International Journal of Computer Sciences and Engineering
Vol.11, Special Issue.1, pp.161-168, November 2023
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Research Paper

A Comparative Study of Popular Multiclass SVM Classification Techniques and Improvement over Directed Acyclic Graph SVM

Sanjib Saha[1]

[1]Dept. of Computer Science and Engineering, National Institute of Technology, Durgapur and Dr. B. C. Roy Engineering College, Durgapur, India

Author's Mail Id: sanjib.saha@bcrec.ac.in

Abstract: Multiclass classification using Support Vector Machine (SVM) is an ongoing research issue. SVM is mainly a binary classifier, but for classification efficiency, it is also used for multiclass classification. In multiclass classification, there are two or more classes and classification is not so easy. That's why many methods are introduced to extend the classification efficiency of SVM. Directed Acyclic Graph Support Vector Machine (DAGSVM), Binary Tree of Support Vector Machine (BTS) and Error Correcting Output Codes (ECOC) methods are more favourable because of their computation efficiency. In the case of DAGSVM there are many improved methods like Decision Directed Acyclic Graph (DDAG), Divide-by-2 (DB2), and Weighted Directed Acyclic Graph of Support Vector Machine (WDAG SVM) have been developed. The BTS-based methods are SVM with Binary Tree Architecture, and Adaptive Binary Tree (ABT). There are many methods related to ECOC like One-Per-Class (OPC), Discriminant Error Correcting Output Codes (DECOC), and Adaptive ECOC. This paper presented a comparative and analytical survey of those methods and introduces a new model which is an improvement over the existing DAGSVM methods. This model uses Gaussian Mixture Model, K-Means Clustering and Naïve Bayes Classifier for data classification. This model can give better results than existing DAGSVM methods.

Keywords: Multiclass SVM, Directed Acyclic Graph SVM, Binary Tree SVM, Error Correcting Output Codes.

1. Introduction

For multiclass classification, SVM [1-10] is the most popular method though it is mainly a binary classifier. The multiclass classification problem has not any easy solution that's why researcher selected the SVM for this because its classification efficiency is better than any method. But the SVM is mainly a two class classifier and in multiclass classification there is N number of classes, so the classification task is not so easy in this case by SVM. Because building N SVMs is the typical approach for N-class SVMs. Each example in the i^{th} class will have a positive label, while all other examples will have a negative label, in order to train the i^{th} SVM. The class which belongs to the SVM with the greatest output assessment is the ultimate result of a well-known approach called 1-v-r (one-versus-rest) of N. 1-v-1 SVMs (one-versus-one SVMs) are a different approach for creating N-class classifiers using SVMs. and the number of classifiers would be $N(N-1)/2$. In Friedman's Max Wins method, each 1-v-1 classifier casts one vote for the class it prefers, and the class with the highest number of votes wins out in the end. A major disadvantage of 1-v-1 is that the overall N-class classifier system will tend to over fit unless the individual classifiers are carefully regularized. Also this can be slow in case of evaluation of large problems because the size of 1-v-1 classifier may grow super-linearly with N. The Decision Directed Acyclic Graph (DDAG), a novel multiclass learning architecture, excels at multiclass classification.

2. Related Work and Background Method

For comparing the different methods of multiclass SVM classification we surveyed the different methods of Directed Acyclic graph of Support Vector Machine (DAGSVM), Binary Tree of Support Vector Machine (BTS) and Error Correcting Output Codes (ECOC).

2.1 Directed Acyclic Graph SVM

Multiple two-class classifiers are combined into one multiclass classifier using Decision Directed Acyclic Graphs (DDAG) [11]. DDAG requires $N(N-1)/2$ classifiers for a N class problem, one for each pair of classes. At each DDAG decision node, DAGSVM employs two-class maximum margin hyperplanes and works in a kernel-based feature space. The DAG is a graph with no cycles and oriented edges. Starting at the root node, a particular DDAG G is assessed for input $x \in X$ by evaluating the binary function at each node. If the binary function is zero, the left edge is used to leave the node; if it is one, the right edge is used. The nodes of DDAG are maximum margin classifiers over a kernel-induced featured space. This DDAG is obtained by training each i-node or j-node only on the subset of training points labelled by *i* or *j*. By using the large margin the DAGSVM separates the individual classes. The orders of classes are chosen arbitrarily in the DDAG for DAGSVM. The DAGSVM method is better than other multiclass SVM techniques respect to both training time and testing time. In the time of testing the Gaussian kernel gives better result. The quantity of kernel evaluations is a reliable indicator of assessment time. In the case of DAGSVM, the quantity of kernel evaluations is the total amount of distinct support vectors, averaged throughout the assessment routes taken by the test set through the DDAG. The DAGSVM has the fastest evaluation compared to standard method Max Wins and other methods. It is significantly quicker than the conventional 1-v-r SVM technique. The DAGSVM algorithm was put to the test against Friedman's Max Wins combination algorithm and the typical 1-v-r multiclass SVM technique. Although its accuracy is comparable to other methods but it improves over the training and testing time.

DAGSVM plays an important role in case of text classification [12]. This paper introduced a new method named PDHCS (P2P-based Distributed Hypertext Categorization System) and it is used for classifying the hypertext in peer-to-peer networks. DAGSVM framework and harmonic covering network can be combined to perform distributed hypertext categorization quickly in PDHCS. Web information extraction and web mining use hypertext classification and it is difficult to build hypertext classifiers by hand and it is time consuming also. That's why classifiers are learned from labelled categorization examples. Therefore, in this instance, PDHCS was more effective because to the parallel structure of DAGSVM.

A new framework is introduced named Divide-by-2 (DB2) to extend SVM for multiclass problems [13]. It is the replacement of traditional one-against-one and one-against-all methods. In order to generate an *N-1* nodes binary decision tree for a *N* class problem, DB2 uses *N-1* SVM binary classifiers to form the nodes. DB2 has a flexible tree structure and it can adjust with different types of multiclass problems. DB2 is always faster than one-against-one and one-against-all respect to test time and it is also faster than DAGSVM in case of unbalanced data. As future work we can combine DB2 with other existing multi-class methods and create a hybrid structure.

It is the method for performance improvement over the DAGSVM [14]. It divides the input space into many subspaces using a weighted multiclass classification approach. DAG SVM, where *n* is the number of classes, in the training phase finds *N(N-1)/2* hyperplanes related to pair-wise SVMs, and in the testing stage uses a decision tree to handle the problem of unclassifiable regions. A DAGSVM learns during training, and its probability density function is computed as well. To determine the class label of the specified input pattern, a fusion process is designed and useful to the DAG SVM evaluations. Here a new method is introduced named Weighted DAGSVM (WDAG SVM) classifier. Instead of using a single classifier, it applies many expert systems and separates the input space into subspaces. The key benefit of this approach is that each SVM's capability for each respective subspace is boosted. Additionally, it makes it possible to train SVMs using a lot of training data. By using power of classifier it gives better result in mixing. The number of clusters must be determined at the time of the experiment that each cluster contains data points from all of the classes. The train data in each cluster are smaller in size, and the calculation on them is quicker because the data in each cluster is not dependent on others, making the suggested method more efficient in terms of space and time. Following the clustering algorithm's division of the samples into parts, a DAG SVM is trained for each portion, and the weighting technique aids in the fusing of many learned SVMs. While applied the DAGSVM to standard iris and wine data sets it gives 17% extra recognition rate over DAG SVM. Lastly here determining the fusion operation is an important factor for the algorithm results. Finally, based on the experimental findings, WDAGSVM outperforms DAG SVM in multi-class classification problems in terms of accuracy.

The concept of misclassification is offered for DAG-SVM structure selection, and the classification unfairness of DAG-SVM is examined [15]. On the basis of the idea of Misclassification Cost Minimization, a structure selection approach for DAGSVM is suggested for reducing the probability of total misclassification cost. As the decision tree structure in an application, it chooses the structure with the lowest overall risk of misclassification cost. This method has been tested on three benchmark datasets and gives satisfactory results. Lastly it has also some drawbacks- i. It requires the classifiers which are with similar classification abilities. ii. The definition of the Misclassification Cost Matrix requires professional knowledge.

In pair wise classification interclass generalization problems can be treated [16]. The classification outcome in this scenario shouldn't be impacted by the arrangement of the two input examples. In the context of SVMs, specific kernels and symmetric training sets have been recommended for addressing large pair-wise classification issues. When a balanced kernel is used to a pair-wise SVM, a symmetric decision function is always produced. Any symmetric set of training pairings is said to result in a symmetric decision function for pair-wise SVMs. It is demonstrated how successfully pair-wise SVMs fight for a real-world data set.

A new method is proposed to resolve unclassifiable regions for pair wise support vector machines [17]. However, the tree structure of this architecture affects the ability to generalize. So for improving the generalization ability the structure is to be minimized. As a result of it the class pairs with higher generalization abilities are put in the upper nodes of the tree. For pair wise classification, training time is the same for fuzzy SVMs and DDAGs. But in time of classification the DDAG takes less time and it gives better accuracy Fuzzy SVM.

2.2 Binary Tree of SVM

The SVM-based binary tree benefits from both the high classification accuracy of SVMs and the effective computation of tree construction [18]. Despite the fact that *(N-1)* SVMs must be trained for an *N*-class problem, testing log_2N SVMs works for the classification decision. The SVM with Binary Tree Architecture uses a hierarchical binary tree, where each node uses an SVM to make a binary decision, to solve an *N*-class pattern recognition problem. The multi-class problems are transformed into binary trees using Kernel-based Self-Organizing Map (KSOM), where binary predictions are determined by SVMs. The performance reduction of the tree structure is solved by permitting overlaps in binary decision trees. Comparable to the majority of popular multiclass approaches, its classification accuracy is high. Because it is built on a tree architecture, it takes substantially less computing when making decisions. The Kernel-based SOM can be used to convert multi-class problems into appropriate binary trees, which is an essential step. The adaptation of multiclass problems into appropriate binary trees is automatically done by capitalizing on the distributing estimation at the kernel space.

In the existing multi-class SVMs respect to decision making speed, the quickest is multi-class support vector machine with binary tree architecture (SVM-BTA). [19]. But it is not very capable of classification. For the purpose of including several binary SVMs, this study employs the resemblance coefficient method to automatically generate a binary tree. The multi-class SVMs with built-in binary trees are capable of making quick decisions and have strong classification abilities. In this study, the RCA (Resemblance Coefficient Construction Algorithm) is utilized to transform multi-class classification problems into binary-class problems in each layer of the binary tree architecture. Because the binary tree structure was used, the binary tree architecture has high classification capabilities. Additionally, the binary tree-based SVM classifier makes decisions more quickly than other popular multi-class SVM classifiers like the OAA, OAO, DAG, and BUBT (Bottom-up Binary Tree).

Significant classification accuracy for multiclass problems is achieved using the binary tree of support vector machine (BTS) [20]. Binary classifiers are less with BTS and its improved variant, c-BTS. In order to make a choice, *log 4/3((N+3)/4)* number of tests must be constructed. In issues with a large number of classes, it performs substantially quicker than the directed acyclic graph SVM (DAGSVM) and ECOC due to its log complexity. Although BTS works on binary classifications, it reduces the quantity of binary classifiers to the maximum level while keeping the original problem's complexity constant. In contrast to putting several classes together, BTS chooses two classes for training in each node. The average convergent efficiency of BTS is *log4/3((N+3)/4)*. BTS continues to use a one-to-one training approach in order to maintain the original complexity and achieve decision robustness. The tree c-BTS is distinct. It is clear that c-BTS is more balanced than a typical BTS and that it makes decisions more quickly.

The multiclass classification problem of hyper spectral data is solved using a novel Pair-wise Decision Tree of SVM (PDTSVM) approach [21]. For an *N*-class problem, it requires *N-1* binary SVMs for one classification. When choosing a binary SVM, PDTSVM avoids incorrect votes from the one-against-one approach and also has a lot less layers than other tree-based methods that's why it enable to decrease accumulated errors. This approach skips the tree-building phase and shares the training phase with OAO. The experiments with varying numbers of training samples demonstrate that classification accuracy rises as the number of training examples rises, although the training and testing processes slow down in these experiments.

The classifiers arranged with a binary tree structure to solve the multiclass problem with increased efficiency [22]. The multi-class problem is transformed using the clustering approach into a binary tree, where the binary assessments are determined by SVMs. The new clustering model makes use of kernel-space distance measurements rather than input-space distance measurements. The novel Support Vector Machines in Binary Tree Architecture (SVM-BTA) method uses decision tree architecture to provide superior recognition speed while maintaining a similar recognition rate to the other methods that are already in use. Four separate datasets of handwritten numbers and letters were used in the experiment. and it shown the fastest training times. When the number of classes in the problem increases it performs very well.

The multi-class support vector machine test computational complexity is to be reduced using an Adaptive Binary Tree [23]. The two methods it uses to achieve fast classification are choosing the binary SVMs with the lowest average quantity of support vectors and reducing the number of binary SVMs for each classification by using the separating planes of certain binary SVMs to distinguish different binary problems. It is simple to construct the adaptive binary tree by having each internal node choose the binary SVM with the fewest average quantity of support vectors. ABT requires to train *N(N-1)/2* binary SVMs for an *N* classes problem, but just *N-1* binary SVMs for a problem with one classification. The testing time of the ABT is superior to OAA, OAO, A&O, DAGSVM, and BTS, although the differences in accuracy between these methods are minimal.

To achieve significant classification accuracy for multiclass problems a innovative architecture called Efficient Binary Tree Multiclass SVM (EBTSVM) is presented [24]. The suggested approach uses genetic algorithms to construct a binary tree for multiclass SVM with the goal of finding optimal partitions for the ideal tree. It is significantly quicker than other methods in problems with a large class number because of its Log complexity. In comparison to previous methods, this strategy reduces the training execution time and the quantity of support vectors, which affect how long the test execution occurs.

The Binary Decision Tree (BDT) of SVM employed to solve the multiclass problem, which benefits from both the decision tree design's efficient computation and SVMs' excellent classification accuracy [25]. This study examines the impact of various clustering distance measurements on the SVM-BDT's ability to predict outcomes. Here, the three distance measures of Euclidian, Standardized Euclidean, and Mahalanobis are taken into consideration. Here, the performance of the SVM-BDT with various distances is assessed using five distinct datasets. After testing, it was discovered that, in comparison to other distance measures used by SVM-BDT methods, Mahalanobis Distance is most suited for determining how similar classes are throughout the clustering phase. However, in terms of temporal complexity, it is inferior to other methods of measuring distance, such as Euclidean. The SVM-BDT approaches with different distance measures exhibit comparable results to or provide higher accuracy than the other multiclass methods when compared to other SVM and non-SVM based methods. The employment of Euclidean Distance for evaluating the similarity between classes in the clustering procedure of building the classifier architecture, the accuracy of training and testing was increased.

2.3 Error Correcting Output Codes

Four approaches to multiclass learning problems are compared, these are multiclass decision trees, the one-per-class (OPC), the meaningful distributed output approach and the ECOC [26]. The experiment results show that ECOC performs better than the other three methods. If the size of the training sample changes, the new technique performs better. The error-correcting code approach can deliver accurate estimations of class probability. Additionally, it offers a general approach to enhance the effectiveness of inductive learning programs when dealing with multiclass problems. The improvements of ECOC were observed in six of eight domains with decision tree and three of five domains with back-propagation. The ECOC improves decision trees and neural networks, and it also improves over the small sample size and lastly the improvement does not depend on particular assignment of code-words to classes. It also performs better than one-per-class method in case of scaling neural networks to very large classification problems.

An application of ECOCs for the multiclass classification is proposed where samples in the database were derived from AVHRR (Advanced Very High Resolution Radiometer) images [27]. Then a nearest neighbor classifier is modified by replacing each sample's class label with a set of "output" functions that each separates the set of classes differently. Then simple sequential feature selection algorithm FSS is used to select a different set of features for each output function for computing the distance differently for each bit. This method reduces both variance and bias errors. It also performs well in increasing accuracy on most confusable classes in data set. But ECOC is more computationally expensive compared to monolithic class labels and the cost increases for the number of bits used in the ECOC representation.

Automatic face verification is primarily a two-class problem, and it is demonstrated how to apply the ECOC approach to this problem [28]. The ECOC feature space, which makes it simpler to distinguish between modified patterns identifying clients and impostors, is defined by the result of the binary classifiers. Here, two distinct combination procedures are suggested as the face verification matching score. First order Minkowski metric, that needs a threshold to be specified, is the first approach, whereas kernel-based methods don't need any parameters to be set. In order to solve the multi-class recognition problem given by the ECOC matrix, the novel method first suggests a solution that creates a discriminant and then to verify the generated discriminant is checked for consistency with the distribution of responses for the particular client. In comparison to previously published results, the novel technique performs better on the well-known XM2VTS data set.

Decoding and mode selection are two unresolved issues with ECOC that are covered [29]. How to convert the classifier outputs into class code-words is the subject of the decoding problem. The margins are combined using a forecast of their class conditional probabilities in this paper's novel decoding algorithm. A new theoretical result that limits the leave-one-out (LOO) error of kernel machines is offered here for model selection. This result can be utilized to adjust the kernel hyper-parameters. This experiment demonstrates that, in compared to alternative loss-based decoding approaches, converting margins into conditional probabilities assists in recalibrating the classifiers' outputs and also improves the overall multiclass classification accuracy. LOO error bound allows kernel parameters to be successfully modified while simultaneously increasing classification accuracy.

For the purpose of creating compact error-correcting output codes, a novel approach called discriminant ECOC is put forth [30]. Because it employs fewer classifiers and takes less training time during maintenance, the resulting multiclass classifier operates faster. The effectiveness of the remaining ECOC approaches has likewise increased. This technology is also the first to address the issue of designing application-dependent discrete ECOC matrices satisfactorily. Both the UCI database and a practical computer vision application, traffic sign identification, successfully use the discriminant ECOC method. When compared to other ECOC techniques, the discriminant ECOC is quite helpful.

Error Correcting Output Codes (ECOC) is a new technique for multi-view face detection [31]. How to create efficient binary classifiers is one of the primary problems with ECOC-based multi-class classifiers. The fundamental premise is that face patterns can be classified into distinct classes across views, and the ECOC multi-class approach, which is inherently error-tolerant, can enhance multi-view face detection's adaptability over view-based methods. This study places additional emphasis on creating effective binary classifiers by studying informative features and aiming to reduce the ensemble ECOC multi-class classifier's error rate. By learning useful features by reducing mistake rates, the new approach builds binary classifiers. Numerous tests demonstrate that the new method is effective at concurrently detecting faces from several views and that ECOC increases the robustness of face detection against position change. Comparing the novel technique to cutting-edge methods for multi-view face detection, it performs well.

For multi-class learning problems, ECOC is a successful method of combining a number of binary classifiers [32]. The ECOC model is reformed in this study from the viewpoint of multi-task learning, where the binary classifiers are learnt from a shared subspace of data, and a new approach is suggested named adaptive generalization of the ECOC framework, or AECOC. The suggested approach simultaneously improves binary classifiers and data representation. AECOC simultaneously learns the base classifiers and the inherent representation of the input, which creates a connection between the ECOC frameworks and multi-task learning. To deal with complex data, the new model's kernel extension is introduced. The experiment using 14 datasets from the UCI and USPS handwritten digits recognition program shows that it is more effective than cutting-edge dimensionality reduction procedures, ECOC methods, multi-task learning approaches, and deep learning techniques.

4. Proposed Method

After reviewing the papers we focused on improvement of the DAGSVM. Then we found that an improvement over DAGSVM is already exists, then we want to modify the existing model to get better result. To propose a new model various concepts are required and here a brief knowledge is given.

4.1 Gaussian Mixture Model

A Gaussian Mixture Model (GMM) [33] is a parametric probability density function and it is presented by sum of weights of Gaussian component densities. The parameters of GMM are estimated from training data by using iterative Expectation-Maximization (EM) [34] algorithm or *Maximum APosteriori* (MAP) estimation from a well-trained prior model. The Gaussian Mixture Model can be represented by the following equation 1:

$$p(x|\,\lambda) = \sum_{j=1}^{n} w_j\, g(x|\,\mu_j,\Sigma_j) \qquad (1)$$

Where, x is a d-dimensional continuous-valued data vector (or measurement of features), $w_j, j = 1 \dots n,$ are the mixture of weights, and $g(x|\,\mu_j,\Sigma_j), j = 1 \dots n,$ are the component Gaussian densities. Each component density is a d-variate Gaussian function of the equation 2,

$$g(x|\,\mu_j,\Sigma_j) = \exp\{-1/2(x-\mu_j)'\Sigma_j^{-1}(x-\mu_j)\}/(2\pi)^{d/2}\,|\Sigma_j|^{1/2} \qquad (2)$$

Where, mean vector μ_j and covariance matrix Σ_j. The mixture weights satisfy the constraint that $\sum_{j=1}^{n} w_j = 1$.

The parameters mean vectors, covariance matrices and mixture weights from all component densities are represented by following equation 3:

$$\lambda = \{w_i, \mu_i, \textstyle\sum_i\} \quad i=1,\dots,n. \qquad (3)$$

4.2 K-Means Clustering

The K-Means [35] method for partitioning, where each cluster centre is shown using the mean value of the cluster's objects.

Input of K-Means clustering:

- D: A data set containing *n* objects,
- K: The number of clusters.

Output of K-Means clustering:

- A set of K clusters.

Method of K-Means clustering:

1) Select K objects at random from D to serve as the first cluster centres;
2) Repeat
3) Based on the mean value of the objects in the cluster, (Re)assign each object to the cluster to which it is most comparable;
4) Update the cluster means by determining the average value of the objects in each cluster;
5) Until there is no change.

4.3 Naive Bayes Classification

The Bayes Naive [36] chooses the probable classification v_{nb} from the given attribute values $a_1, a_2, \dots a_n$. The Bayes Naive classifier can be represented by the following equation 4:

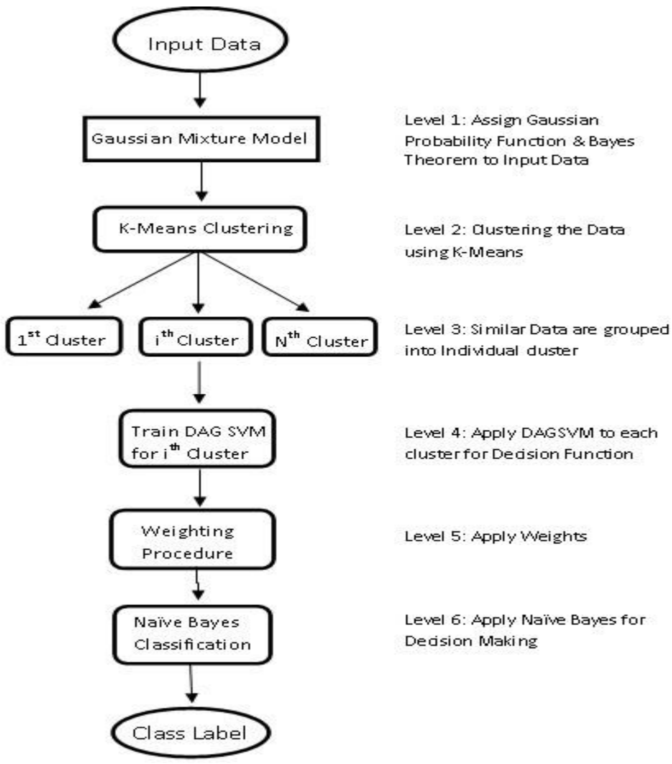
$$v_{nb} = \arg\max_{v_j \in V} P(v_j) \prod P(a_i \mid v_j) \qquad (4)$$

The probability estimation $P(a_i|v_j)$ is calculated in equation 5:

$$P(a_i \mid v_j) = \frac{n_c + mp}{n + m} \qquad (5)$$

Where n = no. of training samples for which $v = v_j$,
n_c = no. of samples for which $v = v_j$ and $a = a_i$,
p = a priori estimate for $P(a_i|v_j)$,
m = total no. of samples.

4.4 Flowchart of Proposed Model



4.5 Proposed Algorithm

Step1: Takes the input data having mixed attributes and missing values.

Step2: On input data Gaussian probability function and Bayes theorem is assigned, because we know that Gaussian probability density function is very useful in case of data having missing values.

Step3: Then we applied most popular and simple clustering method K-Means algorithm to the data.

Step4: After applying the K-Means algorithm similar data are grouped into individual cluster.

Step5: Apply DAGSVM to each individual cluster for decision function and train the DAGSVM for every individual cluster.

Step6: Then apply weight to the each individual cluster to make the classification more accurate.

Step7: Apply Naive Bayes classification method to make the decision for dataset.

Step8: After applying the Naive Bayes classification method the dataset are labeled with correct class.

In proposed work a new model is introduced which is an improvement over the DAGSVM. This model may give better performance than the existing DAGSVM approaches. The proposed model combined training and testing phase into one procedure. In proposed model at first the Gaussian probability function and Bayes Theorem is assigned on the inputted data and then applied K-Means algorithm to clustering the data. We know that Gaussian Mixture Model is very useful in case of data having missing values and mixed attributes. When the K-Means algorithm is applied to the data after assigning the Gaussian probability function the accuracy in clustering will increases automatically so the K-Means algorithm grouped the similar data into individual cluster easily and gives better result. After clustering DAG is applied to each cluster for decision function and train the DAGSVM for every cluster. For easy decision making weight is applied to each cluster. The Naive Bayes classification method is applied for decision making because it selects the most likely classification. At last the data are labelled with correct class label. In this way a better practical model can be introduced which improves over than DAGSVM for multiclass classification.

5. Results and Discussion

Table 1. Comparison of Multiclass SVM Classification

Methods	**Parameters**			
	Training Time (Sec)	**Testing Time (Sec)**	**Accuracy (%)**	**Time Complexity**
DAG SVM	32.3 (Glass)	2.6 (Glass)	73.8 (Glass)	Less than Max Wins
DB2	48.7 (Glass)	2.1 (Glass)	73.5 (Glass)	Less than DAG SVM
OAO	31.0 (Glass)	6.3 (Glass)	72.0 (Glass)	Greater than DAG SVM
WDAG SVM	0.64 (Wine)	Less than DAG SVM	Greater than DAG SVM	Less than DAG SVM
RCBTA	Less than DAG SVM	92.84	Greater than DAG SVM	Less than DAG SVM
BTS	Less than DAG SVM & ECOC	Less than DAGSVM & ECOC	97.5 (Optdigit)	Log_2N (N=Number of classes)
c-BTS	Less than BTS	Less than BTS	97.2732 (Optdigit)	Less than BTS
PDT SVM	Less than OAO	Less than OAO	Less than BTS	Less than OAO
SVM-BTA	1.62 (Pendigit	0.61 (Pendigit)	Same as BTS	Less than BTS
ABT	Less than BTS	Less than BTS	99.92 (Shuttle)	Less than BTS
AECOC	0.0081 (Iris)	Less than ECOC	Greater than ECOC	Less than ECOC
Proposed Method	68.2 (Glass)	3.5 (Glass)	74.6 (Glass)	Greater than DAG SVM

The proposed method of modified DAGSVM with K-Means and Naïve Bayes is implemented using Weka tool [37]. The proposed model has achieved classification accuracy of 74.6% on Glass dataset of UCI machine learning database [38] repository. The performance of the proposed method is compared with state-of-the-art methods as shown in Table 1. It is observed that proposed modified DAGSVM is superior.

In multiclass classification approach the existing methods are One-Against-One (OAO), One-Against-All (OAA) etc. then the DAGSVM come to minimize the computational complexity of multiclass classification. For an N class problem Decision Directed Acyclic Graph (DDAG) needs $N(N-1)/2$ classifiers, one for each pair of classes. For combining many two classes classifier into a multiclass classifier DDAG is used. DAGSVM is very useful in text classification. For text classification the method PDHCS is used. To improve the performance of DAGSVM, Weighted DAGSVM (WDAG SVM) classifier is introduced. Instead of using a single classifier, it applies many expert systems and separates the input space into subspaces. It has the ability to train SVMs using a lot of training data.

To extend the performance of DAGSVM binary tree approach of SVM is introduced. For taking classification decision it requires only log2N SVMs. To achieve more classification accuracy resemblance coefficient method for constructing automatically binary tree is used. For solving the multiclass classification problem of hyper spectral data

PDTSVM is used to get better performance. To reduce the test computational complexity of binary tree SVM, adaptive binary tree approach is introduced. It is faster respect to test time compared to OAA, OAO, A&O, DAGSVM, and BTS.

To solve the multiclass problem there is another method, i.e. Error Correcting Output Codes (ECOC). This technique can provide reliable class probability estimates. It is very useful in cloud classification. It is also applied to automatically face verification problem and gives satisfactory result. In case of decoding problem it gives a new result by providing a new decoding function. To design compact error correcting output codes a new algorithm discriminant ECOC is introduced. The algorithm takes less training time that's why it runs faster and improves the ECOC method. To optimize the representation of data as well as the binary classifiers Adaptive Error-Correcting Output Codes (AECOC) is introduced. It improves over the existing ECOC.

6. Conclusion

After review this papers we have got different type of techniques and their applications. In case of DAGSVM it is much faster than Friedman's Max Wins combination algorithm. It is very useful in text classification. To improve the performance of DAGSVM a new method also presented named Improved DAGSVM. It introduced an approach called Weighted SVM (WSVM). In case of BTS, *(N-1)* SVMs need to be trained for an N-class problem. Here, multi-class problems are transformed into binary trees using Kernel-based Self-Organizing Map (KSOM), where binary decisions are generated by SVMs. To improve the classification accuracy of SVM with Binary Tree Architecture Automatic Construction Algorithm is introduced. Here RCA (Resemblance Coefficient Construction Algorithm) is used to convert multi-class classification problems into binary-class classification problems. To reduce the test computational complexity of multi-class support vector machine (SVM) Adaptive Binary Tree (ABT) is proposed. ABT only requires *N-1* binary SVMs to train for a single classification, but *N(N-1)/2* binary SVMs for a problem with *N* classes. Another approach to solve multiclass problem is Error Correcting Output Codes (ECOC). It performs better than one-per-class method in case of scaling neural networks to very large classification problems. In the area of cloud classification it gives satisfactory results. Additionally, it works well for facial verification. A better solution than existing methods is provided by adaptive error-correcting output codes (AECOC), which creates a connection between multitask learning and ECOC frameworks for multiclass learning problems. From these approaches it can be said that though multiclass classification problem is not so easy but there are different and very useful approaches to solve this type of problem and here in the proposed new model training and testing time can be minimized. In the new model Gaussian probability function is applied and to cluster the dataset K-Means algorithm is used and for decision making Naive Bayes Classification method is applied. This can be an improvement over the existing DAGSVM. It will take less training and testing time than other methods. To get better performance the correct parameter should be selected. The proposed method of modified DAGSVM with K-Means and Naïve Bayes has achieved classification accuracy of 74.6% on Glass dataset of UCI machine learning database repository.

Conflict of Interest
There is no conflict of interest for this article.

Funding Source
There is no funding source for this article.

Authors' Contributions
The single author of this article is the only contributor.

Acknowledgements
The author is thankful to his parents, wife, and daughter for their support.

References

[1]. Zhang, Xian-Da, and Xian-Da Zhang. "Support vector machines." A Matrix Algebra Approach to Artificial Intelligence: pp.**617-679, 2020.**
[2]. James, Gareth, et al. "Support vector machines." An introduction to statistical learning: with applications in R: pp.**367-402, 2021.**
[3]. Cervantes, Jair, et al. "A comprehensive survey on support vector machine classification: Applications, challenges and trends." Neurocomputing 408: pp.**189-215, 2020.**
[4]. Schölkopf, Bernhard, Alexander J. Smola, and Francis Bach. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, **2002.**
[5]. Nalepa, Jakub, and Michal Kawulok. "Selecting training sets for support vector machines: a review." Artificial Intelligence Review 52.2: pp.**857-900, 2019.**
[6]. Das, Subhankar, and Sanjib Saha. "Data mining and soft computing using support vector machine: A survey." International Journal of Computer Applications 77.14, **2013.**
[7]. Datta, R. P., and Sanjib Saha. "Applying rule-based classification techniques to medical databases: an empirical study." International Journal of Business Intelligence and Systems Engineering 1.1: pp.**32-48, 2016.**
[8]. Saha, Sanjib, and Debashis Nandi. "Data Classification based on Decision Tree, Rule Generation, Bayes and Statistical Methods: An Empirical Comparison." Int. J. Comput. Appl 129.7: pp.**36-41, 2015.**
[9]. Saha, Sanjib. "Non-rigid Registration of De-noised Ultrasound Breast Tumors in Image Guided Breast-Conserving Surgery." Intelligent Systems and Human Machine Collaboration. Springer, Singapore, pp.**191-206, 2023.**
[10]. Saha, Sanjib, et al. "ADU-Net: An Attention Dense U-Net based deep supervised DNN for automated lesion segmentation of COVID-19 from chest CT images." Biomedical Signal Processing and Control 85: 104974, **2023.**
[11]. Platt, John, Nello Cristianini, and John Shawe-Taylor. "Large margin DAGs for multiclass classification." Advances in neural information processing systems 12, **1999.**
[12]. Vural, Volkan, and Jennifer G. Dy. "A hierarchical method for multi-class support vector machines." Proceedings of the twenty-first international conference on Machine learning. **2004.**
[13]. Fei, Liu, et al. "A peer-to-peer hypertext categorization using directed acyclic graph support vector machines." Parallel and Distributed Computing: Applications and Technologies: 5th International Conference, PDCAT 2004, Singapore, December 8-10, 2004. Proceedings. Springer Berlin Heidelberg, **2005.**
[14]. Sabzekar, Mostafa, et al. "Improved DAG SVM: A New Method

for Multi-Class SVM Classification." IC-AI. **2009.**
[15]. Yi, Hui, Xiaofeng Song, and Bin Jiang. "Structure selection for DAG-SVM based on misclassification cost minimization." International Journal of Innovative Computing, Information & Control 7.9: pp.**5133-5143, 2011.**
[16]. Brunner, Carl, et al. "Pairwise support vector machines and their application to large scale problems." The Journal of Machine Learning Research 13.1 (2012): 2279-2292, ©2012 Carl Brunner, Andreas Fischer, Klaus Luig and Thorsten Thies.
[17]. Takahashi, Fumitake, and Shigeo Abe. "Optimizing directed acyclic graph support vector machines." Artificial Neural Networks in Pattern Recognition (ANNPR): pp.**166-173, 2003.**
[18]. Cheong, Sungmoon, Sang Hoon Oh, and Soo-Young Lee. "Support vector machines with binary tree architecture for multi-class classification." Neural Information Processing-Letters and Reviews 2.3: pp.**47-51, 2004.**
[19]. Zhang, Gexiang, and Weidong Jin. "Automatic construction algorithm for multi-class support vector machines with binary tree architecture." International Journal of Computer Science and Network Security 6.2A: pp.**119-126, 2006.**
[20]. Fei, Ben, and Jinbai Liu. "Binary tree of SVM: a new fast multiclass training and classification algorithm." IEEE transactions on neural networks 17.3: pp.**696-704, 2006.**
[21]. Chen, Jin, Cheng Wang, and Runsheng Wang. "Combining support vector machines with a pairwise decision tree." IEEE Geoscience and Remote Sensing Letters 5.3: pp.**409-413, 2008.**
[22]. Madzarov, Gjorgji, and Dejan Gjorgjevikj. "Multi-class classification using support vector machines in decision tree architecture." IEEE EUROCON 2009. IEEE, **2009.**
[23]. Chen, Jin, Cheng Wang, and Runsheng Wang. "Adaptive binary tree for fast SVM multiclass classification." Neurocomputing 72.13-15: pp.**3370-3375, 2009.**
[24]. Sidaoui, Boutkhil, and Kaddour Sadouni. "Efficient binary tree multiclass svm using genetic algorithms for vowels recognition." Proceedings of the 10th WSEAS international conference on Computational Intelligence, Man-Machine Systems and Cybernetics, and proceedings of the 10th WSEAS international conference on Information Security and Privacy. **2011.**
[25]. Madzarov, Gjorgji, and Dejan Gjorgjevikj. "Evaluation of distance measures for multi-class classification in binary svm decision tree." Artificial Intelligence and Soft Computing: 10th International Conference, ICAISC 2010, Zakopane, Poland, June 13-17, 2010, Part I 10. Springer Berlin Heidelberg, **2010.**
[26]. Dietterich, Thomas G., and Ghulum Bakiri. "Solving multiclass learning problems via error-correcting output codes." Journal of artificial intelligence research 2: pp.**263-286, 1994.**
[27]. Aha, Divid W., and Richard L. Bankert. "Cloud classification using error-correcting output codes." Ai Applications 11.1: pp.**13-28, 1997.**
[28]. Kittler, Josef, et al. "Face verification using error correcting output codes." Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. Vol.1. IEEE, **2001.**
[29]. Passerini, Andrea, Massimiliano Pontil, and Paolo Frasconi. "New results on error correcting output codes of kernel machines." IEEE transactions on neural networks 15.1: pp.**45-54, 2004.**
[30]. Pujol, Oriol, Petia Radeva, and Jordi Vitria. "Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes." IEEE Transactions on Pattern Analysis and Machine Intelligence 28.6 (2006): 1007-1012.
[31]. Zhang, Hongming, et al. "Robust multi-view face detection using error correcting output codes." Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV 9. Springer Berlin Heidelberg, **2006.**
[32]. Zhong, Guoqiang, and Mohamed Cheriet. "Adaptive error-correcting output codes." Twenty-Third International Joint Conference on Artificial Intelligence. **2013.**
[33]. Reynolds, Douglas A. "Gaussian mixture models." Encyclopedia of biometrics 741.pp.**659-663, 2009.**
[34]. Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the royal statistical society: series B (methodological) 39.1 (1977): pp.**1-22, 1977.**
[35]. MacQueen, J. "Classification and analysis of multivariate observations." 5th Berkeley Symp. Math. Statist. Probability. Los Angeles LA USA: University of California, **1967.**
[36]. Murphy, Kevin P. "Naive bayes classifiers." University of British Columbia 18.60: pp.**1-8, 2006.**
[37]. WEKA3 tool for machine learning and knowledge analysis. Online available at http://www.cs.waikato.ac.nz/~ml/weka/
[38]. Blake, C. and Merz, C. J. "UCI repository of machine learning datasets." University of California, Irvine, Dept. of Information and Computer Sciences.(http://www.cs.waikato.ac.nz/~ml/weka/)

AUTHORS PROFILE

Sanjib Saha earned his Bachelor of Engineering and Master of Technology from Burdwan University and Jadavpur University respectively. He is pursuing PhD in Computer Science and Engineering at National Institute of Technology, Durgapur, India. He is working as an Assistant Professor in Department of Computer Science and Engineering at Dr. B. C. Roy Engineering College, Durgapur. He has published research papers in SCI and Scopus journals including conferences. His main research work focuses on Machine Learning, Deep Learning, and Medical Image Classification, Segmentation & Registration.

