \\ F_2 & Fs_{21} & Fs_{22} & \dots & Fs_{2n} \\ \vdots & \dots & \cdots & \dots & \dots \\ F_{sn} & Fs_{n1} & Fs_{n2} & \dots & Fs_{nn} \end{matrix} \quad (7)$$

Let, (7) be the input feature cum data space, where

F_n represents the selected features, while

sn be the data or instance pertaining to the

F_1 feature. In this case, the classical instance selection or sampling method randomly picks up the sample sn from each feature for subsequent learning and classification. Although, the significance of $sn1$ can be different than snn from F_n feature set. Therefore, the random selection of $sn(1)$ might either cause an instance set to undergo minority or majority, and this process can be continued iteratively due to random sample selection. Moreover, presence of data skewness can potentially lead to the learning model to get skewed towards a specific class due to improper sampling. Additionally, the minority class may not be classified accurately. Consequently, it can compel the entire learning model to exhibit false positives, thereby negatively affecting real-time analytics decisions. Considering this problem, this work hypothesized that retaining stratified samples from the different features can help a learning model learn better. In this relation, the FSPIS model, in which clustered the instances of each selected feature using K-Means clustering algorithm. In this method, once estimating the set of F_n features, applied K-Means clustering over each feature where the samples or the instances per feature vector were clustered into five distinct clusters. Subsequently, applied incremental Stratified Progressive Sampling (SPS) or instance selection concept that selects a specific quantity of instances from each cluster belonging to each feature depicted in Fig. 2. This process is continued till the selected features and corresponding instance yields expected performance.

Noticeably, unlike classical stratified sampling-based instance selection, the proposed FSPIS model is executed as an incremental instance selection method, where it updates the sample size iteratively by selecting the same number of instances from each cluster (per feature) dynamically to achieve target performance. Mathematically, it applies equation (8) to perform (dynamic) instance selection.

$$S_i = S_0 + \Delta S_\theta \quad (8)$$

In (8), S_i denotes the updated size of the data, whereas S_0 represents the initial size of the sample chosen as 20%.

Another parameter ΔS_θ denotes the progressive addition value, ranging between 0.5% to 5%. Through iterative steps ΔS_θ is successively added to S_0 until expected performance (accuracy) is achieved.

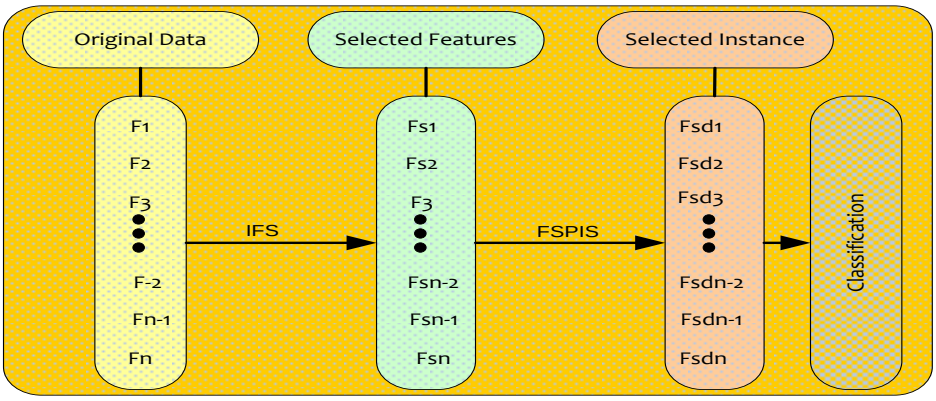


Fig. 2. Proposed Feature Sensitive Progressive Instance Selection (FSPIS) Model

Unlike to sampling approaches based on random selection and feature selection method [39], in the proposed FSPIS model, the samples selected from each cluster, taking into account the different features. This approach ensures maximum diversity of features in the training of the model, resulting in improved accuracy. Additionally, by extracting an equal number of samples from K1 cluster to K5 cluster for every feature, as presented in Fig. 3. It aims to mitigate data skewness and prevent overfitting.

4.3.1 K-Means Clustering Algorithm

Typically, it is a kind of unsupervised machine learning method that clusters a large number of data instances into corresponding groups. In other words, K-Means groups a large number of unannotated or unlabeled data instances to the specific group. Functionally, it intends to estimate the groups in a large set of data based on their respective features. The amount number of clusters is denoted by the *K*. To perform clustering this algorithm is executed iteratively where in each iteration it intends to assign a data element to the most relevant cluster and thus continues assigning the data to one of the *K*-clusters based on respective features (9).

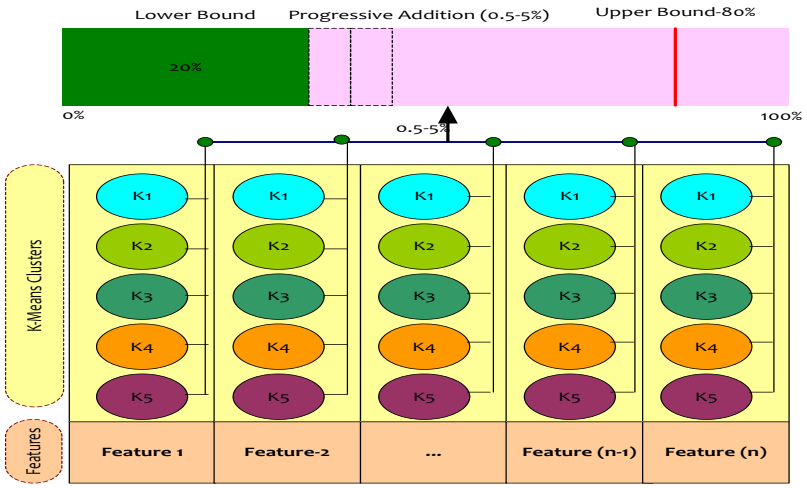


Fig. 3. Proposed Progressive Instance Selection (PIS) Model

Thus, employing feature similarity it maps or assigns each data element to their corresponding cluster. Functionally, it applies two key functions, data assignment and centroid update to perform data clustering.

4.3.1.1 Data Assignment

In the data assignment phase, it randomly selects one centroid for each cluster arbitrarily. Subsequently, each data element is allocated to the closest centroid, determined by calculating the squared Euclidean distance (inter-element distance information). Let c_j be the centroid for the cluster *C*, then in this reference each data or instance *x* is assigned to a group based on the distance-driven condition (9).

$$\underset{c_j \in C}{arg\ min}\ dist(c_j, x)^2 \tag{9}$$

In (9), the function dist (.) signifies a distance function that is considered as the Euclidean distance. Equation (9) indicates that those data elements having minimum distance from the centroid of a cluster j would form a cluster. In this manner, the set of data elements are mapped and allocated to each i^{th} centroid, s_i. The centroid is updated as per the following process.

4.3.1.2 Centroid Update

In this mechanism, the centroid for each cluster is updated iteratively by applying the average value of all data elements mapped to that specific cluster's centroid. It applies (10) to update the centroid iteratively.

$$c_i = \frac{1}{|s_i|}\sum_{x_i \in s_i} x_i \quad (10)$$

The algorithm continues until all the data elements are assigned to the allied cluster. To perform clustering, K-means algorithm applies a centroid value with respect to which the other data elements having the similar or close feature are clustered together. Functionally, with provided input data, K-means algorithm exploits features of each participating data element and maps them to the most related or relevant cluster. In the proposed FSPIS model, K-means algorithm was applied over each selected feature that generated five distinct clusters from each feature. Subsequently, the proposed FSPIS Model appended the final feature vector or allied sample (incremental volume of samples) from each cluster and executed the proposed ensemble learning model to perform classification.

4.4 Ensemble Learning Driven Consensus based Classification

In contrast to conventional self-contained classifier-based learning methods, this study adopts an ensemble learning-supported consensus-based classification framework. In the context of a binary classification problem (as shown in Table I), each classifier assigns a label of 1 or 0 to the data during the classification process. Thus, employing the MVE, estimated the consensus for each data and predicted the data as the class with the higher voting score. In the proposed ensemble model, four base-classifiers have been applied. They are:

1. k-NN
2. Random Forest
3. AdaBoost
4. Extended Tree Ensemble Classifier.

4.4.1 k-NN Algorithm

It is a well-known classifier, is widely recognized and popular model used to classify unlabelled observations. It assigns the unlabelled data to the class associated with the most related labelled examples. While k-NN offers a straightforward implementation, making it suitable for various data mining tasks involving regression prediction, it has proven to be robust in various classification frameworks. By default, the classifier utilizes Euclidean distance metric to calculate inter-attribute distance using (11).

$$D(p,q) = Sqrt((p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_n - q_n)^2) \quad (11)$$

In (11), both r and s are subjected to comparison across *n* number of features.

The k-NN algorithm effectiveness relies on the choice of *K*, which determines the number of neighbors needed for classification. Optimal selection of *K* leads to improved performance. When *K* is large, it reduces the impact of random errors and requires a smaller number of sample training data to be utilized. In essence, finding the right balance between overfitting and underfitting, which is important for achieving optimal performance while managing computational resources, depends on selecting the appropriate *K* value. Traditionally, researchers have often assigned *K* to be the square root of the number of observations or instances in the training data in traditional approaches. However, this technique may not guarantee efficacy when handling extensive datasers exhibiting diverse patterns. In many of the current approaches, *K* values are determined based on the size of the sample using the cross-validation scheme, although, leads to significant time consumption.

Unlike conventional k-NN algorithm, applied a kTree learning that enables learning distinct *k* values for the various training samples. During the training process, the kTree model initially learns the optimal value of *k* for each data sample through the utilization of a sparse reconstruction mechanism. Consequently, a decision tree (kTree) is constructed using the training samples and the optimal *k* values that were learned. The proposed kTree model rapidly outputs the value of *K* for each testing data sample during the testing phase. This is followed by performing k-NN classification using the learned optimal *k* value and training data. The proposed FSPIS model enables comparable running cost with better accuracy that makes it a potential candidate towards BigData analytics.

4.4.2 Adaboost

It is an adaptive boosting method, is a widely used learning paradigm known for enhancing the classification capability iteratively. During initialization, a set of prerequisite tests is allocated equal weights to achieve weak learners with limited training focus. In each iteration, the error rate of the weak classifier is evaluated, leading to an increase in the weight of correctly classified samples and a decrease in weight for misclassified samples. Eventually, this process strengthens the weak learner and enables successful classification [40].

4.4.3 Random Forest (RF) Algorithm

It represents an ensemble machine learning approach comprising numerous classifiers structured like trees. In the ensemble, every individual tree contributes a single vote to determine the most probable class for a given input. To

construct each tree, a random sample of *N* cases is selected from the original dataset, where *N* represents the total number of cases. Further, the chosen sample is then utilized for the training to build the tree. The tree nodes are then split based on the best division among the M input variables. The value for *M* maintains constant during the forest development, and each tree is grown to its maximum level. Compared to other machine learning algorithms such as Support Vector Machine, J48, Neural Network, Decision Tree (DT), and k-NN, the RF algorithm offers lower parameter estimations, making it computationally more efficient.

A group of various classifiers with tree structured is defined as (12) in RF algorithm,

$$\{RFc(p, \theta_k), k = 1,2, \dots i \dots\} \quad (12)$$

In (12), *RFc* denotes the classifier used, and θ_k represents an identically distributed random vector It contributes that each tree has a vote for the class that is most likely to occur based on a specific input variable called p. $\theta_k$characteristics and number of dimensions depends on its usage in the building of a tree. The RF algorithm's outcome lies in the construction of each decision tree that makes up the forest.

It involves training each tree on a randomly sampled subset of training data, achieved through bootstrapping, which allows for the utilization of nearly 70% of the training dataset. The portion of the remaining dataset is known as Out-Of-Bag (OOB) data samples. These OOB samples are commonly used for internal cross-validation to assess the classification model's performance.

As stated above, the random forest algorithm acts as an ensemble comprising *T* trees. In the training phase, the decision trees are autonomously constructed using bootstrap training set with arbitrarily selected features by means of random sub-space selection and bagging approach. Here, each DT is formed by applying the following methods.

Select the training subset from the original training dataset *S* with replacement. The variable significance and misclassification error are calculated using OOB samples that are not included in the bootstrapped sample.

Randomly select D ≥ M features and identify the optimal split using Gini-Index.

Without being pruned, the tree should grow its highest extent

In the process of classification, the input data sample *s* is categorized by traversing via individual trees until a leaf node is reached. At each leaf node, the classification result is assigned (decision function *h*). In the last step estimated class label *y* is calculated by choosing the class with the highest number of vote count among the leaf nodes.

Mathematically,

$$y = \underset{c\in\{\ \}}{\text{argm}1,2,\dots,\text{Cax}} \sum_{t:h_t(x)=c}^{T} 1 \quad (13)$$

4.4.4 Extended Tree (ET) Classifier

It is a unique ensemble method consisting of a cluster of unpruned decision trees. Unlike the RF algorithm, it introduces randomness in both choices of attribute and selection of cut-point when tree node splitting, resulting in the creation of fully randomized trees that are not influenced by the output values of the training samples. This classifier stands out from other ensemble methods based on tree structures for two main factors. Firstly, randomly selects cut-points to split nodes, and secondly, it utilizes the complete training sample instead of bootstrap replicas to facilitate tree growth. To generate the final prediction output, the ET classifier combines the classified predictions from all the trees using the MVE (Multiple Voting Ensemble) method. This approach aims to reduce variance more effectively compared to the weaker randomization methods employed by other techniques. Additionally, the classifier achieves more accurate and effective classification results by using original training samples instead of bootstrap replicas which decreases the risk of bias.

4.5 Maximum Voting Ensemble (MVE) Consensus Classification

By utilizing the aforementioned classifiers as the foundational classifiers, the model executed an ensemble decision using MVE methodology. This approach involved aggregating the consensus values derived from the foundational classifiers to facilitate two class classification. The implemented ET Classifier categorized each data element or instance into two distinct categories, denoted by the labels 0 or 1. These labelled values were later used to build consensus so as to perform final prediction. The subsequent sections present the simulation results and related findings.

4.6 Performance Analysis

In this work performance metrics for the classification problem used are Accuracy, AUC, F-Measure and Computational Time in seconds.

5. Results and Discussions

This paper presents a state-of-art novel and robust incremental feature selection designed for BigData analytics. In sync with the 4V objectives of BigData analytics, in addition to the proposed Incremental Feature Selection model, this work incorporated a novel feature sensitive progressive instance selection. Thus, in FSPIS model, the IFS was targeted to cap the amount of features, while the FSPIS model was developed specially to reduce the data size (say, instance size or volume) so as to improve the analytics performance. To ensure lightweight

(incremental) feature selection, this work applied Chi-Square driven Select-k-Best (SkB) method. However, unlike classical Chi-Square approaches, the proposed FSPIS model applied an objective function or fitness function that aims to attain a minimal feature subset while maintaining better levels of accuracy. In other words, the initial step involved processing the original data to perform Chi-Square evaluation, which aided in determining the ranking of an individual feature for subsequent sorting. After sorting the features based on their corresponding ranks, the model implemented an incremental feature selection method. This concept involved selecting a specific amount of features based on desired accuracy level. Gradually incremented the number of features in ascending order, starting from a lower threshold to an upper limit. Specifically, the lower limit for the sorted features was set at 20%, with the upper limit remaining at 80%. It means that the proposed FSPIS model initially uses 20% of the total features. The model evaluates its accuracy then keeps increasing the feature percentile until it achieves the desired performance.

However, the highest possible feature selection was fixed at 80%. Subsequently, once selecting the appropriate feature set guaranteeing optimal performance, we initiated FSPIS, which applied K-means clustering over each feature. In the proposed FSPIS model, clustered each selected feature into five (K=5) clusters, and applied incremental stratified sampling concept to select instances from each cluster (per feature) in an incremental manner. Similar to the incremental feature selection method, assigned a lower limit of instance as 20% of the total data size (per feature), which was increased by 1% iteratively so as to get superior or expected accuracy level. Here fixed the highest sample size of data size as 80% of the total size. Thus, starting with the 20% of the instance volume or size, the proposed FSPIS model incremented the instance size by 1% iteratively, till it meets the expected performance. Unlike [39], where authors merely applied static stratified feature selection concept, the proposed FSPIS model employs both feature selection as well as instance selection in incremental manner which could provide the minimum possible feature sets as well as instance size while guaranteeing expected performance.

Moreover, unlike other approaches employing rough-set or entropy driven incremental feature selection methods [18-31], the proposed FSPIS model is relatively very computationally efficient and scalable to meet the desired accuracy while retaining minimum feature as well as data size. To evaluate performance of the FSPIS model, considered a total of 10 benchmark datasets encompassing Breast Cancer, Sonar, Lung Cancer, Parkinson, WDBC, Ionosphere, KC1, Page Blocks, PC1 and Scene. The consider datasets were having different numbers of features with varied instance sizes. Though, these data represented only two class classification problems. To evaluate performance of the FSPIS model, examine the outcome in terms of accuracy and the selected feature volume. The FSPIS model was implemented using Python 3.7 simulated over the Anaconda Jupiter platform. The computer system used was armored with Intel i-3 processor, 8GB RAM, 2.86 GHz frequency. The simulated results and related performance inferences are presented as follows.

To assess the effectiveness of the FSPIS model, performed two distinct analyses in the form of intra and inter model characterization. The intra-model refers to the analyses of the FSPIS model with the different operating conditions or algorithmic combinations. On the contrary, inter-model comparisons have compared the performance with the other related works. The following section provides an analysis of the obtained results along with relevant inferences.

5.1 Intra-Model Comparison

During this evaluation, compared the performance of the proposed model in three different simulating conditions. These are:

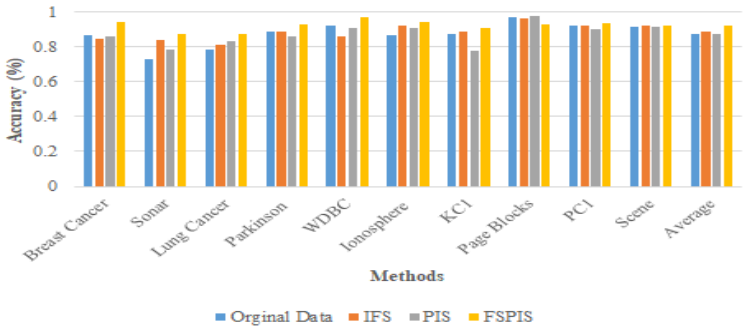
1. Original Features
2. Incremental Feature Selection (IFS)
3. PIS
4. FSPIS

In sync with the above stated simulation conditions, at first, executed and performed classification with the original data or original features; however, by considering data normalization and proposed maximum voting ensemble learning model as classifier. Subsequently, simulated the proposed IFS model, which is the part of the proposed work, and obtained the performance results encompassing selected features, computing time and corresponding accuracy. In the third simulation condition, mainly focused on reducing the feature instances (while keeping the number of features same as that of the original features). Finally, by combining the proposed IFS feature selection method with FSPIS algorithm that intended to retain minimum features as well as corresponding (minimum) instance size or volume. The simulation results under the different operating conditions are given in Table III.

As depicted in Table II, with the original data the accuracy obtained is relatively lower in comparison to the selected features (*i.e.,* by IFS feature selection method). Here, the impact of redundant feature learning can easily be visualized. In major BigData analytics problems, learning over the insignificant and relatively redundant features or even data elements impact the classification accuracy. On the contrary, training over a set of suitable features with the optimal data elements helps gaining superior performance. In sync with this hypothesis, the results obtained (Table III) confirms that the proposed feature selection method (*i.e.*, IFS) performs better average accuracy (88.70%) than the original data-based simulation (87.51%). On the other hand, the proposed K-Means driven progressive instance selection

Table III. Intra-Model Feature Sensitive Accuracy Assessment

Datasets	Original Data		IFS		PIS		FSPIS	
	Features	Accuracy	Features	Accuracy	Features	Accuracy	Features	Accuracy
Breast Cancer	9	0.8672	3	0.8491	9	0.8632	3	0.943
Sonar	60	0.7317	11	0.8427	60	0.7864	11	0.877
Lung Cancer	23	0.7869	4	0.8126	9	0.8357	4	0.873
Parkinson	22	0.8869	6	0.8894	22	0.859	6	0.929
WDBC	30	0.9231	6	0.8613	30	0.9071	4	0.969
Ionosphere	34	0.8686	4	0.9199	34	0.9093	4	0.94
KC1	21	0.8769	3	0.8894	21	0.779	3	0.909
Page Blocks	10	0.972	3	0.961	10	0.9737	3	0.927
PC1	21	0.9247	3	0.9223	21	0.9006	3	0.939
Scene	299	0.9135	13	0.9229	299	0.9154	13	0.923
Average		**0.8751**		**0.887**		**0.8729**		**0.9229**



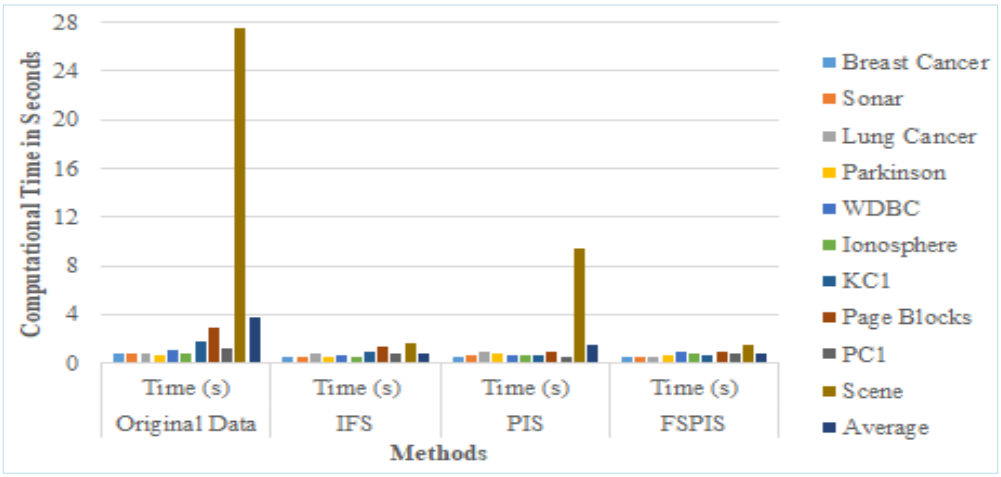
model (PIS) focuses on reducing the instance size so as to improve learning. Though, this approach applied the same features; it reduced the data size or instance size. The average accuracy obtained by PIS is 87.29%, which is almost near the original data; and its contribution in reducing the sample size cannot be ignored. Moreover, despite the fact that the use of PIS model would merely reduce the same size, while retaining the feature vectors as

it is, the minimally changed accuracy confirms that the use of PIS can be vital to achieve higher accuracy even with reduced sample size. It can make the BigData analytics model more computationally efficient. Now, realizing the fact that the IFS model can reduce the feature size, while the PIS model can help reduce both features as well as instances and hence can be more appropriate towards real-time BigData analytics. Considering this fact, the proposed model amalgamated both IFS feature selection and PIS sample selection of instance selection method. This as a result gave rise to the proposed Feature Sensitive Instance Selection model (FSPIS). To be noted, unlike PIS, which was executed over the original features only, the proposed FSPIS model was applied over the selected feature vector (by IFS algorithm).

Table IV. Intra-Model Feature Sensitive Computational Time Assessment

Datasets	Original Data		IFS		PIS		FSPIS	
	Features	Time (s)	Features	Time (s)	Features	Time (s)	Features (s)	Time
Breast Cancer	009	00.8421	03	0.5245	009	0.5741	03	0.5684

Sonar	060	00.8118	11	0.5804	060	0.6711	11	0.5696
Lung Cancer	023	00.8332	04	0.7503	009	0.9567	04	0.5265
Parkinson	022	00.6701	06	0.5365	022	0.7599	06	0.6941
WDBC	030	01.0465	06	0.6223	030	0.6123	04	0.8856
Ionosphere	034	00.7400	04	0.5654	034	0.5951	04	0.7579
KC1	021	01.7619	03	0.8796	021	0.6964	03	0.6921
Page Blocks	010	02.8750	03	1.4241	010	0.8992	03	0.9015
PC1	021	01.1837	03	0.7639	021	0.5804	03	0.8008
Scene	299	27.5499	13	1.6327	299	9.4098	13	1.4860
Average		**3.831**		**0.8279**		**1.5755**		**0.7882**



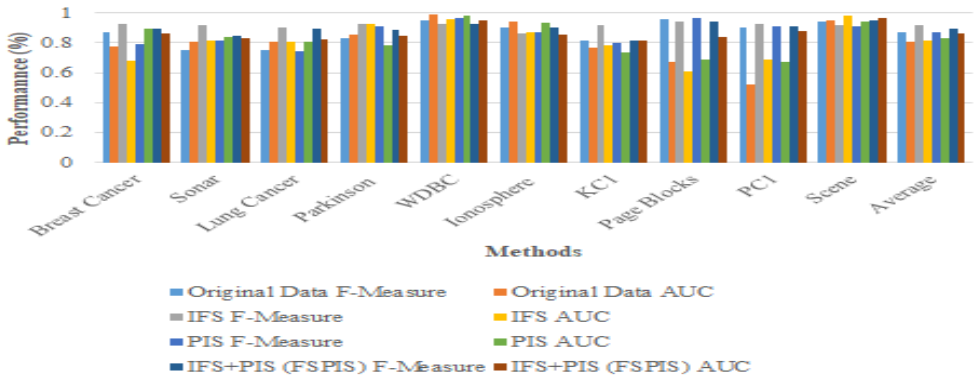
The results obtained from Table III demonstrate the superior performance of the proposed FSPIS model with IFS. It achieves an accuracy of 92.29% and also excels in selecting the optimal set of features. Typical BigData analytics solutions, there is a constant expectation for execution time to remain smaller to meet Velocity demands. Considering this fact, we examined the FSPIS model for its computational time efficiency.

Observing the results for the computational time efficiency (Table IV), it becomes evident that the proposed ensemble learning classifier consumes almost 3.83 seconds (average time) to perform computation over the considered datasets with the original data. Conversely, the proposed IFS model which reduces the size of features significantly takes 0.82 seconds, which is significantly lower than the original data. Similarly, the PIS model retained the original features but with reduced samples took almost 1.57 seconds to process the data. Interestingly, the proposed model (*i.e.*, FSPIS) took the minimum computational time (only 0.78 second) to process the entire data. Here, the role of IFS and PIS can easily be visualized. In other words, the proposed FSPIS model, IFS helped in reducing the feature set or size, while PIS helped in reducing the instance sizes of the IFS selected features. As a result, contributed in reducing the computational time. In addition to the accuracy assessment, it measured F-Measure as well as AUC performances.

Table V. F-Measure and AUC Performance

	Original Data		**IFS**		**PIS**		**FSPIS**	
Datasets	**F-Measure**	**AUC**	**F-Measure**	**AUC**	**F-Measure**	**AUC**	**F-Measure**	**AUC**
Breast Cancer	0.8697	0.7740	0.9289	0.6808	0.7919	0.8945	0.8966	0.8626

Sonar	0.7498	0.8093	0.9193	0.8110	0.8124	0.8348	0.8482	0.8266
Lung Cancer	0.7512	0.8082	0.9033	0.8056	0.7429	0.8077	0.8909	0.8196
Parkinson	0.8320	0.8531	0.9220	0.9277	0.9073	0.7790	0.8871	0.8468
WDBC	0.9507	0.9869	0.9256	0.9593	0.9640	0.9832	0.9289	0.9534
Ionosphere	0.8984	0.9417	0.8629	0.8714	0.8733	0.9321	0.8981	0.8533
KC1	0.8110	0.7667	0.9178	0.7802	0.7960	0.7389	0.8155	0.8171
Page Blocks	0.9574	0.6680	0.9442	0.6113	0.9666	0.6914	0.9400	0.8380
PC1	0.9038	0.5223	0.9261	0.6843	0.9075	0.6688	0.9109	0.8766
Scene	0.9439	0.9489	0.9159	0.9794	0.9135	0.9445	0.9485	0.9678
Average	**0.8667**	**0.8079**	**0.9166**	**0.8111**	**0.8675**	**0.8274**	**0.8964**	**0.8661**



The results presented in Table V highlight the proposed model with IFS+PIS (say, FSPIS) exhibits superior F-Measure (0.89) and AUC performance (0.86), which is higher in comparison to the original data, PIS and FIS as standalone simulation. The result confirms that the proposed FSPIS model achieves superior performance towards BigData analytics tasks, even under diverse non-linear features and class-imbalance conditions. It can be affirmed due to higher value of AUC (the AUC more than 0.66 confirms robustness of the proposed model). Meanwhile, higher F-score or F-Measure too confirms robustness of the proposed system in context of real-time BigData analytics. In sync with the above discussed results and allied inferences, we consider FSPIS as the proposed model for BigData analytics. For further inter-model comparison the performance by FSPIS is considered as the reference value.

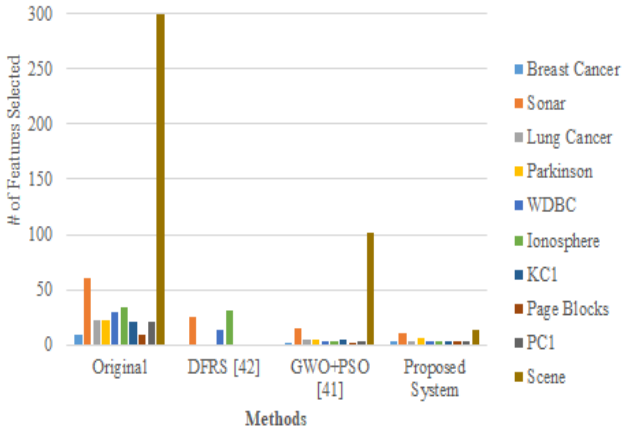
5.2 Inter-Model Assessment

To assess the relative effectiveness of the proposed system in comparison to existing methods, FSPIS model considered as the proposed system, while a few recent works like [41], [42]. Noticeably, in [41] authors proposed a feature selection model based on heuristic. Authors applied Gray Wolf Optimizer (GWO) and Particle Swarm Optimization (PSO) algorithms in their approach.

Table VI. Comparison of Results with Related Works

	Original Data	**GWO + PSO [41]**		**DFRS [42]**		**Proposed System**	
Datasets	**Features**	**Features**	**Accuracy**	**Features**	**Accuracy**	**Features**	**Accuracy**
Breast Cancer	9	2.25	0.967	-	-	3	0.943
Sonar	60	14.65	0.855	26	0.8558	11	0.877

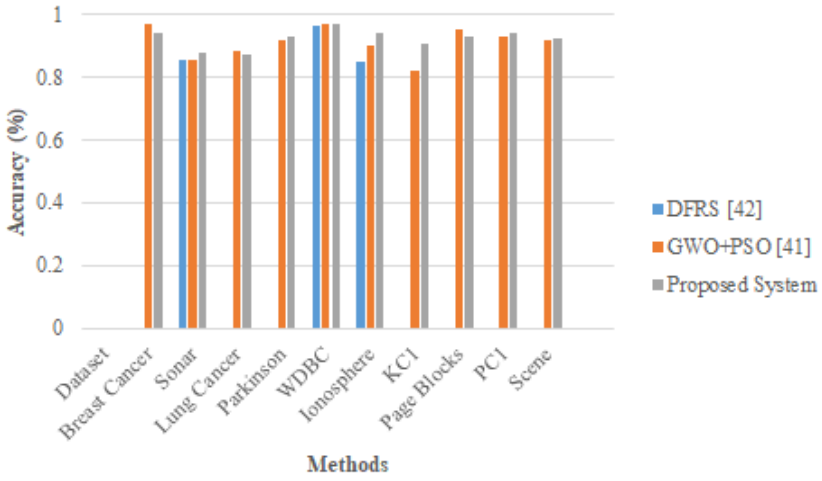
Lung Cancer	23	5.45	0.882	-	-	4	0.873
Parkinson	22	4.25	0.92	-	-	6	0.929
WDBC	30	3.8	0.97	13	0.9631	4	0.969
Ionosphere	34	3.9	0.901	32	0.8519	4	0.94
KC1	21	4.65	0.821	-	-	3	0.909
Page Blocks	10	2.3	0.955	-	-	3	0.927
PC1	21	3.1	0.931	-	-	3	0.939
Scene	299	101	0.919	-	-	13	0.923



Noticeably, these state-of-the-art methods primarily prioritize the selection of feature sets without considering the size of the sample. In such cases, when dealing with gigantically large data sizes and reduced feature vectors, the model may undergo challenges related to local minima and convergence. Authors [41] and [42] did not this problem. Moreover, heuristic-based approaches are mainly executed in an offline manner as it might take significantly large time to estimate the execution parameters and solution tuning. To perform relative performance analysis, consider average performance by both GWO and PSO in [41], while in [42] the performance outputs for the common data elements were taken into consideration. To be noted, in sync with performance characterization, for inter-model assessment, consider only those algorithms applying the same benchmark datasets.

The results presented in Table VI show that the FSPIS model exhibits superior accuracy.

By examining the above Table VI results, it is revealed that the proposed model amount of features selected is superior (*i.e.*, lower) even without losing accuracy performance. In other words, the average performance in [41] exhibited that the minimum features possible for breast cancer was 2.25, while the proposed FSPIS model identifies 3 crucial that have significant impact on the classification performance. Except for the performance with breast cancer data, the FSPIS model with other datasets has exhibited

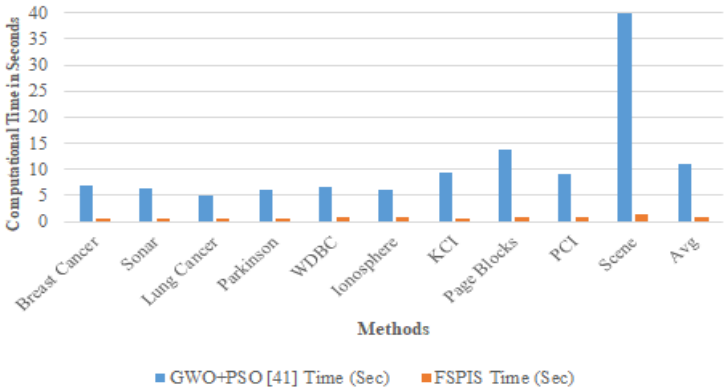


superior results. In reference to the Sonar dataset, the method proposed in [41] identified a total of 26 features as vital. On the other hand, GWO+PSO selected 14.65 features as the key features to be retained. While the proposed FSPIS model identified merely 11 features as significant (feature) to perform classification. Interestingly, with this dataset, the accuracy obtained with merely 11 features is 87.7%, which is better than DFRS [42] (85.58%) and GWO+PSO [41] (85.5%). It confirms that the proposed FSPIS model can obtain better accuracy even with the lower feature and instance volume. It backs up the proposed FSPIS model to real-time BigData analytics. Similar to these results, the Lung Cancer dataset performance demonstrated that the proposed FSPIS model identifies merely 4 features, while GWO+PSO chose 5.45 features for classification. The relative accuracy performance too confirms that the proposed FSPIS model maintains near [41] performance even with minimized feature set. Noticeably, as opposed to [41] or [42], the proposed FSPIS model performs both features as well as instance selection and therefore can be more computationally efficient towards real-time BigData analytics tasks. Considering the KC1 dataset, the existing GWO+PSO model selected an average of 4.65 feature sets, whereas the proposed FSPIS model selected merely 3 features to accomplish classification. In relation to, the proposed FSPIS model showcased an accuracy of 92.3%, which is better than the GWO+PSO model (*i.e.*, 91.2%). The similar performance patterns can be found with other datasets as well. In comparison to the efforts made in [41] or [42], or any other existing incremental feature selection methods (see, Section II), the computational efficiency of the proposed FSPIS model makes it more realistic and viable towards real-time BigData analytics.

Table VII. Analysis of Computational Time (sec) Comparison

Datasets	**Original Data Features**	**GWO + PSO [41]**		**Proposed System**	
		Features	**Time(s)**	**Features**	**Time (s)**
Breast Cancer	9	2.25	7	3	0.5684
Sonar	60	14.65	6.3	11	0.5696
Lung Cancer	23	5.45	4.9	4	0.5265
Parkinson	22	4.25	6	6	0.6941
WDBC	30	3.8	6.75	4	0.8856
Ionosphere	34	3.9	6.1	4	0.7579
KC1	21	4.65	9.4	3	0.6921
Page Blocks	10	2.3	13.75	3	0.9015
PC1	21	3.1	9	3	0.8008
Scene	299	101	40	13	1.486
Avg		**10.92**			**0.7882**

Here, the strength towards both feature selection as well as feature sensitive instance selection (as incremental selection measure) makes the proposed FSPIS system robust and more efficient than other state-of-art methodologies (refer, Section II). Aligned with this assertion, the computing time analysis substantiates the notion.



In Table VII, presented a computational time consumed by the proposed FSPIS model contrasted with other techniques, providing insights into its relative computational efficiency [41].

As already stated, authors in [41] introduced feature selection concepts driven by heuristics, which have faced criticism due to their extensive computational requirements and associated costs. In contrast, the proposed FSPIS model employs a straightforward analytical approach for feature selection. The results obtained, as presented in Table VI, confirms that the FSPIS model significantly reduces the time required for feature selection and classification. Conversely, despite GWO+PSO estimating same features it imposes significant time-related computational expenses, measured in seconds. The mean time utilized by the current GWO+PSO model amounted to 10.92 seconds, while the proposed FSPIS model consumed merely 0.78 seconds to carry out the overall task. Here, the efficiency of the proposed FSPIS model can easily be confirmed. Thus, Considering the comprehensive assessment of performance, it is concluded that the FSPIS model surpasses other existing approaches in terms of achieving optimal performance with fewer feature sets, minimum sample requirements, greater accuracy, AUC, and F-Measure. Also, it is crucial to emphasize its superior time efficacy. It confirms the resilience of the FSPIS model when applied to real-time analytics tasks. The comprehensive research findings and related deductions are detailed in the following section.

6. Conclusions

The exponential advancement of software technologies, internet and affordable hardware has revitalized global humanity to explore it for more efficient decision-making. The technological revolution and up-surging demands across the industries have introduced to a new technology referred as BigData analytics serving different purposes including healthcare, business communication, business intelligence, civic management, finance, science and technologies, social media, *etc*. Despite numerous significances, BigData analytics require addressing its challenges like heterogeneous unstructured data, multi-dimensional features, large instances, class-imbalance *etc.*, to gain optimally accurate target information. On the other hand, the key aspects of BigData like volume, variety, velocity and veracity too demand an analytics solution to be robust, time-efficient and accurate. To ensure fast computation, BigData analytics executes feature selection technique; however, its resulting accuracy remains a challenge.

Moreover, feature selection using random method results in reduced accuracy that make them inferior towards real-time purposes. Majority of the current feature selection methods apply thresholds, entropy information or level of significance to perform feature selection; although, fail in addressing large redundant data learning that eventually makes it computationally exhaustive and time consuming. To simplify this complexity, the paper introduces a robust incremental feature selection method was designed that intends to maintain lower feature dimension along with minimum possible instance set. To achieve it, the FSPIS model applied incremental feature selection followed by feature sensitive progressive instance selection. For incremental feature selection, the model utilizes the select k-best method with the Chi-Square algorithm, which employs a heuristic fitness function to minimize the number of features while maximizing accuracy. Thus, executing the proposed feature selection method, we ensure that the selected features provide sufficient volume while meeting the veracity demands in terms of accuracy. Moreover, the use of Chi-Square driven select-k-best model helped guaranteeing minimum feature requirement, while feature sensitive progressive sampling or instance selection enabled retaining minimum possible data size to meet accuracy demands. Thus, the proposed model retained minimum feature(s) and instance size that helped in achieving higher computational efficiency, higher accuracy with minimum computation time. It makes the proposed system suitable for the real-time BigData analytics tasks. To ensure higher

accuracy with minimum features and instances, the proposed model applied a maximum voting ensemble learning models encompassing Bagging, AdaBoost, k-NN, Random Forest and Extended-Tree Classifiers as foundational classifiers. Therefore, unlike standalone classifier-based model, the use of MVE consensus helped achieve highly accurate performance. The statistical performance analysis reveals that the proposed model estimates almost 18%-20% lower features than other feature selection methods. Moreover, with such reduced feature sets, it retained superior accuracy of the classification, F-Measure and AUC across different datasets. It enabled the proposed model to be used in different BigData analytics problems. Though, the proposed work achieved better performance than other state-of-art methods like rough-set algorithms, heuristic driven feature selection algorithms; it could not address class-imbalance problems which might come into picture due to significantly reduced feature size. Moreover, there can be a different data environment where the probability of class-imbalance cannot be ignored. In future, the proposed model can be improved with different resampling methods to alleviate class-imbalance probability.

Author contributions

Conceptualization, Dataset collection, Methodology, Implementation, result comparison and draft preparation done by first author. Supervision, review of work and managing project work have been done by second and third author.

References

[1] O. Duda, V. Kochan, N. Kunanets, O. Matsiuk, V. Pasichnyk, A. Sachenko, and T. Pytlenko, "Data Processing in IoT for Smart City Systems," *in proceedings of the IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, Technology and Applications*, vol. 1, pp. 96–99, 2019.

[2] H. Chiroma, U. A. Abdullahi, A. A. Alarood, L. A. Gabralla, N. Rana, L. Shuib, I. A. T. Hashem, D. E. Gbenga, A. I. Abubakar, and A. M. Zeki, "Progress on Artificial Neural Networks for Big Data Analytics: A Survey," *IEEE Access*, vol. 7, pp. 70 535–70 551, 2018.

[3] Kamble S, Arunalatha JS, Venkataravana Nayak K, Venugopal K R, "Chi-Square Top-K Based Incremental Feature Selection Model for BigData Analytics," *in proceedings of the Emerging Trends and Technologies on Intelligent Systems*, pp. 127-139, 2022.

[4] X. Zhang, C. Mei, D. Chen, Y. Yang, and J. Li, "Active Incremental Feature Selection using a Fuzzy Rough Set Based Information Entropy," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 901–915, 2019.

[5] C. Wang, Y. Huang, M. Shao, Q. Hu, and D. Chen, "Feature Selection Based Neighborhood Self-Informations," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 4031–4042, 2019.

[6] N. L. Giang, T. T. Ngan, T. M. Tuan, H. T. Phuong, M. Abdel Basset, A. R. L. de Macedo, and V. H. C. de Albuquerque, "Novel Incremental Algorithms for Attribute Reduction from Dynamic Decision Tables using Hybrid Filter-Wrapper with Fuzzy Partition Distance," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 858–873, 2019.

[7] L. Sun, L. Wang, W. Ding, Y. Qian, and J. Xu, "Neighborhood Multi-Granulation Rough Sets-Based Attribute Reduction using Lebesgue and Entropy Measures in Incomplete Neighbourhood Decision Systems," *Journal of Knowledge-Based Systems*, vol. 192, p. 105373, 2020.

[8] Z. A. Zhao and H. Liu, "Spectral Feature Selection for Data Mining," *Taylor & Francis*, pp. 1–220, 2012.

[9] F. Li, Z. Zhang, and C. Jin, "Feature Selection with Partition Differentiation Entropy for Large-Scale Data Sets," *Journal of Information Sciences*, vol. 329, pp. 690–700, 2016.

[10] J. Liu, Y. Lin, Y. Li, W. Weng, and S. Wu, "Online Multi-Label Streaming Feature Selection Based on Neighbourhood Rough Set," *Journal of Pattern Recognition*, vol. 84, pp. 273–287, 2018.

[11] H. Liu and L. Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," I*EEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[12] J. Dai, H. Hu, W.-Z. Wu, Y. Qian, and D. Huang, "Maximal-Discernibility-Pair-Based Approach to Attribute Reduction in Fuzzy Rough Sets," I*EEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2174–2187, 2017.

[13] C. Wang, Y. Qi, M. Shao, Q. Hu, D. Chen, Y. Qian, and Y. Lin, "A Fitting Model for Feature Selection with Fuzzy Rough Sets," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 741–753, 2016.

[14] X. Zhang, C. Mei, D. Chen, and Y. Yang, "A Fuzzy Rough Set-Based Feature Selection Method using Representative Instances," *Journal of Knowledge-Based Systems*, vol. 151, pp. 216–229, 2018.

[15] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[16] F. Wang, J. Liang, and Y. Qian, "Attribute Reduction: A Dimension Incremental Strategy," *Journal of Knowledge-Based Systems*, vol. 39, pp. 95–108, 2013.

[17] W. Shu and H. Shen, "Updating Attribute Reduction in Incomplete Decision Systems with the Variation of

Attribute Set," *International Journal of Approximate Reasoning*, vol. 55, no. 3, pp. 867–884, 2014.

[18] W. Qian, W. Shu, and C. Zhang, "Feature Selection from the Perspective of Knowledge Granulation in Dynamic Set-valued Information System," *Journal of Information Science and Engineering*, vol. 32, no. 3, 2016.

[19] Y. Jing, T. Li, J. Huang, and Y. Zhang, "An Incremental Attribute Reduction Approach Based on Knowledge Granularity Under the Attribute Generalization," *International Journal of Approximate Reasoning*, vol. 76, pp. 80–95, 2016.

[20] S. Eskandari and M. M. Javidi, "Online Streaming Feature Selection using Rough Sets," *International Journal of Approximate Reasoning*, vol. 69, pp. 35–57, 2016.

[21] M. M. Javidi and S. Eskandari, "Streamwise Feature Selection: A Rough Set Method," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 4, pp. 667–676, 2018.

[22] P. Zhou, X. Hu, P. Li, and X. Wu, "Online Feature Selection for High-Dimensional Class-Imbalanced Data," *Journal of Knowledge-Based Systems*, vol. 136, pp. 187–199, 2017.

[23] F. Wang, J. Liang, and C. Dang, "Attribute Reduction for Dynamic Data Sets," *Journal of Applied Soft Computing*, vol. 13, no. 1, pp. 676–689, 2013.

[24] W. Shu and H. Shen, "Incremental Feature Selection Based on Rough Set in Dynamic Incomplete Data," *Journal of Pattern Recognition*, vol. 47, no. 12, pp. 3890–3906, 2014.

[25] Z. T. Liu, "An Incremental Arithmetic for The Smallest Reduction of Attributes," Journal of Acta Electronica Sinica, vol. 27, no. 11, pp. 96–98, 1999.

[26] D. Chen, Y. Yang, and Z. Dong, "An Incremental Algorithm for Attribute Reduction with Variable Precision Rough Sets," *Journal of Applied Soft Computing*, vol. 45, pp. 129–149, 2016.

[27] K. Das, S. Sengupta, and S. Bhattacharyya, "A Group Incremental Feature Selection for Classification using Rough Set Theory Based Genetic Algorithm," *Applied Soft Computing*, vol. 65, pp. 400–411, 2018.

[28] J. Liang, F. Wang, C. Dang, and Y. Qian, "A Group Incremental Approach to Feature Selection Applying Rough Set Technique," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 294–308, 2012.

[29] Zeng, T. Li, D. Liu, J. Zhang, and H. Chen, "A Fuzzy Rough Set Approach for Incremental Feature Selection on Hybrid Information Systems," Journal of Fuzzy Sets and Systems, vol. 258, pp. 39–60, 2015.

[30] Y. Yang, D. Chen, H. Wang, E. C. Tsang, and D. Zhang, "Fuzzy Rough Set Based Incremental Attribute Reduction from Dynamic Data with Sample Arriving," *Journal of Fuzzy Sets and Systems*, vol. 312, pp. 66–86, 2017.

[31] Y. Yang, D. Chen, H. Wang, and X. Wang, "Incremental Perspective for Feature Selection Based on Fuzzy Rough Sets," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1257–1273, 2017.

[32] P. Zhou, X. Hu, P. Li, and X. Wu, "OFS-Density: A Novel Online Streaming Feature Selection Method," *Journal of Pattern Recognition*, vol. 86, pp. 48–61, 2019.

[33] H. Liu, H. Motoda, and L. Yu, "A Selective Sampling Approach to Active Feature Selection," *Journal of Artificial Intelligence*, vol. 159, no. 1-2, pp. 49–74, 2004.

[34] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature Selection using an Improved Chi-Square for Arabic Text Classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp. 225–231, 2020.

[35] T. Siswantining, D. Sarwinda, and A. Bustamam, "RFE and Chi-Square Based Feature Selection Approach for Detection of Diabetic Retinopathy," *in Proceedings of the International Joint Conference on Science and Engineering*, pp. 380–386, 2020.

[36] T. Parlar, S. A. O¨ zel, and F. Song, "QER: A New Feature Selection Method for Sentiment Analysis," *Journal of Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–19, 2018.

[37] S. Parthasarathy, "Efficient Progressive Sampling for Association Rules," *in Proceedings of the IEEE International Conference on Data Mining*, pp. 354–361, 2002.

[38] K. T. Chuang, M. S. Chen, and W. C. Yang, "Progressive Sampling for Association Rules Based on Sampling Error Estimation," *in Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 505–515, 2005.

[39] Y. Yang, P. Yin, Z. Luo, W. Gu, R. Chen, and Q. Wu, "Informative Feature Clustering and Selection for Gene Expression Data," *IEEE Access*, vol. 7, pp. 169 174–169 184, 2019.

[40] C. X. Zhang and J. S. Zhang, "RotBoost: A Technique for Combining Rotation Forest and AdaBoost," *Journal of Pattern Recognition Letters*, vol. 29, no. 10, pp. 1524–1536, 2008.

[41] M. El-Hasnony, S. I. Barakat, M. Elhoseny, and R. R. Mostafa, "Improved Feature Selection Model for Big Data Analytics," *IEEE Access*, vol. 8, pp. 66 989–67 004, 2020.

[42] L. Kong, W. Qu, J. Yu, H. Zuo, G. Chen, F. Xiong, S. Pan, S. Lin, and M. Qiu, "Distributed Feature Selection for Big Data using Fuzzy Rough Sets," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 846–857, 2019.

[43] Mr. Kaustubh Patil. (2013). Optimization of Classified Satellite Images using DWT and Fuzzy Logic. International Journal of New Practices in Management and Engineering, 2(02), 08 - 12. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/15

[44] Lavanya, A. ., & Priya, N. S. . (2023). Enriched Model of Case Based Reasoning and Neutrosophic Intelligent System for DDoS Attack Defence in Software Defined Network based Cloud. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 141–148. https://doi.org/10.17762/ijritcc.v11i4s.6320