IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 May 2022, accepted 12 June 2022, date of publication 30 June 2022, date of current version 5 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187412

An Integrated Novel Framework for Coping Missing Values Imputation and Classification

MONALISA JENA AND SATCHIDANANDA DEHURI
Department of Computer Science, Fakir Mohan University, Balasore, Odisha 756019, India

Corresponding author: Monalisa Jena (bmonalisa.26@gmail.com)

This work is fully supported by Science and Engineering Research Board (SERB), Department of Science and Technology (DST), Govt. of India through the Teachers' Associateship for Research Excellence (TARE) Fellowship awarded to Prof. Satchidananda Dehuri by vide Sanction no. TAR/2021/000065.

ABSTRACT This work presents an integrated framework for imputation of missing values and prediction of class label of unseen samples by using the best features of rule based inductive decision tree (DT) and Support Vector Machine (SVM) classifier (DT-SVM). In this work, the decision tree is used for imputation of missing values of the datasets containing both categorical and numerical valued attributes. In addition, some of the other popular and simple missing value imputation techniques like drop, mean, median, mode, and k-nearest neighbor (kNN) are used for a comparative analysis. The imputed datasets are then classified using SVM. The performance of the proposed integrated novel framework DT-SVM has been compared with Drop-SVM, Mean-SVM, Median-SVM, Mode-SVM, and kNN-SVM and it is found that DT-SVM outperforms others. Further, a new variant of kNN named it as approximated kNN (A-kNN) has been proposed to overcome some of the shortcomings of canonical kNN while learning from a training set imputed by DT. Unlike canonical kNN, A-kNN does not scan the entire training set. Instead, it processes some of the representative instances from the training dataset to identify the nearest neighbor. The class centroid approach is adopted to find the representative instances of the training set. The effectiveness in term of accuracy as well as computational time of A-kNN is examined by comparing with canonical kNN. It is found that computational time of the proposed A-kNN is drastically reduced as compared to canonical kNN without compromising with the classification accuracy.

INDEX TERMS Classification, data mining, decision tree, kNN classifier, missing values imputation, SVM.

I. INTRODUCTION

Data pre-processing is a process to transform a raw dataset to a useful and understandable dataset and ensure the improvement of performances in tasks like classification, clustering, etc. of data mining, machine learning, pattern recognition, big data analysis, and data science. Classification is one of the fundamental tasks of any predictive mining [1]. The data in real-world databases are of high volume and heterogeneous. Hence the classification process is highly susceptible to missing, inconsistent, noisy data and outliers. To ensure reliability and good quality of data, data pre-processing plays a crucial role in data mining. The handling of missing values is one of the significant tasks of data pre-processing. In real world, organizations employ various data collection methods for their decision making processes. The data collection can be made in different ways like, manually by a person, through on-line questionnaires, from sensors, IoTs, through a survey, social media [2], field study, etc. In many cases there is a chance of getting incorrect, anomalous or incomplete data due to several reasons like purposive or accidental human errors, equipment malfunctioning, incomplete or incorrect observations, error in devices, incorrect measurements, etc. In several cases, organizations cannot disclose their data to everybody as the data may contain many confidential pieces of information. All these situations lead to missing values in the tuples of datasets. Classification is a way of categorizing objects to several classes based on their characteristics. There are several classification techniques in data mining [3]. The classification of a dataset containing missing values may cause inaccuracy and inconsistency in the prediction process. Hence, the missing values in the datasets need to be imputed

The associate editor coordinating the review of this manuscript and approving it for publication was Yonghong Peng.

first, and then the model should be constructed using the complete dataset. The process of filling the values of an attribute, which were missing initially, using certain techniques, procedures, or algorithms is known as missing value imputation.

Now -a- days, it is not an exception to experience databases that have up to 50% of the attribute values missing, making it extremely difficult to handle them using the traditional data processing tools [4], [5]. The simplest procedure adopted to handle missing values is to discard the tuples containing missing values. But, it is not a wise decision as it can lead to significant loss of information. In many cases missing values are not uniformly distributed over all the attributes. So, deleting records with few missing values is not equivalent to deleting records with many missing values. And further, performance will not be as per the expectations as analysis of incomplete data will lead to biasness. So, the most effective way of dealing with the missing values is to impute them. In the last few decades, several approaches have been adopted for imputing missing values [6], [7]. Some of them are mean or median value substitution, nearest neighbor imputation, naive Bayes' missing value treatment, etc. In Bayesian method of imputation, missing values are replaced with the most probable values [8]. In some cases, missing values in the datasets are replaced by a value like one or zero and then the imputed dataset is classified. Rahman and Islam [9], [10] used decision trees and forests to recognize horizontal segments within a dataset, where similarities and correlations within the segments are high. They employed decision tree for missing value imputation. They have employed DT algorithm C4.5 and expectation maximization algorithms [11] to impute missing values of attributes containing categorical and numeric values, respectively. They argued that correlations within the attributes belonging to a segment are higher than that concerning the whole dataset. They obtained higher imputation accuracy for the datasets where attribute correlations were higher. Further, they have observed that correlations within attributes of a horizontal segment are higher than that concerning the entire dataset.

In recent years, Support Vector Machine (SVM) is applied to many classification and regression problems. In addition, it is gaining popularity in many machine learning and pattern recognition fields like face detection [12], cloud classification [13], on-line handwriting recognition [14], bankruptcy prediction [15], spam categorization [16], visual speech recognition [17], text categorization [18], football goal detection [19], object detection, [20] and many more [21], [22]. It was originally proposed by Vapnik and his co-workers for binary classification problems [23]. Later it was extended to non-linear classification, regression, clustering, prediction and many machine learning tasks [24]. SVM is based on the structural risk minimization principle where it not only considers the experimental data but also keeps account of the structural behavior of these data. SVM has better generalization abilities for unseen test data [25].

The kNN model is also one of the simplest and effective machine learning techniques that can be used for classification. However, it suffers from inductive biases and is computationally expensive as it does not have any training phase. Unlike SVM, it does not build any model that learn from the existing data by tuning the parameters. For this reason, it is also known as a lazy learner. The major contributions in this paper are mentioned below:

- In this work, an integrated framework DT-SVM has been proposed for coping missing value imputation and classification by combining the best features of decision tree and SVM. The model uses decision tree for missing value imputation, and SVM for classification of the imputed dataset.
- As the decision tree has the ability to handle the complexity of the datasets, the decision tree algorithm, Classification and Regression Trees (CART) [26] is used for missing value imputation. It is observed that the attribute correlations of the tuples at the leaf nodes of a decision tree are higher than the attribute correlations of the complete dataset [10]. In this paper, the missing value of an attribute of a tuple is imputed by considering these correlations.
- In addition to the integrated framework DT-SVM, a new variant of kNN, i.e. Approximated-kNN (A-kNN) has been proposed. Despite of being widely adopted by several users, the canonical kNN is having a few shortcomings of which the computational time is of major concern. It is found to be computationally expensive. In this paper, an effort has been made to reduce the time complexity of canonical kNN. An extensive comparison between A-kNN and canonical kNN is performed by varying the k values from 3 to 15.
- It is observed that the time complexity of A-kNN approach is better than the canonical kNN. An integrated model DT-A-kNN has also been developed to cope up with the missing values in the dataset. The performance of DT-A-kNN is then compared with DT-kNN in terms of computational time and accuracy. DT-A-kNN is found to be computationally efficient without compromising the accuracy.

The remaining sections are set out as follows: In Section II, related works in this field of research are discussed and in Section III, the background details are described. In Section IV, our integrated framework is explained in details with supporting algorithms and diagrams. Section V highlights the numerical evaluation of our integrated framework, dataset description, numerical results, and computational complexity analysis. Lastly, in Section VI, the work is concluded accompanying with some discussions related to future works.

II. RELATED WORKS

This section explores some of the important works contributed in the field of missing value imputation and classification. Batista and Monard [27] adopted kNN as

missing value imputation technique and compared the performance with four other missing value imputation techniques. The two simple methods are mean and mode imputation methods and the other two techniques are machine learning based imputation using C4.5 and CN2. In their experiment, kNN outperformed the other four imputation approaches. Acuna and Rodriguez [28] analyzed the effect of missing values on the classification accuracy using two classifiers Linear Discriminant Analysis (LDA) and kNN. They observed the effect of mean, median, deletion, and kNN imputation methods on the two classifiers using 12 datasets from UCI repository. In both the cases, the performance is better in comparison to others if imputation is done using kNN imputation method. Pelckmans *et al.* [29] proposed the method of handling missing values using linear SVM and least square SVM (LS-SVM) classifiers. The input attributes in the datasets taken by them have missing values which are missing completely at random (MCAR). They proposed a global method for handling missing values in which instead of imputing the missing values, the expected outcome of the observations are taken into consideration for prediction of class label of unknown sample. Kargupta *et al.* [30] used orthogonal decision trees for classification. Orthogonal trees are the decision trees which are orthogonal to each other. A pair of decision trees are said to be orthogonal to each other if they satisfy the orthogonality condition. They adopted substitution method for imputing missing values. They replaced missing values by one and then classified the dataset. Farhangfar *et al.* [31] stated the impact of missing values on the classification accuracy. They have used five single imputation methods and one multiple imputation method and observed their effect on the classification accuracy for six popular classifiers C4.5, SVM with polynomial and RBF kernels, kNN, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), and naive-Bayes. They used 15 datasets which contain discrete data only and observed that some of the classifiers like Naive-Bayes and C4.5 are missing value resistant, i.e. in the presence of missing data also they can classify the datasets accurately, but in case of the classifiers like SVM and k-nearest-neighbors, performance improves after imputation.

Ghannad *et al.* [8] proposed a selection-fusion approach for classifying incomplete data based on missing value pattern discovery. The subsets of the original dataset were selected based on the missing value pattern and each subset comprises a set of tuples consisting of a set of attributes whose values in the original dataset are not missing. A weak classifier was built for each subset. They were called weak as they were trained using less number of tuples, hence were not efficient and reliable. The result of each weak classifier was combined forming a single strong classifier. This procedure is called fusion. Their approach was designed particularly for datasets having fewer number tuples and a high percentage of missing values and for those datasets for which traditional imputation approaches do not perform effectively. They performed five experiments in which they randomly removed features from the datasets in some experiments and they removed features in some experiments based on the percentage of missing values. They employed MCAR, Missing at Random (MAR), and systematic missing value models. They claimed that their approach was better than the classification methods based on expectation maximization (EM), multiple imputation, and CART. Sajja [32] in her experimental work implied the effect of missing values on classification using heart disease datasets. She has adopted substitution method for missing value imputation of both categorical and numeric attributes. The missing values in categorical attributes were replaced with the mode and numerical values were replaced with mean. In the experiment, one dataset was used as a predictor of the other dataset. The dataset which contained very less, around less than 2% missing values were preprocessed and the model was built from that dataset using the classifiers naive-Bayes, decision tree, and multilayer perceptron. Other datasets were used for testing and their results were compared using the three classifiers. It was found that multilayer perceptron had better accuracy than the other two classifiers.

Poolsawad *et al.* [33] through a case study on heart failure dataset, employed feature selection and classification techniques to handle missing values. For handling missing values, they adopted mean, EM, and kNN imputation techniques among which ANN showed the best result. They employed t-test, entropy and nonlinear gain analysis (NLGA) feature selection techniques for dimensionality reduction where NLGA outperformed the other two. The classification model is built using classifiers multilayer perceptron, decision tree, and radial basis function neural network (RBFN) on the selected features through NLGA and their performance is compared. Through experiment, they showed that ANN for imputation and NLGA with decision tree for feature selection and classification are capable of locating significant variable in the dataset. Silva and Hruschka [34] examined the impact of five nearest-neighbor based imputation algorithms and two simple algorithms mean and majority method on classification problems. They employed MCAR and MAR mechanisms of missing values with four different missing value ratios and made statistical analysis of 3600 different scenarios on the above mentioned imputation algorithms using normalized root mean square error (RMSE) and classification bias as the performance metrics. Through MCAR mechanism, IKNNImpute showed best results for both the metrics and through MAR, SKNNImpute, KNNImpute and EACImpute performed better than others. Ozan *et al.* [35] applied k-NN based missing value imputation technique for a 2-class classification problem of imbalanced dataset with missing values. They provided a solution to the IDA 2016 machine learning prediction challenge in which the Scania trucks dataset is employed which contains missing data from several sensors. They adopted an optimized kNN based classification approach and compared their model with classifiers SVM, random forest, and AdaBoost. Their approach was found to have the least classification cost.

TABLE 1. Comparative study of several works related to missing value imputation and classification.

Author(s)	Year	Imputation Technique	Missingness Mechanism	Classifier Used / Effect Analyzed	Imputation technique(s) Compared with /Analyzed	Performance Metrics	Dataset Details
Batista and Monard	2003	k-NN	MCAR	-	mean, mode, C 4.5 and CN2	mean square error	four data sets from UCI repository (Bupa, Cmc, Pima and Breast)
Acuna and Rodriguez	2004	case deletion, mean, median, and k-NN imputation	MAR	LDA and k-NN	mean, median, deletion and k-NN	cross-validation error	12 datasets from UCI repository
Pelckmans et al.	2005	Global method based on expectation of outcomes	MCAR	Linear SVM, LS-SVM	mean, median	empirical risk	Artificial dataset (Ripley) and Hepatitis dataset from UCI repository
Kargupta et al.	2006	Substitution (Missing values replaced by one)	MCAR	Orthogonal Decision Trees, C 4.5, bagging, random forest	-	classification error, tree complexity	SPECT, NASDAQ; DNA, House of Votes and Contraceptive Method Usage Data from UCI repository
Farhangfar et al.	2008	five single imputation and one multiple imputation technique	MCAR	C4.5, SVM with polynomial and RBF kernels, k-NN, RIPPER and naive-Bayes	comparision made on amount of missing values	classification error, accuracy	15 datasets (with discrete valued attributes) from UCI and KDD repositories
Ghannad et al.	2010	Random removal of features / removal of features based on missing value percentage	MCAR, MAR, systematic missing value model	Selection - fusion approach based on missing value pattern discovery	expectation maximization (EM), multiple imputation, CART	accuracy, cluster validity index (CVI)	08 datasets (07 from UCI repository, one epilepsy dataset from Henry Ford Hospital, Detroit, Michigan)
Sajja	2010	Substitution (categorical attributes replaced with mode and numeric attributes replaced with mean)	MAR	Naive-Bayes, decision tree and multilayer perceptron	Imputation using class frequency, removal of records / attributes	prediction accuracy, precision, recall, classification error	Heart disease datasets Cleveland, Hungary, Switzerland and VA long beach from UCI repository
Poolsawad et al.	2012	ANN	MAR	Multilayer Perceptron, DT, Radial Basis Function Network (RBFP)	EM, kNN, mean imputation	precision, recall, accuracy	Cardiological dataset LIFELAB collected from the outpatient clinic based in England, the University of Hull Medical Centre, UK
Silva and Hruschka	2013	Nearest Neighbor based Imputation	MCAR, MAR	C4.5, k-NN, Multilayer Perceptron, and Naive Bayes	KNNImpute, IKNNImpute, SKNN, K-means, EACImpute, Mean, Majority	prediction accuracy and classification bias	six datasets (Iris, Glass Identification, Yeast, Pen-Digits and Segmentation datasets from UCI repository and a Synthetic dataset)
Ozan et al.	2016	k-NN	MCAR	k-NN, SVM, Random Forest and AdaBoost	-	classification cost	Scania trucks dataset

In TABLE 1, a comparative view of several works related to missing value imputation and classification has been given.

- After making a thorough study on the related works in the field of missing value imputation and classification, it has been observed that the imputation using drop approach is the most inefficient technique as it leads to loss of vital information [36].
- The other statistical approaches used for imputation like mean, median, and mode are better in comparison to drop but their performance is found to be lesser in comparison to machine learning based imputation techniques.
- Earlier, several authors had obtained better results using kNN as per their requirements, but when kNN was compared with other machine learning based imputation techniques like DT, ANN, SVM, etc., it was found to be computationally expensive.
- It has been observed that very few authors have employed decision tree for missing value imputation, though very good results have been obtained using DT [10]. This motivated us to apply decision tree for missing value imputation.
- Further, it is observed that, the classifiers C4.5 and naive-Bayes are missing value tolerant, i.e, they can correctly classify the datasets even if they contain missing values.
- However, the classifiers SVM and kNN have successfully handled the imputed datasets, and the classification performance also increased.

III. BACKGROUND DETAILS

In the present circumstances, data are generated almost everywhere: from sensor networks, through Internet of things, from submarines, during a social survey, opinion polls about any topic, from social media, etc. Most of these real-world applications endure a common problem, missing or unrevealed data [37]. In most of the cases, the missing data may contain several vital informations hidden inside them, hence they must be handled using effective imputation techniques like decision tree. The pre-processed, imputed data must be classified using an effective and efficient classifier like SVM to extract useful patterns and knowledge inside them and to predict the class label of an unknown instance accurately.

A. MISSING VALUES

Missing values are highly unenviable in datasets. An appropriate strategy with proper methodologies and goal must be developed to handle the missing values. While developing the algorithms to handle missing data, not only the final output but also the type of missing data, missing value percentage and the distribution of missing values should be contemplated. There are three patterns of missing data based on the dependency on the attribute itself or other attributes [38]:

1) Missing Completely at Random (MCAR) - The missing value of one attribute A has no dependency on others [39], [40].
2) Missing at Random (MAR) - In this mechanism, the missing value of attribute A is dependent on other attributes. The missing value of A can be imputed using existing values of other attributes [41], [42].

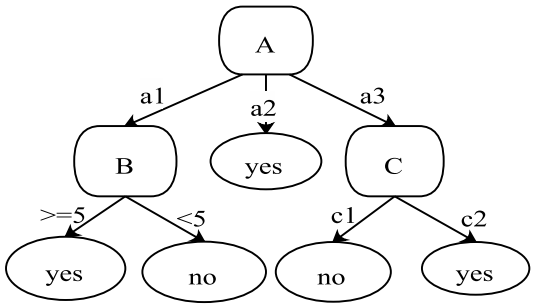


FIGURE 1. A decision tree Example.

3) Missing not at Random (MNAR) - The missing value of A neither depends on itself nor depends on other attributes' existing values. It depends on other missing values. This one is difficult as missing data cannot be imputed using existing attributes' values [43], [44].

B. DECISION TREE (DT)

A DT is a rooted, directed tree-structured classifier. The internal nodes of a DT contain attributes and the leaf nodes contain class labels. The edges in a DT depict either distinct attribute values or conditions. The edges contain the attribute values in case of categorical attributes and conditions in case of numeric attributes [45]. A sample DT is cited in FIGURE 1. It depicts a small dataset containing 3 attributes A, B, C and one class label containing two values $\{yes, no\}$. The intermediate nodes are represented using rounded rectangles and leaf nodes are depicted using ovals. The attributes are of both numeric and categorical in nature. The attribute A is having 3 categorical values $\{a1, a2, a3\}$ and similarly, C is having two values c1 and c2. The numeric attribute B contains values lying between 1 to 10. The values are generalized into two categories, 0-5 (<=5) and 6-10 (>5). The root node is selected using an appropriate attribute selection measure [46]. Because of its structure and self understanding nature, DT is being adopted in several research areas.

C. SUPPORT VECTOR MACHINE (SVM)

SVM is a supervised machine learning-based algorithm used for classification [47]. It is a discriminant technique as it analyzes and solves complex optimization problems. SVM can be used to solve two-class classification problems as well as multi-class classification problems [48]. FIGURE 2 depicts a two-class classification problem. The two classes are separated by hyperplane H. Two equidistant hyperplanes H_1 and H_2 from H help in finding the margin m. These two hyperplanes play a vital role in the classification process. The data points lying on H_1 and H_2 are called support vectors. In SVM, all data points are not vital for classification. The support vectors only contribute to the decision-making process.

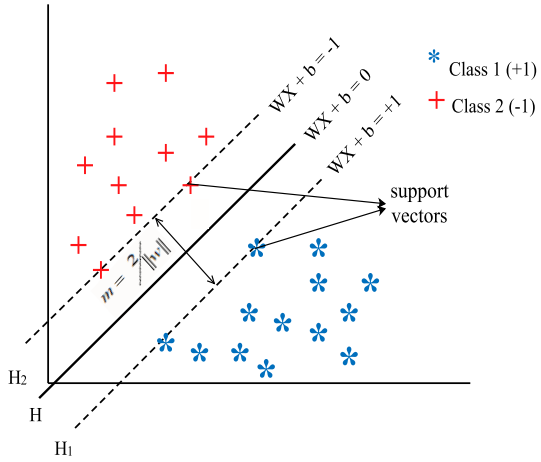


FIGURE 2. Binary classification using linear SVM.

D. K-NEAREST NEIGHBOR (kNN) CLASSIFIER

The kNN classifier is also a supervised machine learning based technique. In kNN, a new instance is assigned to the majority class among its k-nearest neighbors [49], [50]. Consider a set of n pairs (x_1, y_1), (x_2, y_2), …, (x_n, y_n), where x_i training samples are there and y_i class labels. A new instance (x, y) is classified by calculating the distance between itself and its neighbors. The new data point $x'_n \in \{x_1, x_2, \ldots, x_n\}$ is called nearest neighbor of x [50]:

$$min\{d(x_i,\ x)\} = d(x'_n,\ x), \quad i = 1, 2, \ldots, n. \tag{1}$$

For calculating the distance between an unseen data and its neighbors, several distance measures are used in the literature of which Euclidean distance, as shown in equation (2) is the most used one [51].

$$d(x_k, x) = \sqrt{\sum_{i=1}^{k} (x_i - x)^2} \tag{2}$$

The Euclidean distance is suitable for continuous variables. For datasets with categorical and mixed attributes, hamming distance (D_H) as given in equation (3) is preferable.

$$D_H = \sum_{i=1}^{k} \|x_i - x\| \tag{3}$$

IV. OUR INTEGRATED FRAMEWORK

In this paper, two integrated approaches have been proposed to handle the missing values in the datasets. The first one is DT-SVM and the second one is DT-A-kNN. In both cases, DT has been used to impute the missing values followed by the classification process. DT-SVM is found to be the better integrated model in term of accuracy and the DT-A-kNN is better in term of computational time. A comparative analysis of both the frameworks is presented in the result section. The overall approach for integrated frameworks is presented in FIGURE 3.

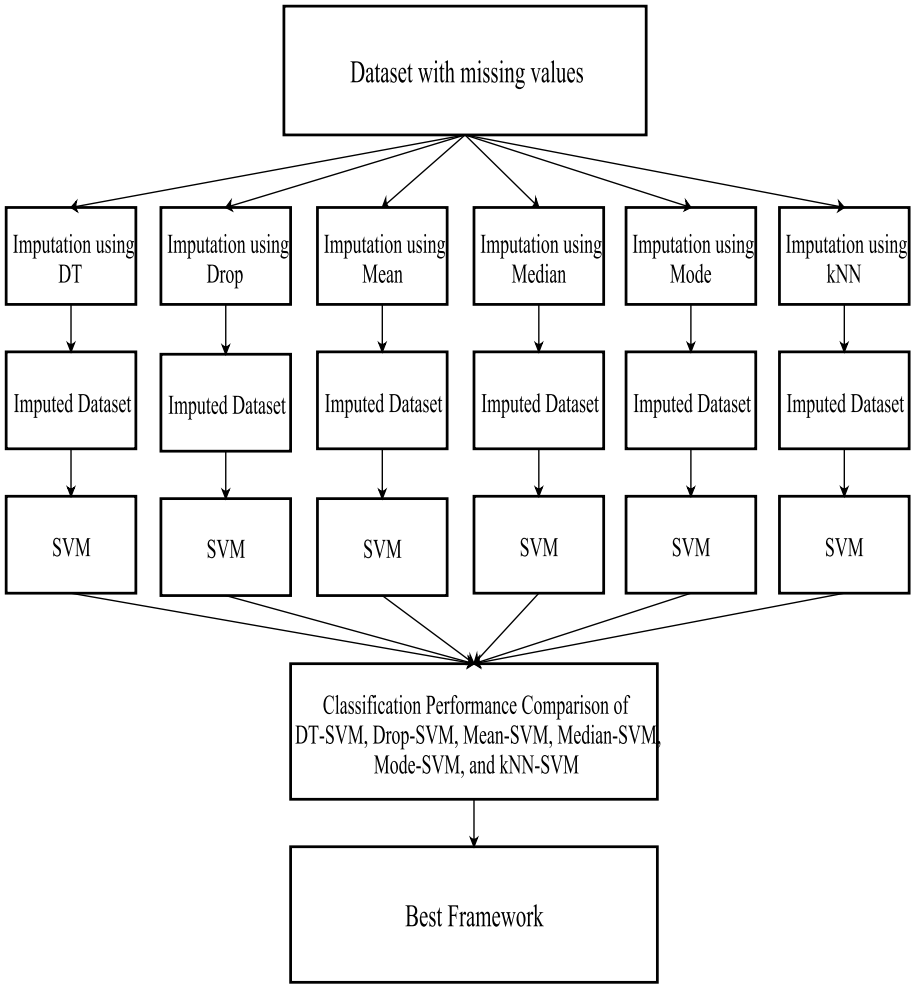


FIGURE 3. Integrated framework for missing value imputation and classification.

A. DT-SVM

An integrated framework for classification has been proposed which inherently computes the missing values in the pre-processing step. The hybrid model combines decision tree with SVM (DT-SVM) for imputation followed by classification. The proposed work DT-SVM is a two-phase process, where the first phase involves the missing value imputation and the second one is concerned with the classification task. In the first phase, the decision tree algorithm CART has been used for missing values imputation and in the second phase, SVM is used for classification.

1) PHASE I: MISSING VALUE IMPUTATION USING DT

Several approaches have been developed to impute the missing values which vary based on the type of data (as discussed in section 3.1) available in the dataset. In this work, the dataset considered is assumed to be MAR. The missing can be deducible by learning the pattern of dependencies with other attributes. The imputation techniques may be classified based on the patterns of missing values such as simple, complex, medium, or blended. A dataset is considered to be simple if every record contains a maximum of one missing value. It is considered as medium if at least two attributes in a record are having missing values and the maximum 50% of the attributes are having missing values. In the complex pattern, records with at least 50% of the attribute values are missing and a maximum of 80% of the attributes may have missing data. In the case of blended pattern, it may have records with each of the above categories [52], [53]. It has been observed that missing value models also depend on the distribution of missing values over the dataset. If each of the attributes is having an equal number of missing values it is said to be uniform. However, if the missing values are not uniformly distributed over the attributes, these are known as the overall category. In this work, blended and overall categories of datasets are considered for experimental purposes.

Decision tree algorithms are usually classified based on the different types of splitting criteria [54]. For example, ID3 uses Information Gain (IG), C4.5 uses gain ratio (GR) and CART algorithm uses gini Index (GI), as the splitting criteria. In the proposed model, CART algorithm has been used for imputing the missing values. Splitting criteria plays a vital role in

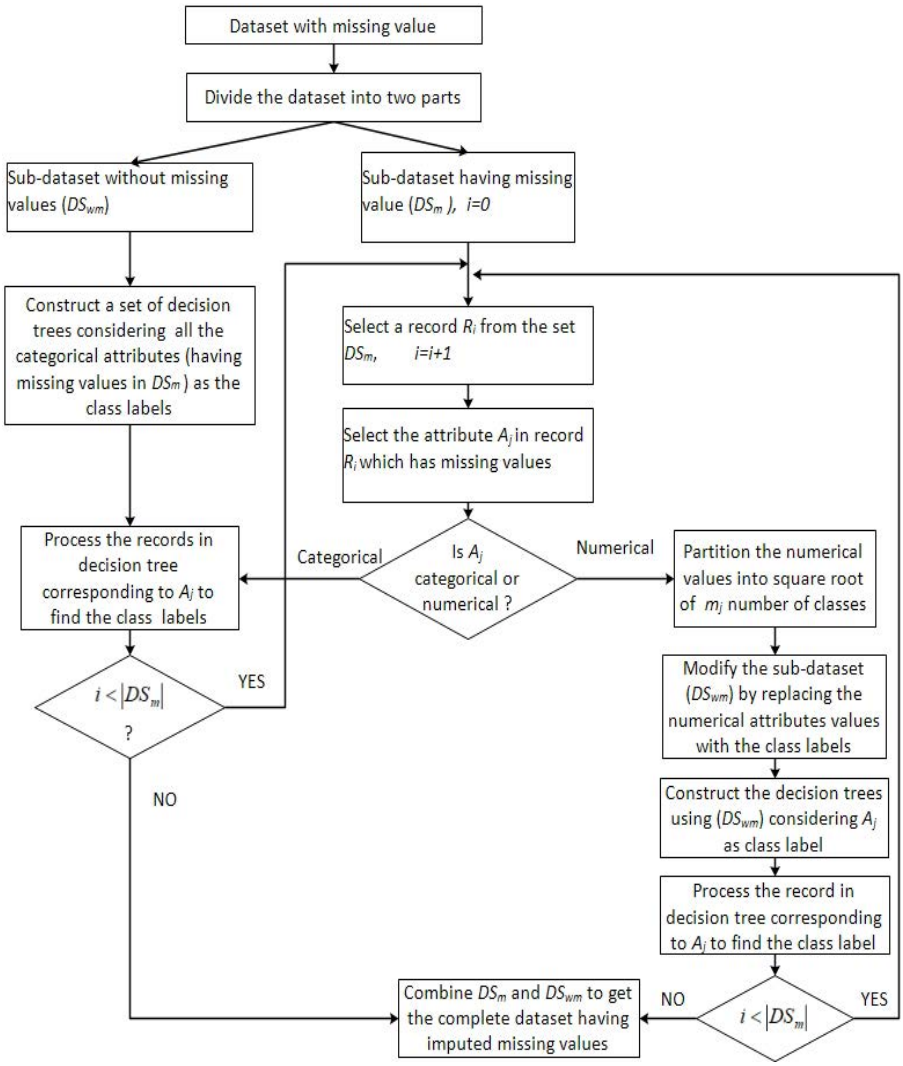


FIGURE 4. Missing value imputation using decision tree.

constructing the decision tree in classification and regression problems. Several splitting criteria like information gain and gini index are usually used to build the decision tree. Information gain is based on the degree of randomness or entropy of the dataset which is measured through the logarithm function. The computational complexity of the classification increases due to the logarithm function used in information gain. In the proposed model, CART algorithm of decision tree is used for missing value imputation. CART uses gini-index as the splitting criterion. The major advantage of the gini-index is that it is easy to implement and also computationally more efficient as compared to other splitting criteria. The gini index results in binary splits and has fewer error rates. Further, the pruning algorithm in CART uses cost complexity as a performance measure [1]. The cost complexity is a function of error rate and number of leaves in the tree. It performs tree pruning to reduce irrelevant branches, if the cost complexity of a subtree is smaller. Gini index can be mathematically expressed as follows:

$$GI(D) = 1 - \sum_{i=1}^{n} P_i^2, \tag{4}$$

where, $P_i = \frac{\|T_i\|}{\|T\|}$, T_i and T are the set of tuples having class i and the set of tuples in the complete dataset, respectively. The mathematical expression for GI for a binary split corresponding to attribute a is as follows:

$$GI_a(D) = \sum_{i=1}^{2} \frac{\|D_i\|}{\|D\|} GI(D_i) \tag{5}$$

where, D_i is one part of the dataset and $GI(D_i)$ is the gini index for the partition D_i. The reduction in impurity after the splitting can be expressed as follows:

$$GI_{red}(a) = GI(D) - GI_a(D) \tag{6}$$

While splitting, all possible partitions for every attributes are tested. The spits with least impurity i.e., minimum $GI_{red}(a)$ is considered as the best split. For each attribute, every feasible splits are considered. The subset with minimum $GI_{red}(a)$ is selected as the best splitting subset. The midpoint procedure is employed to find a possible split point in case of continuous attribute.

The overall picture of missing value imputation is presented in FIGURE 4. The given dataset (DS) is first partitioned into two parts: Part-I, i.e., DS_{wm} consists of all the records with no missing data and part-II, i.e., DS_m consists of tuples with missing values. In this step, all the attributes A_j $(1 \leq j \leq n)$ having missing values are identified, where n is the number of attributes having missing values in DS_m. For each A_j, a decision tree is constructed using CART algorithm taking A_j as the class attribute for the dataset DS_{wm}. If A_j is categorical, then the class labels of the instances are directly reflected at the leaves of the decision tree. Each records in DS_m with missing attributes A_j are then traced through the paths in the decision tree corresponding to A_j. If the attribute A_j is numerical, each value is mapped into one of the $\sqrt{m_j}$ classes, where m is the domain size for attribute A_j. The missing values in the dataset DS_m for A_j is imputed using the constructed decision tree. Likewise, missing value for all the attributes A_j is imputed using the corresponding decision tree. After all the missing values are imputed, the DS_m and DS_{wm} are then merged to get the complete dataset. The flow diagram for the missing value imputation in decision tree is shown in FIGURE 4. The algorithm for missing value imputation using decision tree is presented in Algorithm 1.

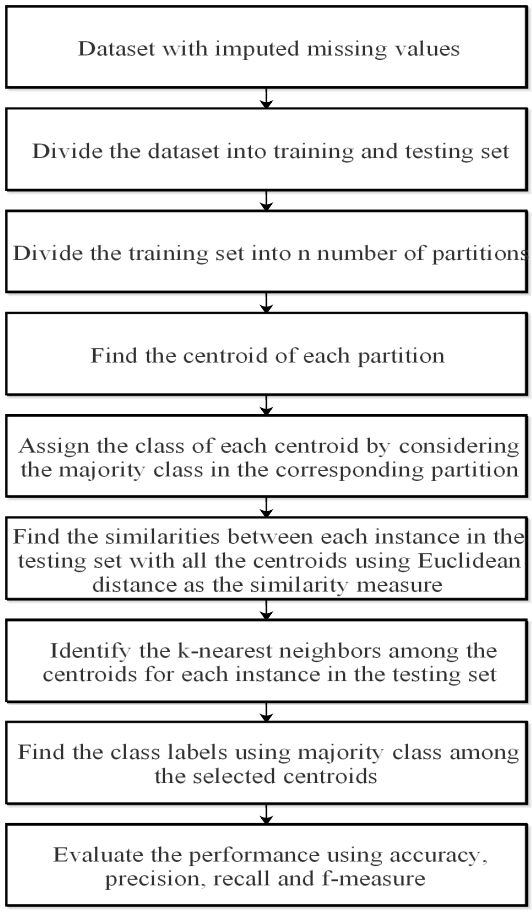


FIGURE 5. Flow diagram for Approximated kNN (A-kNN).

2) PHASE II: CLASSIFICATION USING SVM

In the second phase, the imputed dataset has been classified using SVM. The details of classification approach in the proposed model is explained below.

Given a dataset D consisting of N training data points $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_N, y_N)\}$, where the vector of input space $X_i \in R^d$ and the output vector $Y_i \in \{\pm 1\}$, the linear separating hyperplane classifier can be learned as [55]:

$$H : WX + b = 0 \tag{7}$$

where, W $\{w_1, w_2, w_3, \ldots, w_N\}$ is the weight vector and b is the bias value. In addition, this hyperplane is desired to have the maximum separating margin with respect to the two classes. The margin is nothing but the distance between H_1 and H_2. Here the aim is to find the hyperplane H: WX + b = 0 and two equidistant hyperplanes parallel to it with the condition that there should not be any data point between H_1 and H_2, and the distance between H_1 and H_2 is maximized.

$$H1 : WX + b = +1, \tag{8}$$

$$H2 : WX + b = -1. \tag{9}$$

The points lying on the hyperplanes H_1 and H_2 are called support vectors. Only support vectors participate in the definition of the separating hyperplane, and other examples can be removed and/or moved around as long as they do not cross the planes H_1 and H_2 [56]. The margin between H_1 and H_2 can be computed as: $m = \frac{2}{\|w\|}$. The objective of SVM is either to maximize m or minimize $\|w\|$. All the points should satisfy the following constraints: For each vectorX_i, either

$$WX_i + b \geq 1, \quad for\ Y = +1 \tag{10}$$

or

$$WX_i + b \leq -1, \quad for\ Y = -1 \tag{11}$$

Multiplying Y_i on both sides, of the equations (7) and (8), the combined equation can be written as:

$$Y_i(WX_i + b) \geq 1, \quad \forall 1 \leq i \leq N. \tag{12}$$

So, the constrained optimization problem can be written as:

$$\begin{aligned} &minimize\ Q(W) = \frac{1}{2}\|W\|^2, \\ &subject\ to\ Y_i(WX_i + b) \geq 1, \quad \forall (X_i, Y_i) \in D \end{aligned} \tag{13}$$

By solving the above equation, a pair (W, b) is obtained for which $\|W\|$ is the smallest possible. Then class label of a new data point is predicted using the values of W and b.

The above equation is a primal problem. It is solved using Lagrangian multipliers. So, using Lagrangian multipliers $(\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m)$, the constrained optimization problem in equation (10) can be formulated as [57]:

$$J(W, b, \alpha) = \frac{1}{2} W.W - \sum_{i=1}^{m} \alpha_i \{Y_i(W.X_i + b) - 1\} \quad (14)$$

where, $J(W, b, \alpha)$ is the Lagrange function. As the dot product of a vector with itself is equal to square of its norm, hence, $W.W = \|W\|^2$. As there is difficulty in solving the primal problem, the alternative is formulating the dual problem. Differentiating $J(W, b, \alpha)$ with respect to W and b and setting the results equal to zero, the following value is obtained:

$$\|W\| = \sum_{i=1}^{m} \alpha_i Y_i X_i$$

$$\sum_{i=1}^{m} \alpha_i Y_i = 0 \quad (15)$$

Expanding equation (11) and substituting values of equation (12) in it, the dual is obtained:

$$J_D(W, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j Y_i Y_j X_i.X_j \quad (16)$$

where, α_i are non-negative. So, the dual problem is:

$$maximize\ J_D(W, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j Y_i Y_j X_i.X_j;$$

$$subject\ to \sum_{i=1}^{m} \alpha_i Y_i = 0\ and \alpha \geq 0 \quad (17)$$

B. APPROXIMATED kNN (A-kNN)

The kNN classifier is found to be simple and effective for most of the classification problems. However, kNN is known as a lazy learner as it does not learn a discriminative function from the training data but memorizes the training dataset instead. It only stores the training set instead of adjusting any parameters by making any model which could be useful at testing phase. To identify the nearest neighbor, it searches entire training data for each instance in the testing set. It is expensive in terms of time complexity. In this paper, an effort has been made to improve the kNN algorithm to reduce the time complexity. It is named as approximated kNN (A-kNN) as the class label is predicted based on a set of instances instead of all the instances of the training set. The flow diagram for the proposed A-kNN is shown in FIGURE 5. Unlike canonical kNN, A-kNN does not process entire training set. It processes only small number of instances which are chosen as representative for the entire training set. In the proposed approach, the training set is partitioned into several groups and a centroid is calculated for each group. The centroid acts as the representative of the corresponding group/partition. The similarity is measured between each instance in the testing data with all the centroids instead of all the instances in the training set. In this manner, the time complexity is reduced by a factor of group size (β). The five fold cross validation have been used for better performance analysis. The proposed algorithm for A-kNN is presented in Algorithm 2.

TABLE 2. Characteristics of datasets.

Dataset	No of Instances	No. of Attributes	Attribute type	% of missing value
Diabetes	748	8	Numerical	21.11%
Mammaographics	961	6	Numerical	13.63%
Automobiles	205	26	Mixed	22.44%
Dermatology	366	33	Mixed	15.15%

Algorithm 1 Decision Tree Based Missing Value Imputation

Require: Dataset having missing values at multiple attributes.

Ensure: Complete dataset with imputed missing values.

Initialization: $S_1 = \phi$, $i = 0$, k=no. of attributes
Divide the dataset into two parts; Part1- Sub-dataset having instances without missing values (DS_{wm}), Part2- Sub-dataset having instances with missing values (DS_m);
n=No. of rows in DS_m
for each categorical attributes A_i having missing value **do**
    DT_{A_i}=CreateDecisionTree(DS_{wm}, A_i)
    $S_1 = S_1 \cup DT_{A_i}$
end for
for j=1 to n **do**
    Select the record r_j from DS_m;
    for i=1 to k **do**
        if A_i is the categorical **then**
            Select the decision tree DT_{A_i} from S_1.
            Process the record in DT_{A_i} to find the class level(l);
            $MissingValue(A_i) = l$
        else if A_i is the numerical **then**
            m = Range(A_i)
            Partition the numerical values in A_i in $\sqrt{m}$ number of categories;
            NumericalValue(A_i)=C_a: $C_a \in$ Categories(A_i)
            DT_{A_i}=CreateDecisionTree(DS_{wm}, A_i)
            Process the record r_i in DT_{A_i} to get the missing value;
        end if
    end for
end for
Combine the Sub-dataset DS_{wm} with all the records having imputed missing values;
Return the complete dataset for the classification;

V. NUMERICAL EVALUATION OF INTEGRATED FRAMEWORK

The proposed framework has been implemented using python. The experiment has been performed on a system with an i7 processor and 8 GB RAM. For experimental work,

Algorithm 2 Approximation Approach for Classification Using kNN (A-kNN)

Require: Complete dataset with imputed missing values.
Ensure: Classification result along with other performance parameters.

1: Divide the dataset into training and testing set;
2: Divide the training set into γ ($= n/\beta$) no. of partitions, which is also known as partition factor;
3: Calculate the centroid of each partition;
4: **for** each instance X in testing set **do**
5: Calculate the similarity index (SI) with each centroid (C) using Euclidean distance as follows:

$$SI(X, C) = \sqrt{\sum_{i=1}^{r}(X_i - C_i)^2} \tag{18}$$

where r is the number of attributes of an instance;
6: Sort all the centroids based on decreasing order of the similarity index;
7: Select the top k-centroids (k-nearest neighbors) from the sorted list;
8: Find the majority class label from the selected k-centroids;
9: Assign that class label to the instance X;
10: **end for**
11: Evaluate the performance parameters of the testing dataset.

four real-world datasets are used. They are collected from the UCI machine learning repository and Kaggle repository [63]. A brief description of the datasets are given here.

A. DESCRIPTION OF THE DATASETS

The following four real-world datasets have been considered for experimental purposes as these datasets are widely used in several domains for classification [58] [59]. In these datasets, there are several attributes containing missing values. Hence, the process of imputation becomes a challenging task. Some of these datasets contain mixed types of attributes, i.e., both numerical and categorical, which makes them more suitable to validate the performance of the proposed model. The percentage of missing values in the datasets are listed in TABLE 2. The detailed description of the datasets are presented below:

- Diabetes [60]: This dataset is collected from the repository in kaggle.com, where it is obtained from National institute of Diabetes and digestive diseases. Several constraints were placed to select a few number of instances from a large pool of data e.g., only female member with less than 22 age were selected. This dataset includes eight diagnostic measurement features which are used to classify the patient's condition as diabetes or non-diabetes.
- Mammaographics [61]: Mammaographics dataset consists of all the information about the biopsy report of breast cancer i.e., either malignant or benign. It contains five features and 961 instances.
- Automobile [62]: This dataset contains the specification and characteristics of automobile which includes the cost of insurance along with the risk rating. It has 26 number of attributes and 205 number of instances. It has four class labels that indicate the type of automobile.
- Dermatology [63]: This dataset includes differential diagnosis of erythemato-squamous disease which is one of the major problems in the field of dermatology. It contains 33 features out of which, 22 are histo-pathological features and 12 are the features obtained after the clinical diagnosis of patient skin. It contains both categorical and numerical attributes.

TABLE 3. Performance results of various hybridized algorithms in four real-world datasets.

	Accuracy	Precision	Recall	Fmeasure
Diabetes				
Drop-SVM	0.72	0.333	0.5811	0.423381
Mean-SVM	0.759	0.7	0.5422	0.611077
Median-SVM	0.753	0.692	0.5933	0.63886
Mode-SVM	0.734	0.644	0.5611	0.599699
kNN-SVM	0.821	0.675	0.6733	0.674149
DT-SVM	**0.856**	**0.742**	**0.789**	**0.764779**
DT-A-kNN	0.835	0.715	0.744	0.729212
DT-kNN	0.831	0.713	0.766	0.73855
Mammaographics				
	Accuracy	Precision	Recall	Fmeasure
Drop-SVM	0.732	0.426	0.5933	0.49592
Mean-SVM	0.743	0.419	0.622	0.500707
Median-SVM	0.768	0.573	0.6512	0.609602
Mode-SVM	0.753	0.522	0.644	0.576617
kNN-SVM	0.803	0.622	0.7732	0.689407
DT-SVM	**0.851**	**0.672**	**0.8123**	**0.735519**
DT-A-kNN	0.84	0.663	0.7822	0.717684
DT-kNN	0.838	0.652	0.7833	0.711644
Automobiles				
	Accuracy	Precision	Recall	Fmeasure
Drop-SVM	0.7776	0.446	0.6332	0.523364
Mean-SVM	0.798	0.477	0.66621	0.555947
Median-SVM	0.803	0.512	0.6723	0.581301
Mode-SVM	0.812	0.612	0.6512	0.630992
kNN-SVM	0.822	0.634	0.7321	0.679528
DT-SVM	**0.847**	**0.723**	**0.8111**	**0.76452**
DT-A-kNN	0.837	0.705	0.7811	0.741102
DT-kNN	0.826	0.708	0.7823	0.743298
Dermatology				
	Accuracy	Precision	Recall	Fmeasure
Drop-SVM	0.716	0.623	0.6123	0.617604
Mean-SVM	0.723	0.643	0.6231	0.632894
Median-SVM	0.745	0.653	0.6532	0.6531
Mode-SVM	0.727	0.633	0.6345	0.633749
kNN-SVM	0.827	0.673	0.7543	0.711335
DT-SVM	**0.863**	**0.732**	**0.7987**	**0.763897**
DT-A-kNN	0.834	0.722	0.7612	0.741082
DT-kNN	0.833	0.702	0.7537	0.726932

B. PERFORMANCE METRICS USED FOR EVALUATION

The performance of the proposed integrated model is validated through the confusion matrix. The confusion matrix is an efficient way to represent the outcomes of the

TABLE 4. Performance results for various hybridized models in term of standard deviation.

Diabetes				
	Accuracy	Precision	Recall	Fmeasure
Drop-SVM	0.051	0.073	0.031	0.06
Mean-SVM	0.023	0.033	0.027	0.019
Median-SVM	0.023	0.04	0.017	0.019
Mode-SVM	0.019	0.071	0.027	0.034
kNN-SVM	0.015	0.037	0.016	0.017
DT-SVM	**0.002**	**0.001**	**0.006**	**0.002**
DT-AkNN	0.005	0.002	0.009	0.005
DT-kNN	0.004	0.02	0.014	0.013
Mammographics				
	Accuracy	Precision	Recall	Fmeasure
Drop-SVM	0.012	0.013	0.031	0.014
Mean-SVM	0.006	0.011	0.025	0.009
Median-SVM	0.01	0.014	0.02	0.013
Mode-SVM	0.013	0.017	0.013	0.011
kNN-SVM	0.017	0.008	0.011	0.006
DT-SVM	**0.005**	**0.001**	**0.008**	**0.002**
DT-AkNN	0.006	0.002	0.016	0.006
DT-kNN	0.005	0.004	0.009	0.003
Automobile				
	Accuracy	Precision	Recall	Fmeasure
Drop-SVM	0.008	0.016	0.014	0.015
Mean-SVM	0.005	0.009	0.01	0.014
Median-SVM	0.017	0.019	0.014	0.015
Mode-SVM	0.011	0.008	0.025	0.011
kNN-SVM	0.013	0.007	0.015	0.006
DT-SVM	**0.004**	**0.005**	**0.004**	**0.004**
DT-AkNN	0.005	0.009	0.008	0.007
DT-kNN	0.006	0.007	0.008	0.012
Dermatology				
	Accuracy	Precision	Recall	Fmeasure
Drop-SVM	0.011	0.005	0.02	0.011
Mean-SVM	0.01	0.017	0.03	0.01
Median-SVM	0.02	0.011	0.021	0.014
Mode-SVM	0.01	0.015	0.022	0.016
kNN-SVM	0.006	0.007	0.019	0.013
DT-SVM	**0.004**	**0.001**	**0.002**	**0.004**
DT-AkNN	0.003	0.002	0.005	0.006
DT-kNN	0.005	0.005	0.012	0.006

classification problem. It consists of four parameters as described below:

- True Positive (TP): It represents the number of instances which are correctly classified as positive i.e., the instances belongs to positive class and classified as positive.
- True Negative (TN): It represents the number of instances which are correctly classified as negative i.e., the instances belongs to negative class and classified as negative.
- False Positive (FP): It represents the number of instances which are incorrectly classified as positive i.e., the instances are actually belongs to negative class but classified as positive.
- False Negative (FN): It represents the number of instances which are incorrectly classified as negative i.e., the instances are actually belongs to positive class but classified as negative.

To validate the proposed integrated approach, the following evaluation metrics have been considered. Each evaluation parameter has been measured using the parameters of the confusion matrix.

- Accuracy: The accuracy of the proposed model is computed as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{19}$$

- The precision of the proposed model is computed as:

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

- Recall: The recall of the proposed model is calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

- F-Measure: It is defined as the harmonic mean of precision and recall. Mathematically, it can be defined as:

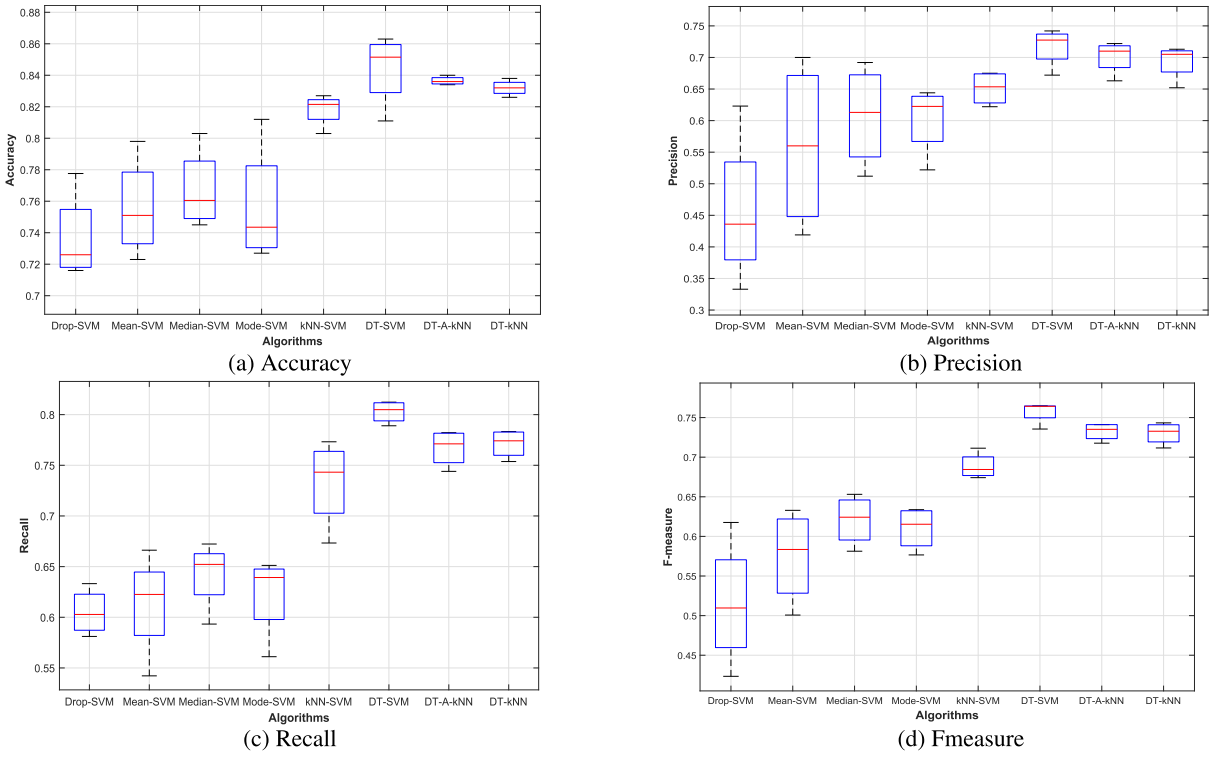
$$Precision = \frac{2 * Precision * Recall}{Precision + Recall} \tag{22}$$

C. EXPERIMENTAL RESULTS AND DISCUSSIONS

Missing values in the datasets pose a challenging task in the classification and/or regression for the machine learning models. Various statistical and machine learning techniques have been adopted for imputing missing values in the dataset. It has been observed that imputation using machine learning techniques like kNN and DT are found to outperform other statistical techniques. Decision tree is found to have better performance results as compared to kNN and other statistical techniques. After the imputation, classification performance has been measured using various parameters like accuracy, precision, recall, and F-measure.

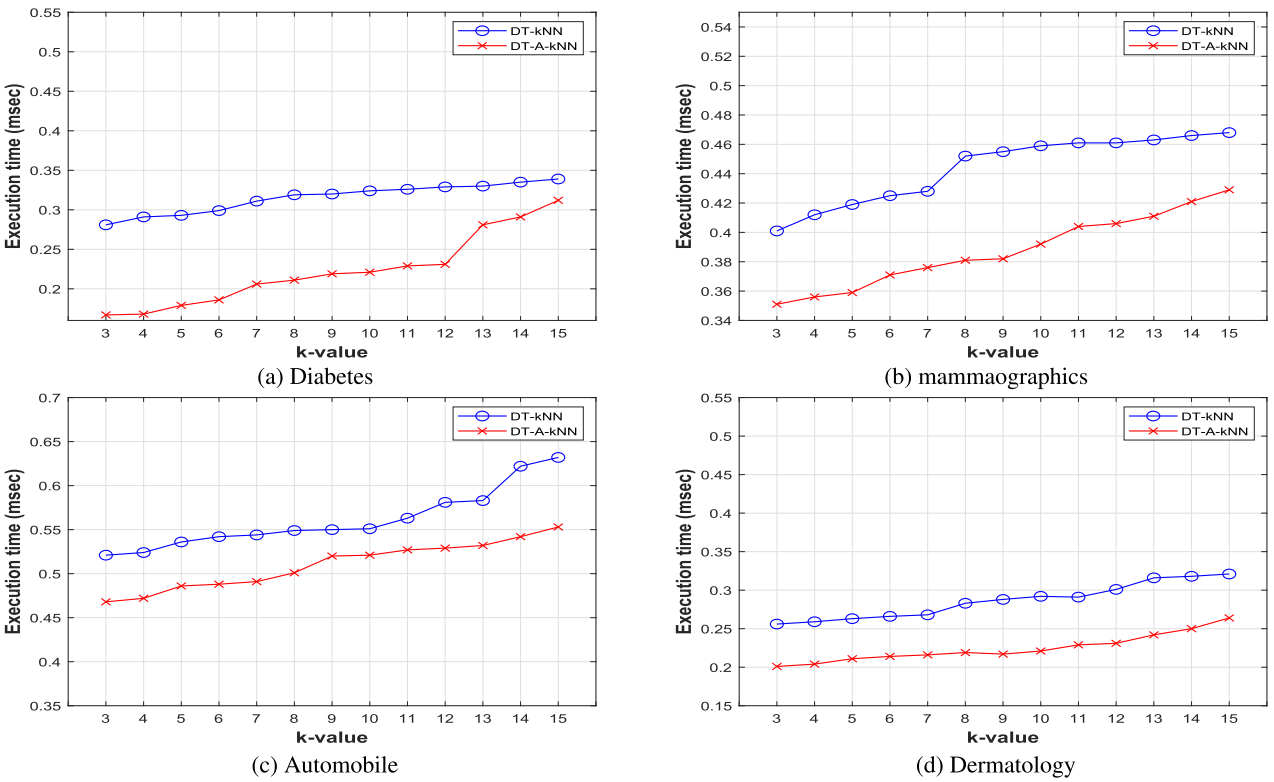
The performance of various hybrid approaches for classification is compared on four real-world datasets as described in the above section. The performance results of the several hybridized approaches are quantified in terms of accuracy, precision, recall, and F-measure, which are listed in TABLE 3. It may be noted that the accuracy of DT-SVM is nearly 85% in all the datasets, which is better than all other hybrid models. The best results in the table are highlighted in bold letters. From TABLE 3, it can also be observed that the proposed DT-A-kNN has a similar performance to DT-SVM, which is found to be better as compared to all other models. Dropping the missing value is not at all a better solution for handling the missing value problem as it can be observed that the Drop-SVM model's performance is not good as compared to other models in terms of all the parameters. In this model, the columns which are having missing values are removed from the dataset, which leads to the loss of some important features that might be useful in predicting the class labels. The machine learning models have better performance as compared to statistical approaches like mean, median, and mode.

The statistical analysis in terms of standard deviation is also performed for all hybridized models. The standard deviation measures the degree of divergence from the mean results.



(a) Accuracy (b) Precision (c) Recall (d) Fmeasure

FIGURE 6. Performance analysis of several hybrid algorithms using real-world datasets.



(a) Diabetes (b) mammaographics (c) Automobile (d) Dermatology

FIGURE 7. Comparative analysis of kNN and A-kNN in terms of computational time.

The standard deviation of performance results is presented in TABLE 4. It is calculated by executing each hybridized model ten times for each dataset. The best result is highlighted in bold letter. The low standard deviation indicates a more

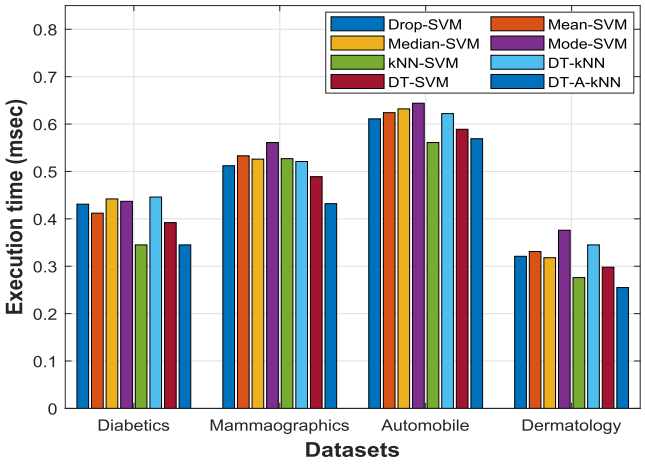


FIGURE 8. Comparative analysis of execution time of all the hybridized models.

consistent performance. It can be observed from TABLE 4 that the standard deviation of DT-SVM is least among all the models, which indicates that it has consistent performance in terms of accuracy, precision, recall, and F-measure.

The comparative analysis in term of accuracy, precision, recall and fmeasure is shown in FIGURE 6a, 6b, 6c and 6d respectively. From FIGURE 6a, it may be noted that DT-SVM has better mean-accuracy as compared to other hybridized algorithms. Among the statistical hybridization, missing value imputation using median (median-SVM) has accuracy better as compared to mean, mode and drop approaches. It can also be observed that all of machine learning approaches like kNN-SVM, DT-SVM, DT-kNN and DT-A-kNN have better accuracy as compared to other statistical models. The hybridized model kNN-SVM has better accuracy as compared to all other statistical approaches. However, its accuracy for classification is less when decision tree is used for missing value imputation.

From FIGURE 6b, it can be observed that the mean precision value of DT-SVM is better as compared to all the hybridized model. Drop-SVM has the least precision value among all the hybridized model. DT-A-kNN has second highest precision value among all the hybrid models as shown in FIGURE 6b. Likewise, the performance comparison in terms of recall and F-measure is shown in FIGURE 6c and 6d, respectively.

In order to improve the computational time, the canonical kNN approach is modified for the classification. Hierarchical partition approach has been used for approximating the kNN algorithm to improve the computational complexity. The time complexity has been improved by a factor of γ which is also known as partition factor of the dataset. It can be measured by dividing the number of instances in training set (n) with the size of each partition (β). For the sake of simplicity, the value of β is fixed to be 10 in our paper. The partition factor (γ) varies from dataset to dataset. The execution time between DT-kNN and DT-A-kNN is compared by varying the k value from 3 to 15. The comparative analysis between DT-kNN and DT-A-kNN for four real-world datasets is shown in FIGURE 7. It can be observed that DT-A-kNN has less execution time as compared to DT-kNN for all the datasets. An extensive comparison of execution time of all the hybrid algorithms has also been performed as shown in FIGURE 8. It can be observed that the overall execution time for DT-A-kNN is better as compared to other hybridized machine learning models.

D. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational time complexity of the canonical kNN is found to be $O(n * d + n * k)$, where n is the number of instances available in the training set, d is the dimension of each instance. The first part i.e., $n * d$ is required to calculate the distance of an instance with all the instances in the training set and the second part is used to return the k indices corresponding to the k-neighboring instances in training set. The time complexity of the proposed A-kNN approach has been reduced by a factor of β, as the distance has been measured with the n/β number of centroids instead of all the instances in the training set. The number of instances has been reduced from n to n/β, where β is the size of each partition. The time complexity of A-kNN is observed to be $O((n/\beta) * d + (n/\beta) * k)$.

VI. CONCLUSION AND FUTURE WORK

The execution model for decision tree is simple and elegant. The tree like structure in decision tree allows the users to develop complex applications in an effective and simple way. The popularity of decision tree has been increasing rapidly over the years in the field of data mining. Missing values in the dataset pose a challenging task to predict the class label in accurate manner. In this work, decision tree and kNN has been used to impute the missing values in the datasets. It has been compared with some of the widely available statistical approaches for missing value imputation techniques like mean, median, mode, and drop. For classification, the machine learning models SVM and A-kNN have been used. The canonical kNN algorithm is simple and effective for classification. However, it is computationally expensive and therefore also widely known as lazy learner. To improve the time complexity, the canonical kNN is optimized and named A-kNN. Through experimental work, it is found that A-kNN has better computational time for classification. It can be concluded from the experiments that DT-SVM is a better hybridized algorithm when performance parameters like accuracy, precision, and recall are the major concerns. In some cases, the hybrid model DT-A-kNN could be the better choice for missing value imputation and classification when computation time is a major concern.

REFERENCES

[1] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[2] S. Das, R. K. Behera, M. Kumar, and S. K. Rath, "Real-time sentiment analysis of Twitter streaming data for stock prediction," *Proc. Comput. Sci.*, vol. 132, pp. 956–964, Jan. 2018.

[3] T. N. Phyu, "Survey of classification techniques in data mining," in *Proc. Int. MultiConf. Eng. Comput. Scientists*, vol. 1, 2009, pp. 18–20.
[4] S. Nikfalazar, C.-H. Yeh, S. Bedingfield, and H. A. Khorshidi, "Missing data imputation using decision trees and fuzzy clustering with iterative learning," *Knowl. Inf. Syst.*, vol. 62, no. 6, pp. 2419–2437, Jun. 2020.
[5] W. Rashid and M. K. Gupta, "A perspective of missing value imputation approaches," in *Advances in Computational Intelligence and Communication Technology*. Singapore: Springer, 2021, pp. 307–315.
[6] C. S. K. Dash, A. Saran, P. Sahoo, S. Dehuri, and S.-B. Cho, "Design of self-adaptive and equilibrium differential evolution optimized radial basis function neural network classifier for imputed database," *Pattern Recognit. Lett.*, vol. 80, pp. 76–83, Sep. 2016.
[7] B. E. T. H. Twala, M. C. Jones, and D. J. Hand, "Good methods for coping with missing data in decision trees," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 950–956, May 2008.
[8] M. Ghannad-Rezaie, H. Soltanian-Zadeh, H. Ying, and M. Dong, "Selection–fusion approach for classification of datasets with missing values," *Pattern Recognit.*, vol. 43, no. 6, pp. 2340–2350, Jun. 2010.
[9] G. Rahman and Z. Islam, "A decision tree-based missing value imputation technique for data pre-processing," in *Proc. 9th Australas. Data Mining Conf.*, vol. 121, 2011, pp. 41–50.
[10] M. G. Rahman and M. Z. Islam, "Missing value imputation using decision trees and decision forests by splitting and merging records: Two novel techniques," *Knowl.-Based Syst.*, vol. 53, pp. 51–65, Nov. 2013.
[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the *EM* algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.
[12] H. Ai, L. Liang, and G. Xu, "Face detection based on template matching and support vector machines," in *Proc. Int. Conf. Image Process.*, vol. 1, 2001, pp. 1006–1009.
[13] M. Azimi-Sadjadi and S. Zekavat, "Cloud classification using support vector machines," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Taking Pulse Planet, Role Remote Sens. Manag. Environ.*, vol. 2, Jul. 2000, pp. 669–671.
[14] A. R. Ahmad, C. Viard-Gaudin, M. Khalid, and E. Poisson, "Online handwriting recognition using support vector machine," in *Proc. IEEE Region 10 Conf. TENCON*, Nov. 2004, pp. 311–314.
[15] A. Fan and M. Palaniswami, "Selecting bankruptcy predictors using a support vector machine approach," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN) Neural Comput., New Challenges Perspect. New Millennium*, vol. 6, Jul. 2000, pp. 354–359.
[16] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999.
[17] M. Gordan, C. Kotropoulos, and I. Pitas, "Visual speech recognition using support vector machines," in *Proc. 14th Int. Conf. Digit. Signal Process. (DSP)*, vol. 2, 2002, pp. 1093–1096.
[18] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, 1998, pp. 137–142.
[19] N. Ancona, G. Cicirelli, A. Branca, and A. Distante, "Goal detection in football by using support vector machines for classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 1, 2001, pp. 611–616.
[20] N. Ancona, G. Cicirelli, E. Stella, and A. Distante, "Object detection in images: Run-time complexity and parameter selection of support vector machines," in *Proc. Object Recognit. Supported User Interact. Service robots*, vol. 2, 2002, pp. 426–429.
[21] R. K. Behera, K. S. Sahoo, D. Naik, S. K. Rath, and B. Sahoo, "Structural mining for link prediction using various machine learning algorithms," *Int. J. Social Ecol. Sustain. Develop.*, vol. 12, no. 3, pp. 66–78, Jul. 2021.
[22] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data," *Inf. Process. Manage.*, vol. 58, no. 1, Jan. 2021, Art. no. 102435.
[23] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.
[24] C. S. K. Dash, P. Sahoo, S. Dehuri, and S.-B. Cho, "An empirical analysis of evolved radial basis function networks and support vector machines with mixture of kernels," *Int. J. Artif. Intell. Tools*, vol. 24, no. 4, Aug. 2015, Art. no. 1550013.
[25] H. Byun and S.-W. Lee, "A survey on pattern recognition applications of support vector machines," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 17, no. 3, pp. 459–486, 2003.
[26] L. Breiman, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.
[27] G. E. A. P. A. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Appl. Artif. Intell.*, vol. 17, nos. 5–6, pp. 519–533, 2003.
[28] E. Acurna and C. Rodriguez, "The treatment of missing values and its effect on classifier accuracy," in *Proc. Meeting Int. Fed. Classification Soc. (IFCS)*, 2004, pp. 639–647.
[29] K. Pelckmans, J. D. Brabanter, J. A. Suykens, and B. De Moor, "Handling missing values in support vector machine classifiers," *Neural Netw.*, vol. 18, nos. 5–6, pp. 684–692, Aug. 2005.
[30] K. Kargupta, B.-H. Park, and H. Dutta, "Orthogonal decision trees," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1028–1042, Aug. 2006.
[31] A. Farhangfar, L. Kurgan, and J. Dy, "Impact of imputation of missing values on classification error for discrete data," *Pattern Recognit.*, vol. 41, no. 12, pp. 3692–3705, 2008.
[32] S. Sajja, "Data mining of medical datasets with missing attributes from different sources," Ph.D. thesis, Youngstown State Univ., Youngstown, OH, USA, 2010.
[33] N. Poolsawad, L. Moore, C. Kambhampati, and J. G. F. Cleland, "Handling missing values in data mining—A case study of heart failure dataset," in *Proc. 9th Int. Conf. Fuzzy Syst. Knowl. Discovery*, May 2012, pp. 2934–2938.
[34] J. D. A. Silva and E. R. Hruschka, "An experimental study on the use of nearest neighbor-based imputation algorithms for classification tasks," *Data Knowl. Eng.*, vol. 84, pp. 47–58, Mar. 2013.
[35] E. C. Ozan, E. Riabchenko, S. Kiranyaz, and M. Gabbouj, "An optimized k-NN approach for classification on imbalanced datasets with missing data," in *Proc. Int. Symp. Intell. Data Anal.* Cham, Switzerland: Springer, 2016, pp. 387–392.
[36] X. Zhu, J. Wang, B. Sun, C. Ren, T. Yang, and J. Ding, "An efficient ensemble method for missing value imputation in microarray gene expression data," *BMC Bioinf.*, vol. 22, no. 1, pp. 1–25, Dec. 2021.
[37] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: A review," *Neural Comput. Appl.*, vol. 19, no. 2, pp. 263–282, 2010.
[38] R. J. Little and D. B. Rubin, *Statistical Analysis With Missing Data*, vol. 793. Hoboken, NJ, USA: Wiley, 2019.
[39] H. Rosado-Galindo and S. Dávila-Padilla, "Tree-based missing value imputation using feature selection," *J. Data Sci.*, vol. 18, no. 4, pp. 606–631, Jan. 2021.
[40] E. Van Wolputte and H. Blockeel, "Missing value imputation with MERCS: A faster alternative to missforest," in *Proc. Int. Conf. Discovery Sci.* Cham, Switzerland: Springer, 2020, pp. 502–516.
[41] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm," *Inf. Sci.*, vol. 233, pp. 25–35, Jun. 2013.
[42] B. Walczak and D. L. Massart, "Dealing with missing data: Part I," *Chemometrics Intell. Lab. Syst.*, vol. 58, no. 1, pp. 15–27, 2001.
[43] C.-H. Liu, C.-F. Tsai, K.-L. Sue, and M.-W. Huang, "The feature selection effect on missing value imputation of medical datasets," *Appl. Sci.*, vol. 10, no. 7, p. 2344, Mar. 2020.
[44] H. Huang, H. Wang, and M. Sun, "Incomplete data classification with view-based decision tree," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106437.
[45] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers—A survey," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 476–487, Nov. 2005.
[46] J. R. Quinlan, "Decision trees and decision-making," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 339–346, Mar./Apr. 1990.
[47] T.-J. Hsieh and W.-C. Yeh, "Knowledge discovery employing grid scheme least squares support vector machines based on orthogonal design bee colony algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 5, pp. 1198–1212, Oct. 2011.
[48] P. Sahoo, A. K. Behera, M. K. Pandia, C. S. K. Dash, and S. Dehuri, "On the study of GRBF and polynomial kernel based support vector machine in web logs," in *Proc. 1st Int. Conf. Emerg. Trends Appl. Comput. Sci.*, Sep. 2013, pp. 1–5.
[49] M. J. Islam, Q. M. J. Wu, M. Ahmadi, and M. A. Sid-Ahmed, "Investigating the performance of classifiers and *K*-nearest neighbor classifiers," in *Proc. Int. Conf. Converg. Inf. Technol. (ICCIT)*, Nov. 2007, pp. 1541–1546.
[50] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[51] S. Tan, "An effective refinement strategy for KNN text classifier," *Expert Syst. Appl.*, vol. 30, no. 2, pp. 290–298, Feb. 2006.
[52] J. Scheffer, "Dealing with missing data," Massey Univ., Palmerston North, New Zealand, Tech. Rep. 10179/4355, 2002.
[53] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen, "Methods for imputation of missing values in air quality data sets," *Atmos. Environ.*, vol. 38, no. 18, pp. 2895–2907, Jun. 2004.
[54] S. Hwang, H. G. Yeo, and J.-S. Hong, "A new splitting criterion for better interpretable trees," *IEEE Access*, vol. 8, pp. 62762–62774, 2020.
[55] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[56] S. Balasundaram and N. Kapil, "Application of Lagrangian twin support vector machines for classification," in *Proc. 2nd Int. Conf. Mach. Learn. Comput.*, 2010, pp. 193–197.
[57] M. E. Mavroforakis and S. Theodoridis, "A geometric approach to support vector machine (SVM) classification," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 671–682, May 2006.
[58] A. M. Bagirov, A. M. Rubinov, N. V. Soukhoroukova, and J. Yearwood, "Unsupervised and supervised data classification via nonsmooth and global optimization," *Top*, vol. 11, no. 1, pp. 1–75, Jun. 2003.
[59] W. Duch, R. Setiono, and J. M. Zurada, "Computational intelligence methods for rule-based data understanding," *Proc. IEEE*, vol. 92, no. 5, pp. 771–805, May 2004.
[60] P. Royston, "Multiple imputation of missing values," *Stata J.*, vol. 4, no. 3, pp. 227–241, 2004.
[61] M. Elter, R. Schulz-Wendtland, and T. Wittenberg, "The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process," *Med. Phys.*, vol. 34, no. 11, pp. 4164–4172, Oct. 2007.
[62] Y. Wang, "A new approach to fitting linear models in high dimensional spaces," Ph.D. thesis, Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 2000.
[63] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2019. [Online]. Available: http://archive.ics.uci.edu/ml

MONALISA JENA received the M.Tech. degree in computer science from SOA University, Bhubaneswar, Odisha, India. She is currently pursuing the Ph.D. degree with Fakir Mohan University, Balasore, Odisha, with specialization in data mining and soft computing under the supervision of Prof. Satchidananda Dehuri. She has been working as an Assistant Professor with the Department of Computer Science, Fakir Mohan University, since 2015. Prior to that, she has worked with Government Polytechnic, Balasore, as a Lecturer in computer application under the Skill Development in Technical Education Department, Odisha, from November 2013 to December 2015, appointed by the Odisha Public Service Commission (OPSC). She qualified UGC-National Eligibility Test for lectureship, in 2012; and the Graduate Aptitude Test in Engineering (GATE), in 2010 and 2012. She has published several research papers in reputed journals and conferences. Her research interests include big data analysis, machine learning, social networking, and wireless mesh networks.

SATCHIDANANDA DEHURI received the M.Tech. and Ph.D. degrees in computer science from Utkal University, Vani Vihar, Odisha, in 2001 and 2006, respectively. He has been working as a Professor with the Department of Computer Science (Erstwhile Information and Communication Technology), Fakir Mohan University, Balasore, Odisha, India, since 2013. Prior to this appointment, for a short stint (i.e., from October 2012 to May 2014) he was an Associate Professor with the Department of Systems Engineering, Ajou University, South Korea. He visited as a BOYSCAST Fellow with the Soft Computing Laboratory, Yonsei University, Seoul, South Korea, under the BOYSCAST Fellowship Program of DST, Government of India, in 2008. In 2021, he received the Teachers Associateship and Research Excellence (TARE) Fellowship from SERB, DST, Government of India, for three years to carry out intensive research on higher order neural networks for big data analysis at the Host Institute, the ISI Kolkata and Parent Institute, and Fakir Mohan University. He was with the Center for Theoretical Studies, Indian Institute of Technology Kharagpur, as a Visiting Scholar, in 2002. From May 2006 to June 2006, he was a Visiting Scientist with the Center for Soft Computing Research, Indian Statistical Institute, Kolkata. He has already published about 200 research papers in reputed journals and referred conferences, has published five text books for undergraduate and post graduate students, and edited more than ten books of contemporary relevance. Under his direct supervision, 19 Ph.D. scholars have been successfully awarded and one more is pursuing his Ph.D. work. He has successfully guided two postdoctoral scholars during the stay at Ajou University as an Associate Professor with the Department of System Engineering for two years. He has completed three different research projects obtained from DST, UGC, and DRDO. His H-index as per Google Scholar is more than 25. As a part of academic collaboration, he has visited Ireland, New Zealand, Hong Kong, France, and Nepal. His research interests include evolutionary computation, neural networks, pattern recognition, data warehousing and mining, object oriented programming, and its applications and bioinformatics. In 2010, he received the Young Scientist Award in Engineering and Technology for the year 2008 from Odisha Vigyan Academy, the Department of Science and Technology, Government of Odisha.