Computer Science and Information Systems 20(3):949–976 https://doi.org/10.2298/CSIS211030001S

Class Probability Distribution Based Maximum Entropy Model for Classification of Datasets with Sparse Instances

Saravanan Arumugam*, Anandhi Damotharan, and Srividya Marudhachalam

Department of Computing, Coimbatore Institute of Technology
Coimbatore, Tamil Nadu, India – 641014
a.saravanan21@gmail.com
anandhi.cit@gmail.com
msrividya2013@gmail.com

Abstract. Due to the digital revolution, the amount of data to be processed is growing every day. One of the more common functions used to process these data is classification. However, the results obtained by most existing classifiers are not satisfactory, as they often depend on the number and type of attributes within the datasets. In this paper, a maximum entropy model based on class probability distribution is proposed for classifying data in sparse datasets with fewer attributes and instances. Moreover, a new idea of using Lagrange multipliers is suggested for estimating class probabilities in the process of class label prediction. Experimental analysis indicates that the proposed model has an average accuracy of 89.9% and 86.93% with 17 and 36 datasets. Besides, statistical analysis of the results indicates that the proposed model offers greater classification accuracy for over 50% of datasets with fewer attributes and instances than other competitors.

Keywords: classification, fewer attributes and instances, Lagrange multipliers, class probability distribution, relative gain, maximum entropy.

1. Introduction

In this digital era, data mining has become an inevitable technique and a milestone in technological development. It is applied to a wide range of historical data to extract useful information that helps to make decisions effectively [1]. It covers other important areas like machine learning, statistics and the database management system. It is extremely influential and even changed the perspective of handling business. Although it was originally used to develop the business, later it seems to be an inseparable technique in almost every area [2]. It focuses on extracting various piece of knowledge from the vast amount of data. This can be achieved by several data mining functions such as classification, association rule mining, prediction, outlier and cluster analysis and pattern recognition. Nevertheless, classification and prediction have become the two major pillars of data mining [3].

[*] Corresponding author

Classification and prediction are the most common methods that researchers use in all areas to find solutions to various problems. A few of the domain applications where classification and prediction are used frequently include the educational field (students' performance classification, result prediction) [4], bank and financial sectors (customers classification based on their credit risk, fraud detection) [5], health care industries (diagnosing the disease based on the past data containing symptoms) [6], agricultural field (analysing soil nutrients and crop prediction) [7], retail industries (customer churn and sales prediction) [8], classifying spam or junk emails [9], weather forecasting and rainfall prediction [10], predicting current behaviour by analyzing the human activities [11], classifying customer segment [12], classifying attack traffic from normal network traffic [13], software defect prediction [14] and even more.

Generally, classification outcomes are often influenced by the quality of input data. Pre-processing of input data is carried out before applying the classification model to improve its prediction accuracy. The data can be preprocessed by removing missing data and normalizing the attribute values along with the feature selection process [15]. Feature selection aims at selecting the relevant study-related attributes for the target class. In general, classification models can be based on machine learning or statistical models [16]. The machine learning based models include decision trees (DT), random forest (RF), artificial neural networks (ANN), k nearest neighbour (KNN), cased based reasoning (CBR), support vector machines (SVM), AdaBoost, Stochastic gradient descent (SGD), other ensemble and boosting classifiers. The statistical model includes linear and logistical regression and naïve Bayesian classifiers. These models are currently available and many more new models were also suggested by the various researchers. However, most of these existing models are an extension of specific conventional models designed for specific applications.

The type of data to be classified, such as categorical data, real or integer-valued data, typically affects the performance of the classification model. Some algorithms are only suitable for certain types of data like logistic regression cannot manage huge categorical data. In addition to data types, the size of attributes and instances plays a crucial role in the accuracy of classification. If models are not chosen based on the analysis of the adequacy and applicability of the specific characteristics of the datasets, there is a greater possibility of classification error. Moreover, some classifiers classify the data with appropriate results, but with greater computational complexity. Consequently, classification models should be constructed by considering various other characteristics of the underlying datasets.

This paper provides a simple statistical classification model that is appropriate for datasets with fewer attributes and instances. It utilizes the novel idea of using Lagrange multipliers on the class probabilities that is suitable for the classification of samples in small datasets. The proposed class probability distribution based maximum entropy classifier works as follows. To begin with, the dataset is subjected to feature selection and data pre-processing to improve the dataset's data quality and classification accuracy. During the training phase, the datasets are categorized according to the labels assigned to the target classes. Then, for each chosen attribute, the average class relative distance is estimated for the training samples, from which the attribute relative gain is calculated for the given test sample. The Lagrange multipliers are applied and evaluated to assess the class probabilities of the attribute by maximizing the entropy. Finally, the class probabilities of each attribute are aggregated to predict the class label for the given test instance. An extensive experimental analysis is also made to examine the performance and effectiveness of the proposed model.

The organization of the paper is as follows. Section 2 presents the works from the literature that are related to the proposed study. Section 3 discusses the study background. Section 4 describes the proposed class probability distribution based maximum entropy model for classification. The overall framework, algorithm pseudocode and working procedure with an illustration are presented in sub-sections. Section 5 presents the various experimental analysis, results obtained for the proposed model and research findings from the statistical analysis. Finally, the paper is concluded by listing out the scope for future enhancements.

2. Related Works

Owing to the widespread use of data mining and other machine learning techniques, classification models are evolving day by day. Several classification models and their variations were proposed in the literature by the researchers. For easy understanding, the existing classifiers that are related to the study are clustered under two groups. The first category is the standard classifiers that are significant and widely used in classification problems and the second category is the new existing state-of-the-art classifiers that are developed recently yet to be researched further. These categories are presented in this section.

2.1. Standard Classifiers

To properly categorize unlabeled data, the majority of supervised learning algorithms use statistical analysis of the training set in one way or another. Among these classifiers, KNN, Naïve Bayes (NB), Logistic regression (LR) and Decision trees use statistical inference to classify the data. A univariate location estimator, termed proximity based KNN classifier was a simple classic classification model proposed for estimating regression curve. In this model, the classification results of the given test data point are the closest point among a given set of data points [17]. In general, KNN classifier is computationally inefficient and challenging to determine the right k value, even though it is most frequently employed in numerous applications with numerous variants [18].

Naïve Bayes classifiers are probabilistic model that applies the Bayes theorem to predict the class probability of the given instance [19]. The main drawback of this model is that the model treats each attribute independently and so cannot identify the relationship between the attributes. Nonetheless, the model is still frequently utilized in various applications because of its efficient performance [20]. Logistic regression is another statistical model that applies a logistic function for modelling the dependent variable using independent variables [21]. The model is sensitive to overfitting and cannot be used for non-linear problems or when the number of instances is less than the number of attributes. Decision trees are another type of classification model that makes use of the gain of an attribute at each precedent node [22]. Numerous types of trees exist such as ID3, CART and C4.5 among which C4.5 [23] offers better results. However, the decision trees anticipate poor results with small datasets and cause overfitting.

For the datasets with high dimensions or when the number of attributes is greater than the number of instances, SVM offers improved results and so it is widely popular among various fields [24], [25]. However, the SVM is not suitable for non-linear problems. Alternatively, Sparse Representation based Classification (SRC) [26] is another classifier model that offers better performance. However, SRC is more suitable for multimedia datasets involving image, audio and video data.

Ensemble classifiers are another milestone in the classification model. It utilizes two or more classifiers to classify the data and the results are combined using schemes such as majority voting or weighting technique [27]. The ensemble learners can use various techniques such as boosting, bagging or stacking to convert weak learners to strong learners. Algorithms such as AdaBoost (AB) and Gradient Boosting use boosting to reduce the bias between various models used [28]. Random forest algorithm employs bootstrap aggregation (bagging) to reduce the variance [29] or stacking [30] to increase the prediction. As gradient boosting interprets the boosting as an optimization, Stochastic Gradient Boosting Decision Trees (GBDT) apply regression to the gradient boosting algorithm [31]. Though ensemble learners offer better accuracy it is less widely used due to their increased time complexity.

Not only machine learning techniques but artificial intelligence models were also incorporated for the classification of test instances. Artificial neural network (ANN) is widely adapted in classification inspired by the neural networks in the animal brain. These models are specifically designed to recognize patterns [32]. Similarly, Deep Learning (DL), a model that mimics the working of the human brain in recognizing patterns was proposed specifically for making decisions [33], [34]. Extreme Learning Machine (ELM), a feedforward neural network utilizes single-layer feed-forward neural networks [35]. These models offer better classification accuracy in minimum time than other traditional neural networks such as backpropagation. Still, the models are the least widely used since SVM outperforms them in various cases.

Several analyses were made in the literature to examine the performance of the conventional classifiers. An analysis was made using several machine learning classifiers such as NB, Bayesian networks, J48, RF, multilayer perceptron (MLP), and LR to identify the better classifiers [36]. This study with the credit risk dataset indicates that the RF produce improved performance than others. Similar analysis was carried out for SVM, KNN, Gradient boosting, decision tree, RF and LR on diabetes datasets [37]. The results indicate that RF outperforms the other 6 classification algorithms with many of the evaluation metrics. An analysis of the performance of the classification algorithms such as ELM, SRC, DL, GBDT, SVM, RF, C4.5, KNN, LR, AB, and NB on various datasets was evaluated. The result outcomes are surprising that GBDT offers better results across various datasets than SVM and RF [38]. Most of the classification algorithms or the comparative studies found in the literature are specific to a particular application. Though the models were proved to be effective, the results may not be same for all the applications. Thus, lead to performance degradation for other applications or different datasets having different attribute types for the same applications [39].

2.2. State-of-the-Art Classifiers

Several researchers had contributed more on classification problems with various new probabilistic models for different data types. The use of various probabilistic models such as multinomial Bernoulli assuming naive Bayes [40], the combination of Expectation-Maximization (EM) and NB classifier [41], and generative/discriminative model [42] were found in the literature. The detailed study of these probabilistic models shows the performance improvement over text datasets than other data types. The use of conditional random fields based on a probabilistic model was proposed to segment and label the data. However, it was only evaluated sequence data and evidenced to have improved classification accuracy [43].

Many instance based classifiers attained a notable position in the literature. Data Gravitation based Classification (DGC) makes the comparison between the data gravitation and distinct classes for classifying the given input record [44]. This work was extended by adding weights to the data gravitation (DGC+) [45]. Despite the improved accuracy, the models undergo high computational complexity. Another classification model that computes the average weighted pattern score (AWPS) to classify the given data using attribute rank based feature selection was proposed [46]. The comprehensive analysis of the study indicates that the model is suitable for imbalanced datasets and yet the results are not accurate for low dimensional space.

An instant based classifier termed attribute value frequency based instance weighted naive Bayes (AVFWNB) was proposed [47]. In this model, the weights are assigned for the training sets that offer good results than traditional NB. Similarly, a simple model that is a variation of NB called correlation based attribute weighted naive Bayes (CAWNB) was proposed. The model aims at assigning weights for the attributes based on the dependency between the attribute and the class [48]. Moreover, the weights are verified using sigmoid transformation. The results of CAWNB proved to be effective with improved classification accuracy than NB. Inspired by AVFWNB and CAWNB models, a unique model that assigns weight for instances and attributes was proposed recently. This model utilizes two approaches eager learners (AIWNB^{E}) aa lazy learners (AIWNB^{L}) for implementing instance weights [49]. The performance of these classifiers highly depends on the how accurately the weights are assigned to the instances and attributes.

Discriminatively weighted naive Bayes (DWNB) and eager learning approach was opposed that iteratively re-assigns the weights by computing the conditional probability loss. Though the model seems have effective performance in terms of accuracy, the model needs more iterations to improve efficiency in assigning weights [50]. A model that computes the weights for the instances and attributes collaboratively was proposed. The model utilizes posterior probability loss to compute the weights and is termed as collaboratively weighted naive Bayes (CWNB) [51]. An instance weighted hidden naive Bayes (IWHNB) was proposed that integrates the instance weight with a hidden naive Bayes model for computing probabilities [52]. For all these weight assignment based classifiers, the optimization in assigning weights to the instances and attributes is to be incorporated for ensuring effective performance. Moreover, in all these methods, the authors show improved performance than existing models, yet the accuracy still needs improvement. The summary of the significant existing classifiers is presented in Table 1.

Table 1. Summary of Existing State-of-the-Art Classifiers

Model	Authors	Approach	Merits	Drawback
Data gravitation classification (DGC)	Peng et al., (2009) [44]	Classifies the instances by comparing the data gravitation between the different data classes	Simple and effective to implement	Reduction in accuracy when the points are away from centroid and class borders
Extended data gravitation classification (DCG+)	Cano et al., (2013) [45]	Assigns matrix of weights for attributes based on its significance in each class	Improved accuracy	High computational complexity
Discriminatively weighted naive Bayes (DWNB)	Jiang et al., (2012) [50]	Iteratively the weights are re-assigned based on conditional probability loss	Eager learning approach	Needs more iterations
Average weighted pattern score based classification (AWPS)	Sathya Bama and Saravanan., (2019) [46]	Feature selection with and classification using average weighted pattern score	Simple and outperforms many existing classifiers	Not accurate for low dimensional datasets
Correlation-based attribute weighted naive Bayes (CAWNB)	Jiang et al., (2018) [48]	Attributes weight are assigned by computing the difference between attribute-class correlation and attribute-attribute redundancy	Better than NB and simple to implement	Need more time to compute similarity between the attributes in high dimensional space
Attribute value frequency-based instance weighted naive Bayes (AVFWNB)	Xu et al., (2019) [47]	Instance weights are assigned based on attribute value frequency and attribute value number	Better than NB and simple to implement	Low performance on datasets with high dimensions
Attribute and instance weighted naive Bayes (AIWNB)	Zhang et al., (2021) [49]	Weights for instance is assigned based on the distribution of the instance	Applies both lazy and eager approach for assigning weights	Accuracy depends on weight assignment
Collaboratively weighted naive Bayes (CWNB)	Zhang et al., (2021) [51]	Optimal weights for the instance are computed by maximizing conditional log-likelihood with prior and conditional probabilities	More accurate than Naïve Bayes and other similar models	High computational complexity in assigning weights for the instances
Instance weighted hidden naive Bayes (IWHNB)	Yu et al., (2021) [52]	Integrates the instance weighting with improved Hidden naïve Bayes model for computing probability estimates	Better than NB and Low time complexity	No optimization in assigning weight for the instances

3. Study Background

3.1. Attribute Rank based Feature Selection

The attribute rank based feature selection algorithm is a simple probabilistic method that makes use of probability based attribute scores for their contribution toward better classification. The relevant features that are significant for the classification are selected by computing the attribute rank based on the distinct attribute values present in the training set.

Initially, the model computes the overall database score based on the class labels as in Eq. (1) where p_i is the probability that an arbitrary instance in D belongs to class $C_{i.}$

$$Score\ (D) = Avg\left(\sum_{i=1}^{m} {p_i}^{p_i}\right) \qquad (1)$$

$$p_i = \frac{Number\ of\ instances\ belonging\ to\ C_i}{Total\ number\ of\ instances\ in\ D} \qquad (2)$$

The attribute score for each attribute having *n* distinct values can be computed by grouping the tuples based on n distinct values as {G_1, G_2, ..., G_j}. The count of tuples in each group is represented as {n_1, n_2, n_3, ..., n_j}. The calculation of the attribute score A_{score} is given in Eq. (3).

$$A_{score} = Score\ (D) - Avg\left(\sum_{j=1}^{n} p(n_j)\, X\ Score\ (G_j)\right) \qquad (3)$$

where $p(n_j)$ is the probability that an arbitrary instance in G_j belongs to class C_i. Finally, the score is calculated and the rank is allocated for each attribute. The ranks are then converted to rank scores using the rank sum method. The attributes having a rank score higher than the specified threshold value are then selected for the further classification process. A detailed illustration of attribute selection is discussed in [46].

3.2. Lagrange Multipliers

Shannon entropy computes the entropy of a random variable and it specifies the amount of information or the uncertainty in the variable [53]. Consider the random variable *A* with *n* possible outcomes as {A_1, A_2, .., A_n} that occur with the probability {$P(A_1)$, $P(A_2)$, .., $P(A_n)$}. Then the entropy of the variable *A* can be identified as in Eq. (4).

$$S = -\sum_{i=1}^{n} p(A_i)\ log_2\ p(A_i) \qquad (4)$$

However, in a more uncertain situation, the entropy value will be higher and it leads to chaos. Thus, to solve this problem effectively, Lagrange multipliers with maximum entropy can be applied. In simple words, maximum entropy allows choosing the best value from the number of the probability distribution that specifies the knowledge at the current state [54]. Maximum entropy is a powerful probabilistic model that has wide usage in the classification of data in different datasets such as text [55], image [56], audio [57] and video [58]. To solve using Lagrange multipliers, several constraints are to be taken into account.

For a random variable *A*, each possible outcome A_i has some probability of occurrence $p(A_i)$ where *i* represents the index representing possible outcomes. Generally, the probability distribution of a variable *p(A)* has specific constraints such as (a) the probability of occurrence of each outcome $p(A_i)$ always lies between 0 and 1 and (b) the sum of the probability of occurrence of all outcomes is 1 and is represented in Eq. (5).

$$1 = \sum_i p(A_i) \qquad (5)$$

For framing the next constraint, the expected value of the variable is computed by averaging the values corresponding to each outcome and its probabilities. Therefore, for the quantity G with the value $g(A_i)$ for each outcome, the probability distributions having the expected value G will be considered. However, the value of G always lies between the smallest $g(A_i)$ and the largest $g(A_i)$ and the constraint is given in Eq. (6).

$$\mathrm{G} = \sum_i g(A_i)p(A_i) \qquad (6)$$

4. Proposed Class Probability Distribution based Maximum Entropy (CPDME)

The proposed class probability distribution based maximum entropy model anticipates to classify the instances of sparse datasets having a minimum number of attributes and instances. The overall framework of the proposed class probability distribution based maximum entropy classification model (CPDME) is depicted in Fig. 1. The model is subdivided into four phases: 1) data pre-processing and feature selection, 2) relative distance computation, 3) attribute probability computation and 4) class probability based classification. Data pre-processing is an inevitable step in data mining that transforms incomplete raw data into a complete format that is suitable for mining [59]. In the proposed model, the missing and incomplete records are processed using predictive mean imputation [60]. Further, the data is transformed using data discretization [61] and min-max normalization [62]. To achieve feature selection, the model employs an attribute rank based feature selection (ARFS) which has been discussed in section 3.1.

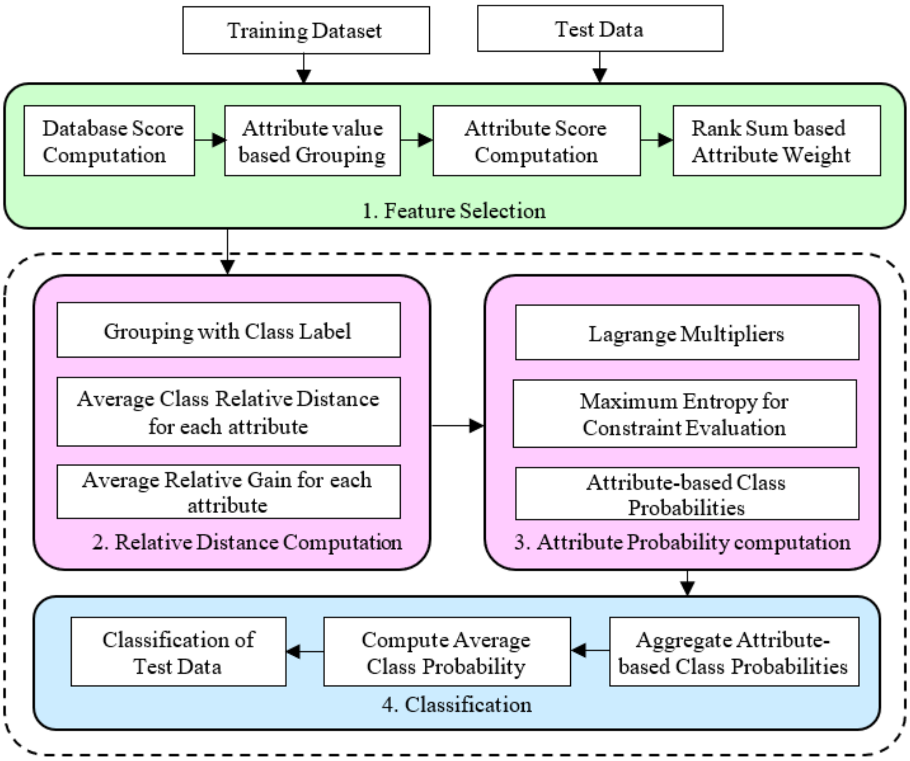
Training Dataset
Test Data
Database Score Computation
Attribute value based Grouping
Attribute Score Computation
Rank Sum based Attribute Weight
1. Feature Selection
Grouping with Class Label
Average Class Relative Distance for each attribute
Average Relative Gain for each attribute
2. Relative Distance Computation
Lagrange Multipliers
Maximum Entropy for Constraint Evaluation
Attribute-based Class Probabilities
3. Attribute Probability computation
Classification of Test Data
Compute Average Class Probability
Aggregate Attribute-based Class Probabilities
4. Classification

Fig. 1. Overall Framework of the Proposed Class Probability Distribution based Maximum Entropy Model

4.1. Relative Distance Computation

To compute the relative distance, the class relative distance and the relative gain are evaluated. Primarily, the set of training records is grouped based on the class label *i* where *i* vary from *1* to *n*. The average class relative distance $g(A_i)$ is computed for each attribute *A* concerning each class *i* as in Eq. (7).

$$g(A_i) = |C(A_i) - A_t| \qquad (7)$$

Thus, the value of $g(A_i)$ is computed by finding the relative distance between the value of an attribute *A* of the test sample represented as A_t with the centroid of the attribute value of all the training records belonging to each class *i* represented as $C(A_i)$. Here the centroid of the attribute value of all training records belonging to the class label *i* is the mean value of attribute *A* of i^{th} class. The centroid is computed as in Eq. (8) in which *m* represents the number of records in each class.

$$C(A_i) = \frac{\sum_{j=1}^{m} A_i^j}{m} \qquad (8)$$

Upon computing the value for $g(A_i)$ for each class *i*, the value of average relative gain G is computed by averaging the distance between each class $g(A_i)$ with all the other classes. The formula to compute the relative gain G of each attribute is presented in Eq. (9).

$$G = \frac{\sum_{j=1}^{n} \sum_{k=j+1,j<k}^{n} \left| g\left(A_i^j\right) - g\left(A_i^k\right) \right|}{n(n-1)/2} \tag{9}$$

4.2. Attribute Probability Computation

To compute the attribute probability, the maximum entropy principle has been extended to the larger system using Lagrange multipliers. Lagrange multipliers are named after the French mathematician, Joseph-Louis Lagrange [63]. Instead of processing the constraint equation to reduce the variables, Lagrange augments two more unknown variables *α* and *β* termed Lagrange multipliers. The Lagrange method assumes Maximum Entropy. Thus, the Lagrange function L can be defined by using the constraints given in Eq. (5) and Eq. (6) as in Eq. (10).

$$\mathrm{L} = \mathrm{S} - (\alpha - \log_2 e)\left(\sum_i p(A) - 1\right) - \beta\left(\sum_i g(A)p(A) - G\right) \tag{10}$$

Here, L can be maximized for each *p(A$_i$)* and is made by differentiating L concerning one of *p(A$_i$)* with *α*, *β*, and other *p(A$_i$)* as constant. The resulting functions are given in Eq. (11) and Eq. (12).

$$\log_2 \frac{1}{p(A_i)} = \alpha + \beta g(A_i) \tag{11}$$

$$p(A_i) = 2^{-\alpha} 2^{-\beta g(A_i)} \tag{12}$$

The values of *α* and *β* can be computed from the above equation specified for *p(A$_i$)* and the results are shown in Eq. (13) and Eq. (14).

$$\alpha = \log_2\left(\sum_i 2^{-\beta g(A_i)}\right) \tag{13}$$

$$f(\beta) = \sum_i (g(A_i) - G) 2^{-\beta(g(A_i) - G)} \tag{14}$$

The value of $f(\beta) = 0$, since it maximizes the L. On determining the value of *α* and *β*, the value of Entropy S can be computed by using the shortcut formula as shown in Eq. (15).

$$S = \alpha + \beta G \tag{15}$$

Thus, by solving Eq. (14), the value of the variable *β* can be obtained. And then by substituting the value *β* in Eq. (13), the value of *α* can be obtained. Once the value of *α* and *β* are known, they can be substituted in the expanded constraint given in Eq. (12) for various cluster groups *i*. Accordingly, the probability of an attribute for each class label *p(A$_i$)* is identified. The process is repeated for all the significant attributes selected through the feature selection phase.

4.3. Class Probability based Classification

Consecutively, to find the class probability, the probabilities *$p(A_i)$* of all attributes for each class label *i* for the given test sample is averaged. Finally, the test sample *t* can be labelled with the class having maximum average class probability as in Eq. (16).

$$t_i = \max_i \left(\frac{\sum_{j=1}^{m} p(A_i^j)}{m} \right) \qquad (16)$$

Here *i* represents the class label that varies from *1* to *n* and *j* represents the attribute index that varies from *1* to *m*.

The algorithm steps for the proposed class probability distribution based maximum entropy model for the classification of instances having fewer attributes are presented below in Algorithm 1.

```
Algorithm1: CPDME_Model
Input: A training set with m attributes, n instance, k classes, and test instances
Output:  Class label prediction for test instances
Procedure CPDME(training_set, test_data)
Begin
   //Preprocessing of Data
1.   Preprocess the given input training set by performing data cleaning by processing missing records, and
     data transformation using discretization and normalization.
   //Phase 1: Feature selection using ARFS
2.   Calculate the probability of the instances in each class c and the database score having k distinct
     classes.
3.   Compute the relevance score of the features having q discrete values
4.   Sort the attributes based on the computed score and rank them accordingly.
5.   Normalize the scores by evaluating rank weights using the rank sum method.
6.   Select the attributes having scores greater than the given threshold.
7.   For each attribute in the test instances
   //Phase 2: Relative Distance Computation
         a.   Group the training instances based on the class variable
         b.   Compute average class relative distance as in Eq. (7)
         c.   Evaluate the value of relative gain G as in Eq. (9)
   //Phase 3: Class Probability Computation using the Lagrange model
              a.   Evaluate the Lagrange multipliers α and β as in Eq. (13) and (14).
              b.   Evaluate the Entropy constraints and compute class probabilities as in Eq. (12) for all
                   classes.
   //Classification of the test instance
8.   For each class
         a.   Aggregate the class probabilities of all the attributes obtained in the previous phase and
              average the class probability as in Eq. (16)
         b.   Classify the instance with the class label having maximum probability
End Procedure
```

Here for each attribute, the sum of the probabilities of all the classes will always be 1. Similarly, the sum of class probabilities for each test instance will be 1. The overall workflow of the proposed CPDME model is presented in Fig. 2. This proposed classification model provides better results for the datasets having fewer attributes and instances.

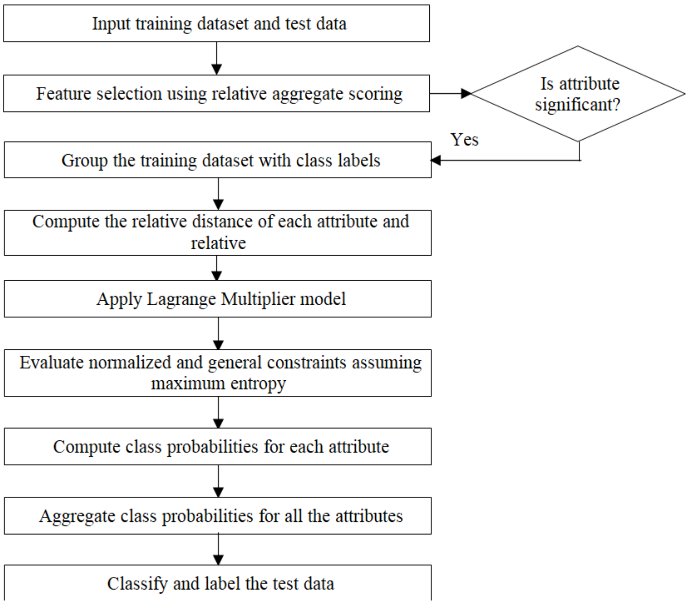
Input training dataset and test data
Feature selection using relative aggregate scoring
Is attribute significant?
Yes
Group the training dataset with class labels
Compute the relative distance of each attribute and relative
Apply Lagrange Multiplier model
Evaluate normalized and general constraints assuming maximum entropy
Compute class probabilities for each attribute
Aggregate class probabilities for all the attributes
Classify and label the test data

Fig. 2. Detailed Workflow of the Proposed Class Probability Distribution based Maximum Entropy Model

4.4. Case Study

The case study for the proposed probability distribution based maximum entropy model is discussed in this section. To explain the proposed model, the Iris dataset, donated by R. A. Fisher is employed. The dataset contains three classes of Iris in which each class contains 50 instances with 4 attributes. Theoretical experimentation has been performed by selecting 3 instances in each class with a total of 9 instances at random. As the dataset does not contain any missing values and as the number of attributes in the dataset is minimum, the selected instances do not undergo pre-processing step. The selected random samples from the Iris dataset that serves as training instances are presented in Table 2. Two random samples are picked from the Iris dataset, to serve as test instances and are shown in Table 3.

Initially, the training samples are grouped based on the class value. Then the probability of an attribute value of a test sample to be in each class is estimated. It is then combined with all the attribute values of the test sample to predict the classification. To proceed with an illustration of classifying the test sample *T*1, the sepal length attribute denoted as A_1 is evaluated. The centroid of an attribute in class 1 denoted as C(A_1) is 5.27 by computing the mean of values of A_1 in class 1. Similarly, the centroid of an attribute for other classes such as class 2 and class 3 are 5.77 and 6.77 respectively.

Table 2. Random Training Samples from Iris Dataset

Test Sample ID	Sepal Length in cm	Sepal Width in cm	Petal Length in cm	Petal Width in cm	Class
S1	4.7	3.2	1.3	0.2	Iris-Setosa
S2	5.4	3.9	1.7	0.4	Iris-Setosa
S3	5.7	3.8	1.7	0.3	Iris-Setosa
S4	6.9	3.1	4.9	1.5	Iris-Versicolor
S5	4.9	2.4	3.3	1	Iris-Versicolor
S6	5.5	2.4	3.8	1.1	Iris-Versicolor
S7	5.8	2.7	5.1	1.9	Iris-Virginica
S8	7.7	2.8	6.7	2	Iris-Virginica
S9	6.8	3.2	5.9	2.3	Iris-Virginica

Table 3. Test Samples to be Classified

Test Sample ID	Sepal Length in cm	Sepal Width in cm	Petal Length in cm	Petal Width in cm
T1	4.7	3.2	1.3	0.2
T2	7.7	3.0	6.1	2.3

Then the value of $g(A_1)$, $g(A_2)$ and $g(A_3)$ are computed as in Eq. (7) which is the difference between the centroid of the attribute value of a class and a test sample T1 and it result in $g(A_1) =0.23$, $g(A_2) =0.27$, $g(A_3) =1.27$. Based on the obtained values $g(A_1)$, $g(A_2)$ and $g(A_3)$, the expected value G is evaluated as in Eq. (9) and results in $G =0.6889$.

Eventually, to find the value of β, Eq. (14) can be expressed as below.

$-0.46 \text{ x } 2^{0.46\beta} - 0.42 \text{ x } 2^{0.42\beta} + 0.58 \text{ x } 2^{-0.58\beta} = 0$

By applying Logarithm, the value of β is computed as 1.08962.

On substituting the value of β in Eq. (13) results in

$\alpha = \log_2 (2^{-0.23\beta} + 2^{-0.27\beta} + 2^{-1.27\beta})$

After evaluating the above equation, the value of α is evaluated as 1.0281

The obtained value of α and β can be substituted in the expanded version of Eq. (12).

$p(A_1) = 2^{-1.0281} \text{ x } 2^{(-1.08962 \text{ x } 0.23)}$

$p(A_2) = 2^{-1.0281} \text{ x } 2^{(-1.08962 \text{ x } 0.27)}$

$p(A_3) = 2^{-1.0281} \text{ x } 2^{(-1.08962 \text{ x } 1.27)}$

Upon solving the equations, we obtain $p(A_1)=0.412177$, $p(A_2)=0.399911$, $p(A_3)=0.187912$.

Table 4. Class Probability of the Test Sample T_1

Class/ Attributes	Sepal Length	Sepal Width	Petal Length	Petal Width	Average Class Probability
Iris-Setosa	0.4122	0.0955	0.3316	0.3232	0.2906
Iris-Versicolor	0.3999	0.5581	0.3365	0.3625	0.4143
Iris-Virginica	0.1879	0.3465	0.3319	0.3143	0.2951
Attribute Probability	1.0000	1.0000	1.0000	1.0000	1.000

Table 5. Predicted Class for the Test Samples

Test Sample	Sepal Length in cm	Sepal Width in cm	Petal Length in cm	Petal Width in cm	Predicted Class
T1	4.7	3.2	1.3	0.2	2(Iris-Versicolor)
T2	7.7	3.0	6.1	2.3	3(Iris-Virginica)

The above steps can be continued for all the attributes in the given training dataset. The probability of all the attributes in each class is evaluated and the obtained results are presented in Table 4. It also specifies the overall probability of the test sample in each class. Here, the average class probability of the test sample of class 2 (Iris-Versicolor) is higher than the other classes. Hence, the test sample can be classified as Iris-Versicolor. It is also noted that the sum of attribute probability for all the classes will always 1. The predicted class labels for both test instances are presented in Table 5.

5. Experimental Analysis

The experimental and result analysis carried out for the proposed study is presented in this section. The experiments are performed on a system with intel core, i3-4005U CPU at 1.70Hz, 8 GB RAM, running 64bit OS of Windows 8.1 Pro windows edition. The experimental analysis is made for the proposed model with various datasets and the results are analysed in two sections 1) performance and statistical analysis with standard classifiers and 2) performance analysis with existing classification models.

5.1. Performance Analysis with Standard Classifiers

To evaluate the performance of the proposed model with standard classifiers, 17 datasets are employed. These datasets are available publically and are extracted from the UCI repository [64,] and KEEL [65] for classification. The number of attributes in the datasets varies widely from a minimum of 4 to a maximum of 60. Among the datasets used in the study, the datasets *Balance*, *Hayes Roth* and *Iris* have a minimum of 4 attributes and the dataset *Sonar* has a maximum number of attributes of 60. The number of classes in each dataset varies from 2 to 11. The datasets *German_credit*, *Ionosphere*, *Mushroom*, *Phoneme*, *Pima* and *Sonar* have the minimum number of class attribute values as 2 whereas *Vowel_context* has the maximum number of class attribute values as 11. Also, the number of instances in the datasets varies from 150 to 8124 with *Iris* as the smallest dataset with fewer instances and *Mushroom* as the largest dataset with a maximum number of instances. The number of attributes (bars graph) and classes (line graph) in each dataset used for the study is presented in Fig. 3 and the number of instances in each dataset is presented in Fig. 4.

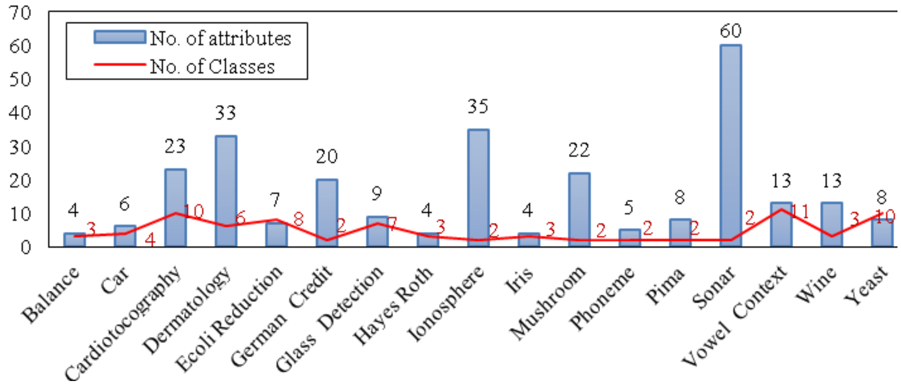

Fig. 3. Number of attributes and classes in different datasets

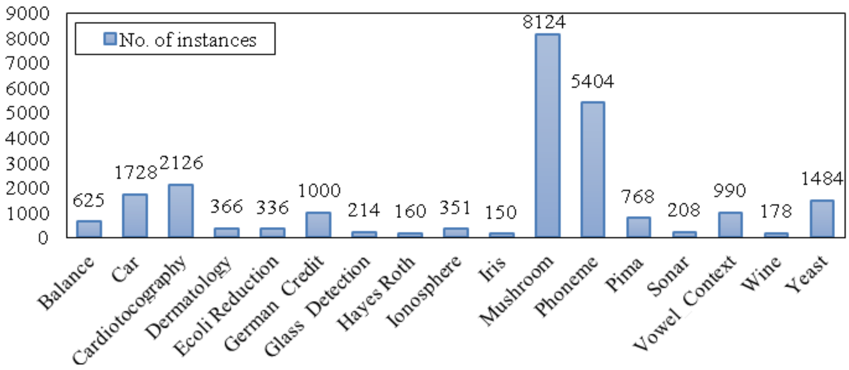

Fig. 4. Number of instances in different datasets

Diverse classifiers such as DL, NB, AB, LR, KNN, SRC, C4.5, SVM, ELM, RF, GBDT, DGC+ and AWPS are used for comparing the results of the proposed CPDME model. In general, 10-fold cross-validation is used for the evaluation of the proposed and existing models. For the classifiers that require parameter tuning, 80% of the instances in the datasets are used for training with 10% of the instances in the datasets being used as testing instances and the remaining 10% of the instances for tuning the parameters. While in the case of classifiers that do not require parameter tuning, 80% of the instances in the datasets are used for training and the remaining 20% of the instances in the datasets are used as a testing set [38]. Also, before applying classification, the significant attributes are selected utilizing the attribute rank based feature selection.

Accuracy Comparison: Table 6 shows the accuracy obtained with different classifiers for various datasets used for the analysis. The underlined values indicate the highest accuracy obtained for each dataset. From the results obtained, it is evident that the proposed model offers a better accuracy rate for 7 datasets such as *Car*, *Ecoli*

Reduction, *Glass_Detection*, *Hayes Roth*, *Iris*, *Phoneme* and *Vowel_Context* out of 17 datasets used for the evaluation. The average accuracy of CPDME with all 17 datasets is 89.9%. Out of 13 classifiers compared, the classifiers such as AWPS, RF and GBDT have the next higher accuracies at 88.2%, 87.9% and 87.3% respectively. Though the proposed model seems to be effective only with 7 datasets, it acquires the top position in average classification accuracy with an average rank of 3.47 and the classifiers GBDT, AWPS and RF, acquire the next three positions with ranks of 4.18, 4.71 and 4.88 respectively.

The statistical analysis for the obtained accuracy for the classification process is made using ANOVA with the null hypothesis that there is no significant difference in the accuracy of the classification algorithms. The statistical model is generated using F-distribution for which the obtained F value is 8.46 and the critical value is 1.76. The computed critical difference is 6.69 and the results are significant at a 5% significance level. Since the F value is greater than F critical value, the null hypothesis can be rejected and thus the alternate hypothesis is accepted indicating that there is a difference in the accuracy of the classification algorithms used for comparison.

Table 6. Accuracy results for different datasets

Dataset	CPDME	AWPS	DGC+	GBDT	RF	ELM	SVM	C4.5	SRC	KNN	LR	AB	NB	DL
Balance	0.987	0.904	0.899	0.968	0.952	0.952	0.921	0.857	1.000	0.952	0.937	0.809	0.968	0.460
Car	1.000	0.995	0.952	1.000	0.971	0.948	0.919	0.954	0.861	0.856	0.676	0.671	0.786	0.671
Cardiotocography	0.892	0.995	0.999	0.911	0.897	0.747	0.855	0.864	0.737	0.718	0.869	0.390	0.714	0.019
Dermatology	0.963	0.979	0.975	0.973	0.946	0.946	1.000	0.946	0.973	0.919	0.973	0.541	0.946	0.324
Ecoli Reduction	0.892	0.829	0.823	0.879	0.818	0.879	0.849	0.849	0.758	0.818	0.788	0.667	0.727	0.364
German_Credit	0.735	0.752	0.732	0.760	0.740	0.710	0.720	0.740	0.690	0.720	0.720	0.710	0.760	0.740
Glass_Detection	0.857	0.758	0.704	0.762	0.810	0.905	0.810	0.429	0.667	0.762	0.714	0.429	0.381	0.429
Hayes Roth	0.872	0.854	0.840	0.786	0.786	0.786	0.786	0.786	0.643	0.500	0.643	0.214	0.786	0.571
Ionosphere	0.912	0.945	0.931	0.917	0.917	0.889	0.806	0.944	0.944	0.889	0.889	0.917	0.806	0.722
Iris	0.975	0.972	0.953	0.947	0.953	0.922	0.960	0.867	0.967	0.967	0.953	0.947	0.867	0.867
Mushroom	0.987	0.999	0.995	1.000	0.995	0.978	1.000	1.000	1.000	0.998	0.987	0.967	0.957	0.967
Phoneme	0.904	0.878	0.871	0.867	0.895	0.880	0.775	0.847	0.899	0.893	0.745	0.771	0.734	0.285
Pima	0.827	0.737	0.745	0.701	0.805	0.662	0.650	0.766	0.597	0.610	0.805	0.831	0.753	0.597
Sonar	0.852	0.835	0.848	0.905	0.952	0.619	0.905	0.762	0.857	0.714	0.667	0.857	0.762	0.667
Vowel_Context	0.999	0.985	0.982	0.849	0.939	0.990	0.970	0.788	0.980	0.950	0.697	0.162	0.636	0.111
Wine	0.982	0.972	0.973	1.000	0.944	0.722	0.944	1.000	0.944	0.833	0.889	0.889	0.944	0.278
Yeast	0.645	0.598	0.593	0.622	0.622	0.649	0.628	0.514	0.574	0.547	0.622	0.412	0.595	0.331
Avg. Accuracy	*0.899*	*0.882*	*0.871*	*0.873*	*0.879*	*0.834*	*0.853*	*0.818*	*0.829*	*0.803*	*0.798*	*0.658*	*0.772*	*0.494*
Avg. Rank	*3.47*	*4.71*	*6.06*	*4.18*	*4.88*	*7.29*	*6.00*	*6.82*	*6.76*	*8.71*	*8.41*	*10.59*	*9.24*	*12.65*

AUC Comparison: Table 7 shows the AUC values obtained for the 13 classifiers with 17 datasets in which the underlined values represent the maximum AUC obtained for each dataset. From the reported result, it is evident that the proposed model offers a better AUC value for 7 datasets such as *Car*, *Ecoli Reduction*, *Glass_Detection*, *Hayes Roth*, *Iris*, *Phoneme* and *Yeast* with an average AUC value of 0.946. Among 13 standard classifiers, the next highest average AUC values are acquired by DGC+, AWPS, and RF as 0.932, 0.930 and 0.883 respectively. Despite acquiring minimum AUC values with 10 datasets, the proposed model holds the first position with an average rank of 2.82 which is better than other classifiers such as DGC+ and AWPS, with average ranks of 3.47 and 3.59.

The statistical analysis for the obtained AUC is carried out using the ANOVA test with the null hypothesis stating that there is a difference in AUC values of the classification algorithms. The statistical model is constructed using F-distribution in which the obtained F value and critical value are 10.326 and 1.76 (10.326 > 1.76). The computed critical difference is 8.56 and the results are significant at a 5% significance level. Thus, the null hypothesis is rejected and the alternate hypothesis is accepted which indicates that there is a difference in AUC values of the classification algorithms under comparison.

Table 7. AUC Comparison among different classifiers

Dataset	CPDME	AWPS	DGC+	GBDT	RF	ELM	SVM	C4.5	SRC	KNN	LR	AB	NB	DL
Balance	0.992	0.862	0.875	0.833	0.833	0.833	0.867	0.781	1.000	0.984	0.956	0.724	0.833	0.500
Car	1.000	0.994	0.998	1.000	0.951	0.930	0.897	0.929	0.878	0.836	0.575	0.500	0.867	0.509
Cardiotocography	0.895	0.999	0.996	0.882	0.874	0.823	0.862	0.844	0.834	0.824	0.922	0.736	0.822	0.500
Dermatology	0.983	0.989	0.991	0.980	0.980	0.960	1.000	0.921	0.980	0.980	0.980	0.752	0.960	0.684
Ecoli	0.982	0.978	0.957	0.875	0.903	0.906	0.895	0.908	0.892	0.892	0.906	0.763	0.888	0.500
German_Credit	0.832	0.751	0.743	0.699	0.939	0.654	0.645	0.685	0.623	0.694	0.654	0.613	0.741	0.500
Glass_Detection	0.992	0.865	0.854	0.719	0.748	0.986	0.790	0.817	0.875	0.963	0.727	0.806	0.815	0.500
Hayes Roth	0.965	0.936	0.948	0.955	0.927	0.952	0.949	0.902	0.936	0.834	0.895	0.914	0.952	0.904
Ionosphere	0.902	0.950	0.931	0.889	0.644	0.844	0.835	0.909	0.909	0.864	0.869	0.889	0.809	0.622
Iris	0.999	0.999	0.994	0.874	0.986	0.987	0.989	0.887	0.878	0.897	0.957	0.960	0.979	0.878
Mushroom	0.992	0.999	0.995	1.000	0.994	0.992	0.992	0.935	0.994	0.927	0.994	0.972	0.991	0.921
Phoneme	0.898	0.875	0.866	0.844	0.859	0.848	0.651	0.799	0.863	0.856	0.639	0.672	0.707	0.500
Pima	0.857	0.788	0.866	0.677	0.747	0.628	0.617	0.731	0.563	0.579	0.774	0.806	0.725	0.500
Sonar	0.799	0.886	0.891	0.896	0.885	0.882	0.799	0.721	0.879	0.882	0.789	0.633	0.856	0.692
Vowel	0.998	0.985	0.982	0.914	0.935	0.999	0.998	0.914	0.997	1.000	0.853	0.713	0.791	0.576
Wine	0.998	0.965	0.973	1.000	0.967	0.719	0.900	1.000	0.967	0.790	0.873	0.917	0.967	0.500
Yeast	0.999	0.996	0.991	0.847	0.837	0.829	0.825	0.825	0.799	0.823	0.836	0.675	0.838	0.500
Avg. AUC	*0.946*	*0.930*	*0.932*	*0.876*	*0.883*	*0.869*	*0.854*	*0.853*	*0.875*	*0.860*	*0.835*	*0.767*	*0.855*	*0.605*
Avg. Rank	*2.82*	*3.59*	*3.47*	*6.18*	*6.41*	*7.12*	*7.76*	*8.41*	*7.12*	*8.29*	*8.35*	*10.94*	*8.47*	*13.59*

Execution Time Comparison: The running time to train and test the proposed CPDME model are analysed and compared with 13 different classifiers with 17 different datasets. The results of the execution time for the proposed and the existing models are presented in Table 8. The underlined values in the table represent the minimum execution time for the dataset. From the results obtained, it is clear that the proposed model has a minimum running time for the datasets such as *Balance*, *Hayes Roth*, *Iris*, *Phoneme* and *Yeast* which are less than 2 ms. In general, the average running time of the proposed model is 4.64 ms and acquires 7th rank whereas the execution times of the top 6 classifiers are 28 ms (NB), 0.30 ms (KNN), 0.34 ms (C4.50), 0.34 ms (AB), 1.03 ms (LR), 3.77 ms (SVM), 4.14 ms (AWPS). Optimistically, still, the proposed model has a minimum execution time than the other 8 classifiers used in the comparison.

Table 8. Execution Time (in ms) Comparison among different classifiers

Dataset	CPDME	AWPS	DGC+	GBDT	RF	ELM	SVM	C4.5	SRC	KNN	LR	AB	NB	DL
Balance	0.121	4.45	5.71	15.47	7.481	4.11	0.723	0.022	4.88	0.022	0.04	0.02	0.016	3.09
Car	0.236	5.06	11.89	34.71	12.73	38.93	0.589	0.044	83.32	0.028	0.15	0.06	0.034	2.964
Cardiotocography	10.1	6.13	14.64	190.8	109.61	64.77	0.285	0.871	307.3	0.631	11.5	0.419	0.621	1.14
Dermatology	9.98	2.12	9.89	20.7	48.25	1.54	0.125	0.879	1.25	0.325	0.879	0.623	0.741	2.85
Ecoli	0.396	2.09	5.72	16.49	10.92	1.21	0.425	0.014	1.09	0.022	0.04	0.01	0.012	1.31
German_Credit	2.36	4.11	13.18	13.29	89.69	11.92	0.171	0.952	20.66	0.369	0.234	0.412	0.357	4.265
Glass_Detection	0.936	2.02	5.98	15.61	15.39	0.52	0.323	0.014	0.35	0.034	0.04	0.01	0.013	1.875
Hayes Roth	0.109	1.96	5.07	11.58	10.36	9.55	7.51	0.187	2.35	0.245	0.09	0.09	0.131	5.87
Ionosphere	2.91	3.94	10.56	7.22	60.14	1.21	0.721	0.532	0.99	0.567	0.236	0.413	0.561	3.89
Iris	1.22	1.86	5.11	8.32	10.48	5.32	7.99	0.057	1.89	0.057	0.08	0.234	0.015	2.457
Mushroom	3.978	13.84	26.01	16.32	31.26	19.49	28.78	0.723	45.5	0.811	1.18	0.987	0.725	30.49
Phoneme	0.203	9.43	23.01	26.64	34.5	388.45	0.331	0.187	3530.1	0.074	0.19	0.22	0.091	2.753
Pima	0.121	2.44	10.24	7.19	16.57	7.58	0.133	0.028	11.88	0.022	0.03	0.241	0.038	2.45
Sonar	12.31	2.35	12.79	15.36	18.18	11.23	15.54	0.977	17.28	0.932	1.23	0.912	0.812	12.36
Vowel	0.102	1.93	13.08	112.3	43.69	11.13	0.245	0.083	20.31	0.024	0.6	0.321	0.125	2.68
Wine	0.222	1.85	6.37	8.57	19.79	0.57	0.345	0.012	0.59	0.023	0.03	0.369	0.235	1.235
Yeast	1.832	4.87	14.24	93.85	25.96	29.36	0.184	0.259	112.8	0.949	0.89	0.412	0.196	1.857
Avg. Exec. Time.	*2.77*	*4.14*	*11.38*	*36.14*	*33.24*	*35.70*	*3.79*	*0.34*	*244.86*	*0.30*	*1.03*	*0.34*	*0.28*	*4.91*
Avg. Rank	*6.47*	*8.71*	*10.88*	*12.41*	*13.06*	*9.94*	*6.29*	*3.18*	*10.94*	*3.06*	*4.35*	*3.65*	*2.71*	*9.12*

From the result analysis, it is clear that the proposed CPDME model outperforms other existing models for the various datasets such as *Car*, *Ecoli Reduction*, *Glass_Detection*, *Hayes Roth*, *Iris*, *Phoneme* and *Vowel_Context* in which most of the datasets have fewer attributes. This shows that the proposed model is effective with the datasets having the minimum number of attributes and offers better performance. The obtained ranks for accuracy, AuC and execution time of the proposed CPDME and the other standard models under comparison are presented as a graph and shown in Fig. 5.

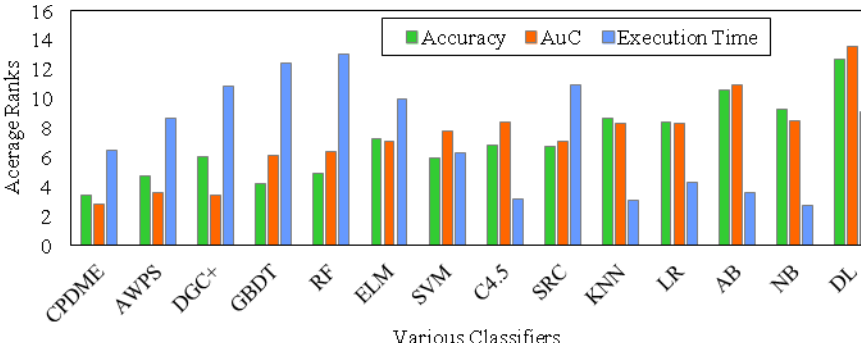

Fig. 5. Ranks obtained for various metrics

5.2. Performance Comparison with Existing Models

An analysis has been carried out for the proposed model by evaluating the classification accuracy of various models using 36 datasets. The datasets used for the study are publically available and are downloaded from the UCI repository [64] and KEEL [65]. The results of the proposed model are compared with the various existing model such as AIWNB^E, AIWNB^L, CAWNB, AVFWNB and NB. The values in Table 9 are the result of performing the average on the accuracies obtained from 10 individual runs under stratified ten-fold cross-validation as in [49]. The various algorithms are applied to the same training and test set. The values represented in the boldface at each row indicate the highest accuracy value corresponding to the dataset. Moreover, the underlined values indicate that the proposed model outperforms other models under comparison with paired two-tailed t-tests at the p=0.05 significance level. The symbol * denotes the significant performance degradation over its competitors. The last two rows indicate the average accuracy and Win/Tie/Lose (W/T/L) of each classifier. Each W/T/L specifies that the proposed model wins W datasets, ties on T datasets and loses on L datasets to the respective competitor model [49].

From the analysis, it is clear that the proposed CPDME classifier has the highest accuracy for 15 datasets which is better than the competitors. The models IWHNB, AIWNB^E, AIWNB^L, CAWNB, AVFWNB and NB have the highest classification accuracy for 8, 3, 7, 3, 2 and 2 datasets respectively. In terms of the average classification accuracy, the proposed CPDME classifier is 86.93%. On the other hand, the models such as IWHNB, AIWNB^E, AIWNB^L, CAWNB, AVFWNB and NB have the average accuracy of 86.37%, 84.94%, 85.52%, 84.41%, 84.21% and 83.86% respectively. The performance of the proposed model is as similar as the IWHNB classifier. Also, the CPDME model has 6 wins, 26 ties and 4 losses which is better than the other models IWHNB (6 wins, 26 ties, 4 losses) AIWNB^E (8 wins, 25 ties, 3 losses), AIWNB^L (6 wins, 27 ties, 3 losses), CAWNB (13 wins, 21 ties, 2 loss), AVFWNB (14 wins, 20 ties, 2 loss) and NB (17 wins, 17 ties, 2 loss).

Table 9. Comparison of Classification Accuracy

Dataset	CPDME	IWHNB	AIWNB^E	AIWNB^L	CAWNB	AVFWNB	NB
Anneal	98.94±1.12	98.31±1.29	**98.94±1.05**	98.90±1.10	98.5±1.29	98.62±1.15	96.36±1.97
Anneal.ORIG	**95.12±2.30**	94.65±2.24	95.06±2.23	95.06±2.23	94.6±2.48	93.32±2.65	92.71±2.7
Audiology	**85.32±6.91**	78.17±7.15	83.93±7.00	84.81±6.83	74.22±6.36	78.58±8.44	75.74±6.58
Autos	**88.93±9.01**	85.56±7.93	78.04±9.02	79.80±8.63	77.95±8.95	77.27±9.43	77.02±9.69
Balance-scale	**74.21±4.51**	69.05±3.74	73.75±4.22	73.52±4.39	73.76±4.15	71.1±4.3	71.08±4.29
Breast-cancer	70.01±7.21	70.47±6.29	71.9±7.55	71.52±7.23	**72.46±7.25***	71.41±7.98	72.32±7.91
Breast-w	97.11±1.63	96.30±1.94	97.17±1.68	97.15±1.77	97.14±1.81	**97.48±1.68***	97.25±1.79
Colic	82.36±5.43	81.20±6.00	**83.45±5.45***	83.4±5.44	83.34±5.62	81.47±5.86	81.2±5.8
colic.ORIG	73.36±6.3	74.23±6.52	73.87±6.4	**74.38±6.7***	73.7±6.46	72.91±6.34	73.43±6.27
credit-a	86.12±3.84	85.23±3.82	**87.03±3.83***	86.93±3.85	86.99±3.81	86.23±3.85	86.17±3.94
credit-g	75.83±3.7	75.85±3.69	75.81±3.6	**75.86±3.67**	75.7±3.53	75.38±3.9	75.4±4.01
Diabetes	**79.36±4.7**	76.75±4.20	77.87±4.86	78.32±4.67	78.01±4.89	77.89±4.66	77.88±4.65
Glass	**78.21±8.3**	77.70±8.98	74.02±8.41	74.9±8.25	73.37±8.38	76.25±8.07	74.2±8.11
heart-c	83.26±6.41	81.52±7.12	82.71±6.61	82.81±6.61	82.94±6.57	83.04±6.68	**83.73±6.46***
heart-h	84.87±5.91	84.56±6.05	84.29±5.85	84.26±5.89	83.82±6.16	**84.9±5.68***	84.43±5.88
heart-statlog	83.31±6.28	82.33±6.59	83.22±6.61	83.19±6.71	**83.44±6.69**	83.78±6.29	83.74±6.25
Hepatitis	85.91±9.24	**87.38±8.43***	85.75±8.97	86±9.07	85.95±9.25	85.38±9	85.05±9.45
Hypothyroid	98.23±0.59	**99.32±0.40***	99.07±0.48	99.05±0.5	98.56±0.56	98.98±0.48	98.74±0.57
Ionosphere	92.25±3.92	**93.96±3.65**	92.4±4.13	92.68±3.76	91.82±4.34	91.94±4.09	91.37±4.55
Iris	**96.23±5.8**	93.27±5.72	94.4±5.5	94.4±5.5	94.4±5.5	94.4±5.5	94.33±5.56
kr-vs-kp	93.85±1.41	92.70±1.37	93.73±1.28	**94.06±1.27***	93.58±1.32	88.18±1.86	87.81±1.9
Labor	**96.33±10.13**	95.90±9.21	94.33±9.3	93.8±10.17	92.1±10.94	94.33±10.13	93.83±10.41
Letter	85.6±0.85	**90.17±0.62***	75.56±0.89	79.6±0.85	75.22±0.83	75.07±0.84	74.67±0.86
Lymphography	**86.12±7.83**	85.89±8.02	84.68±7.99	85.08±7.72	84.81±8.13	85.49±7.83	85.7±7.95
Mushroom	**99.9±0.31**	99.96±0.06	99.53±0.23	99.71±0.2	99.19±0.32	99.12±0.31	98.03±0.49
primary-tumor	**48.21±5.37**	46.14±6.17	47.76±5.25	47.76±5.21	47.2±5.27	45.85±6.53	47.11±5.65
Segment	95.18±1.41	**96.87±1.07**	94.16±1.38	95.32±1.32	93.47±1.46	93.69±1.41	92.91±1.56
Sick	96.23±0.89	**97.52±0.76**	97.33±0.85*	**97.36±0.83***	**97.36±0.84***	97.02±0.86	97.07±0.84
Sonar	83.89±8.57	84.63±7.72	82.23±8.65	82.28±8.57	82.56±8.25	84.49±7.79	**84.96±7.57***
Soybean	94.62±2.23	94.61±2.18	94.74±2.19	**94.82±2.24**	93.66±2.73	94.52±2.36	93.53±2.79
Splice	96.32±1.11	96.24±1.00	96.21±0.99	**96.55±1.01**	96.19±0.99	95.61±1.11	95.58±1.12
Vehicle	72.58±3.58	**73.70±3.41***	63.59±3.92	67.57±3.27	62.91±3.88	63.36±3.87	62.64±3.84
Vote	**94.74±3.21**	94.39±3.21	92.18±3.76	93.68±3.52	92.11±3.74	90.25±3.95	90.3±3.89
Vowel	**89.95±4.12**	**90.32±2.71**	69.98±4.11	74.48±3.93	68.84±4.3	67.46±4.62	66±4.58
waveform-5000	**88.61±1.52**	86.24±1.45	82.98±1.37	83.51±1.38	83.11±1.38	80.65±1.46	80.76±1.49
Zoo	**98.71±5.2**	98.33±3.72	96.05±5.6	96.05±5.6	95.96±5.61	96.05±5.6	95.75±5.68
Average	86.93	86.37	84.94	85.52	84.41	84.21	83.86
W/T/L	-	6/26/4	8/25/3	6/27/3	13/21/2	14/20/2	17/17/2

6. Result Analysis of the Proposed Model

6.1. Complexity Analysis

The computational complexity of the proposed CPDME model using Big-O notation is *O(nmk)* where *m* is the attribute count in the dataset, *k* is the count of target class values and *n* is the count of the instances at each class in the dataset. The computation complexity of the other existing algorithms AWPS and DGC+ having higher ranks in average classification accuracy or AUC values are *O(nm)* and $O(mn^2)$, where *n* is the count of instances and *m* is the count of attributes in the dataset. Similarly, the computational complexity of RF is *O(tmn(log n))* in which the complexity depends on the number of trees (t) to be constructed. Though the computational complexity of the proposed classifier is slightly higher than the other models, the accuracy of CPDME is better than many of the existing algorithms.

6.2. Statistical Analysis of Results

A statistical analysis has been carried out to assess the performance of the proposed model for which the results presented in Table 6 are used. The highest accuracy obtained by the proposed model among the 17 datasets is statistically distributed based on various characteristics such as attribute count, instance count and the number of classes. The statistical distribution is provided in Table 10. The column *count* indicates the number of datasets won by the proposed model with the highest accuracy and the percentage indicates the values in percentage. Thus, out of 12 datasets having an attribute count of less than 20, the proposed model acquires the highest accuracy for 7 datasets indicating a success rate of 58.33%.

Table 10. Statistical Analysis of Data Characteristics for Proposed CPDME

Characteristics	No. of datasets	CPDME		Other Models
		Count	Percentage	
Attributes ≤ 20	12	7	58.33	41.67
Attributes > 20	5	0	0	100
Instances ≤ 500	8	4	50	50
Instances >500	9	3	33.33	66.67
Classes ≤ 5	11	4	36.36	63.64
Classes > 5	6	2	33.33	66.67

Similarly, out of 5 datasets having an attribute count greater than 20, the proposed model has 0 success signifying that the standard models under comparison achieve the highest accuracy (100%). While considering the instances less than or equal to 500, the proposed CPDME model has a success rate of about 50% with the highest accuracy for 4 out of 8 datasets. On the other hand, for the datasets having an instance count greater than 500, the proposed model has less success rate of about 33.33% by achieving the highest accuracy for 3 out of 9 datasets. With a distribution based on the distinct class count, the results are even for the proposed and existing models. Thus, the model offers better results than many other competitors with the datasets having fewer attributes or instances.

Furthermore, the highest accuracy obtained by the proposed model and existing models among the 36 datasets presented in Table 9 is statistically distributed based on various characteristics such as attribute count, instance count. Table 11 provides the statistical distribution of highest accuracies on number of datasets for the models CPDME, IWHNB, AIWNB^E, AIWNB^L, CAWNB, AVFWNB and NB.

Table 11. Comparison of the percentage of datasets having higher accuracy

Characteristics	CPDME	IWHNB	AIWNB^E	AIWNB^L	CAWNB	AVFWNB	NB
Attributes ≤ 25	50.00	20.83	8.33	8.33	8.33	8.33	4.17
Attributes > 25	25.00	25.00	8.33	41.67	8.33	0.00	8.33
Instances ≤ 500	50.00	11.11	5.56	5.56	11.11	5.56	11.11
Instances >500 and ≤ 1000	36.36	27.27	18.18	18.18	0.00	9.09	0.00
Instances > 1000	28.57	42.87	0.00	57.14	14.29	0.00	0.00

From the analysis, the proposed CPDME has a higher success rate of about 50% with the highest accuracy for the datasets having attributes less than or equal to 25, whereas the combined success rate of other 6 models is 50%. For the datasets having a number of attributes greater than 25, the proposed CPDME model has the lowest success rate of about 25% than other models (75%). Correspondingly, the proposed CPDME model has a higher lowest success rate of 50% and 36% with the highest accuracy for the datasets having a number of instances less than or equal to 500 and between 500 and 1000 respectively. With the datasets having instance count greater than 1000, the proposed model has a less lowest success rate of about 28.57% of the highest accuracy. The increase in the number of instances or attributes gradually decreases the performance of the proposed model. Thus, from the results of the statistical analysis, it clear that the proposed model offers better results with the minimum number of attributes and instances.

7. Conclusion

This paper suggests the class probability distribution based on maximum entropy classification to classify the instances of the datasets having fewer attributes and instances. In the first phase, the important features are identified using attribute rank based feature selection. For each selected attribute, the average class relative distance is evaluated for the training samples. Then the relative gain of the attributes is computed from the test sample and the relative distance of each class. The Lagrange multipliers are applied and evaluated and the class probabilities concerning the attributes are computed by maximizing the entropy. Finally, the class label is predicted by aggregating the class probabilities of all the attributes. Experimental analysis has been performed with two sets of experiments using 17 and 36 datasets. The proposed model offers better average accuracy of about 89.9% and 86.93% for the two experiments respectively which is better than many of the other existing and standard models. The statistical result analysis shows that the proposed model offers better results with improved accuracy for more than 50% of the datasets having fewer attributes and instances than other competitors. The future work focuses on the imbalanced class distribution along with the sparse distribution of attributes and instances. Though the proposed model has better accuracy, it suffers from time overhead which is below the top 5 positions in comparison with other models. Thus, future work concentrates on increasing the classification speed of the proposed model.

References

1. Olson, D.L., Shi, Y.: Introduction to Business Data Mining. McGraw-Hill Education, New York. (2007)
2. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques, A Volume in the Morgan Kaufmann Series in Data Management Systems, Third Edition. Elsevier. (2011)
3. Urso, A., Fiannaca, A., La Rosa, M., Ravì, V., Rizzo, R.: Data Mining: Classification and Prediction. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, Vol. 1, No. 3, 384-402. (2018)
4. Francis, B.K., Babu, S.S.: Predicting Academic Performance of Students Using a Hybrid Data Mining Approach. Journal of Medical Systems, Vol. 43, No. 6, 1-15. (2019)
5. Shen, F., Zhao, X., Li, Z., Li, K., Meng, Z.: A Novel Ensemble Classification Model based on Neural Networks and a Classifier Optimisation Technique for Imbalanced Credit Risk Evaluation. Physica A: Statistical Mechanics and its Applications, Vol. 526, 121073. (2019)
6. Fatima, M., Pasha, M.: Survey of Machine Learning Algorithms for Disease Diagnostic. Journal of Intelligent Learning Systems and Applications, Vol. 9, No. 1, 1-16. (2017)
7. Van Klompenburg, T., Kassahun, A., Catal, C.: Crop Yield Prediction using Machine Learning: A Systematic Literature Review. Computers and Electronics in Agriculture, Vol. 177, 105709. (2020)
8. Dingli, A., Marmara, V., Fournier, N.S.: Comparison of Deep Learning Algorithms to Predict Customer Churn within a Local Retail Industry. International Journal of Machine Learning and Computing, Vol. 7, No. 5, 128-132. (2017)
9. Sharaff, A., Gupta, H.: Extra-tree classifier with metaheuristics approach for email classification. In Advances in Computer Communication and Computational Sciences, Springer, Singapore. pp. 189-197. (2019)
10. Choubin, B., Zehtabian, G., Azareh, A., Rafiei-Sardooi, E., Sajedi-Hosseini, F., Kişi, Ö.: Precipitation Forecasting Using Classification and Regression Trees

(CART) Model: A Comparative Study of Different Approaches. Environmental Earth Sciences, Vol. 77, No. 8, 1-13. (2018)

11. Nayak, S., Panigrahi, C.R., Pati, B., Nanda, S., Hsieh, M.Y.: Comparative analysis of HAR datasets using classification algorithms. Computer Science and Information Systems, Vol. 19, No. 1, 47-63 (2022).
12. Rogić, S., Kašćelan, L.: Class balancing in customer segments classification using support vector machine rule extraction and ensemble learning. Computer Science and Information Systems, Vol. 18, No. 3, 893-925, (2021).
13. Sathya Bama, S., Irfan Ahmed, M.S., Saravanan, A.: Network Intrusion Detection using Clustering: A Data Mining Approach. International Journal of Computer Applications, Vol. 30, No. 4, 14-17. (2011)
14. Sahu, K., Srivastava, R.K.: Predicting software bugs of newly and large datasets through a unified neuro-fuzzy approach: Reliability perspective. Advances in Mathematics: Scientific Journal, Vol. 10, No. 1, 543-555 (2021).
15. Liu, H., Motoda, H., Setiono, R., Zhao, Z.: Feature Selection: An Ever Evolving Frontier in Data Mining. In Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, Proceedings of Machine Learning Research, Vol. 10, 4-13. (2010)
16. Kumar, P., Ambekar, S., Kumar, M., Roy, S.: Analytical Statistics Techniques of Classification and Regression in Machine Learning. In Data Mining-Methods, Applications and Systems. IntechOpen. (2020)
17. Altman, N.S.: An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician, Vol. 46, No. 3, 175-185. (1992)
18. Shaban, W.M., Rabie, A.H., Saleh, A.I., Abo-Elsoud, M.A.: A New COVID-19 Patients Detection Strategy (CPDS) Based on Hybrid Feature Selection and Enhanced KNN Classifier. Knowledge-Based Systems, Vol. 205, 106270. (2020)
19. Leung, K.M.: Naive Bayesian Classifier. Polytechnic University Department of Computer Science/Finance and Risk Engineering, 123-156. (2007)
20. Berrar, D.: Bayes' Theorem and Naive Bayes Classifier. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics; Elsevier Science Publisher: Amsterdam, The Netherlands, 403-412. (2018)
21. Cox, D.R.: The Regression Analysis of Binary Sequences. Journal of the Royal Statistical Society: Series B (Methodological), Vol. 20, No. 2, 215-232. (1958)
22. Boonchuay, K., Sinapiromsaran, K., Lursinsap, C.: Decision Tree Induction based on Minority Entropy for the Class Imbalance Problem. Pattern Analysis and Applications, Vol. 20, No. 3, 769-782. (2017)
23. Cherfi, A., Nouira, K., Ferchichi, A.: Very Fast C4. 5 Decision Tree Algorithm. Applied Artificial Intelligence, Vol. 32, No. 2, 119-137. (2018)
24. Cortes, C., Vapnik, V.: Support Vector Networks. Machine Learning, Vol. 20, No. 3, 273–297. (1995)
25. Suthaharan, S.: Support Vector Machine, Machine Learning Models and Algorithms for Big Data Classification, Integrated Series in Information Systems, Springer, Boston, MA., 207-235. (2016)

26. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust Face Recognition via Sparse Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 2, 210-227. (2008)
27. Smolyakov, V.: Ensemble Learning to Improve Machine Learning Results. Stats and Bots, Available: https://blog.statsbot.co/ensemble-learning-d1dcd548e936. (2017)
28. Freund, Y., Schapire, R. E.: Experiments with a New Boosting Algorithm. In International Conference on Machine Learning, Vol. 96, 148–156. (1996)
29. Breiman, L.: Random Forests. Machine Learning, Vol. 45, No. 1, 5–32. (2011)
30. Wolpert, D.H.: Stacked Generalization. Neural Networks. Vol. 5, No. 2, 241–259. (1992)
31. Friedman, J. H.: Stochastic Gradient Boosting. Computational Statistics & Data Analysis, Vol. 38, No. 4, 367–378. (2002)
32. Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H.: State-of-the-Art in Artificial Neural Network Applications: A Survey. Heliyon, Vol. 4, No. 11. (2018)
33. Bengio, Y.: Learning Deep Architectures for AI. Found. Trends Mach. Learn., Vol. 2, 1–127. (2009)
34. Sejnowski, T.J.: The Deep Learning Revolution. MIT Press. (2018)
35. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme Learning Machine for Regression and Multiclass Classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 42, No. 2, 513-529. (2011)
36. Çığşar, B., Ünal, D.: Comparison of Data Mining Classification Algorithms Determining the Default Risk. Scientific Programming, 1-8. (2019)
37. Katarya, R., Jain, S.: Comparison of different machine learning models for diabetes detection. In International Conference on Advances and Developments in Electrical and Electronics Engineering (ICADEE), IEEE, 1-5. (2020)
38. Zhang, C., Liu, C., Zhang, X., Almpanidis, G.: An Up-to-Date Comparison of State-of-the-Art Classification Algorithms. Expert Systems with Applications, Vol. 82, 128-150. (2017)
39. Lessmann, S., Baesens, B., Seow, H.V., Thomas, L. C.: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, Vol. 247, 124–136. (2015)
40. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization, Vol. 752, No. 1, 41-48. (1998)
41. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine learning, Vol. 39, No. 2, 103-134. (2000)
42. Raina, R., Shen, Y., Mccallum, A., Ng, A.: Classification with hybrid generative/discriminative models. Advances in neural information processing systems, Vol. 16. (2003)

43. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In International Conference on Machine Learning 2001, 282-289. (2001)
44. Peng, L., Yang, B., Chen, Y., Abraham, A.: Data Gravitation based Classification. Information Sciences, Vol. 179, No. 6, 809-819. (2009)
45. Cano, A., Zafra, A., Ventura, S.: Weighted Data Gravitation Classification for Standard and Imbalanced Data. IEEE Transactions on Cybernetics, Vol. 43, No. 6, 1672-1687. (2013)
46. Sathya Bama, S.S., Saravanan, A.: Efficient Classification using Average Weighted Pattern Score with Attribute Rank based Feature Selection. International Journal of Intelligent Systems and Applications, Vol. 10, No. 7, 29. (2019)
47. Xu, W., Jiang, L., Yu, L.: An Attribute Value Frequency-based Instance Weighting Filter for Naive Bayes. Journal of Experimental & Theoretical Artificial Intelligence, Vol. 31, No. 2, 225-236. (2019)
48. Jiang, L., Zhang, L., Li, C., Wu, J.: A Correlation-based Feature Weighting Filter for Naive Bayes. IEEE Transactions on Knowledge and Data Engineering, Vol. 31, No. 2, 201-213. (2018)
49. Zhang, H., Jiang, L., Yu, L.: Attribute and Instance Weighted Naive Bayes. Pattern Recognition, Vol. 111, 107674. (2021)
50. Jiang, L., Wang, D., Cai, Z.: Discriminatively weighted naive Bayes and its application in text classification. International Journal on Artificial Intelligence Tools, Vol. 21, No. 01, p.1250007 (2012).
51. Zhang, H., Jiang, L., Li, C.: Collaboratively weighted naive Bayes. Knowledge and Information Systems, Vol. 63, No. 12, 3159-3182. (2021)
52. Yu, L., Gan, S., Chen, Y., Luo, D.: A Novel Hybrid Approach: Instance Weighted Hidden Naive Bayes. Mathematics, Vol. 9, No. 22, 2982. (2021)
53. Vajapeyam, S.: Understanding Shannon's Entropy Metric for Information. arXiv Preprint arXiv:1405.2061. (2014)
54. Glen. S., Maximum Entropy Principle: Definition, From StatisticsHowTo.com: Elementary Statistics for the rest of us! https://www.statisticshowto.com/maximum-entropy-principle/
55. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In workshop on machine learning for information filtering, Vol. 1, No. 1, 61-67. (1999)
56. Lazebnik, S., Schmid, C., Ponce, J.: A maximum entropy framework for part-based texture and object recognition. In IEEE International Conference on Computer Vision, Vol. 1, 832-838. (2005)
57. Feng, Z., Zhou, Y., Wu, L., Li, Z.: Audio classification based on maximum entropy model. International Conference on Multimedia and Expo, IEEE. Vol. 1, pp. I-745. (2003)
58. Alrashdi, I., Siddiqi, M.H., Alhwaiti, Y., Alruwaili, M., Azad, M.: Maximum entropy Markov model for human activity recognition using depth camera. IEEE Access, Vol. 9, 160635-160645 (2021).

59. Arunraj, G, Radha, B.: Feature Selection using Multiple Ranks with Majority Vote Based Relative Aggregate Scoring Model for Parkinson Dataset. International Conference on Data Science and Applications. (2021)
60. Sim, J., Lee, J.S., Kwon, O.: Missing Values and Optimal Selection of an Imputation Method and Classification Algorithm to Improve the Accuracy of Ubiquitous Computing Applications. Mathematical Problems in Engineering, pp. 1-14. (2015)
61. Yang, Y., Webb G.I., Wu X.: Discretization Methods. In: Editor, Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA, 101-116. (2010)
62. Jo, J.M.: Effectiveness of normalization pre-processing of big data to the machine learning performance. The Journal of the Korea Institute of Electronic Communication Sciences, Vol. 14, No. 3, 547-552. (2019)
63. Thomas M. Cover and Joy A. Thomas, Elements of Information Theory, Second Edition, John Wiley. (2006)
64. Frank, A. Asuncion, UCI Machine Learning Repository, Univ. California, School Inf. Comput. Sci., Irvine, CA. [Online]. Available: http://archive.ics.uci.edu/ml/citation_policy.html
65. Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL Data Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework, J. Multiple-Valued Logic Soft Comput., Vol. 17, 255–287. (2011).

Arumugam Saravanan completed his Ph.D. in Computer Applications from Anna University, Chennai. He is currently Associate Professor, Department of Computing, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India. He has 23 years of experience teaching undergraduate and graduate students, as well as 12 years of research experience. He has published more than 30 papers in national and international journals as well as at conferences. His areas of interest and research include web security, network security, machine learning, web mining, and database management.

Damotharan Anandhi received her Doctoral Degree in computer science at Bharathiar University, Coimbatore, Tamil Nadu, India. Presently she works as an assistant professor at the Coimbatore Institute of Technology in Coimbatore, Tamil Nadu, India. She has 18 years of job experience in the field of education. She has published a number of research articles in prestigious national and international journals.

Marudhachalam Srividya received her doctoral degree in computer science at Bharathiar University, Coimbatore. She is currently an Assistant Professor at the Coimbatore Institute of Technology, Coimbatore, India, and has 18 years of work experience in the area of teaching. She has several research publications in well-known international journals and conferences. She has conducted many programmes at CIT, which include workshops and a short-term course under the QIP.

Received: October 30, 2021; Accepted: December 30, 2022.