2022 IEEE World Conference on Applied Intelligence and Computing (AIC) | 978-1-6654-7988-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/AIC55036.2022.9848879

Analysis of a Cascade Scaling Algorithm using Data Mining Methods

Satyajit S. Uparkar
Inter Institutional Computer Centre
RTM Nagpur University
Nagpur, India
uparkarss2204@gmail.com

Ujwal A. Lanjewar
Prerna College of Commerce
RTM Nagpur University
Nagpur, India
ualanjewar@gmail.com

***Abstract*—As a result of the huge amounts of data available in any subject of research, data mining technologies confront considerable obstacles. Due to the large volume of data, most existing data mining methods are inapplicable to many real-world problems. Data mining methods become ineffective when the problem size becomes too large. Scalability is an issue that must be addressed in data mining algorithms in order to construct high-performance, efficient, and scalable data mining algorithms. A new scalability technique is proposed and applied to several data mining problems in this research. The proposed scaling methodology is developed using a cascade approach. The approach begins with the collection of a large data set from several sources, which is then preprocessed. Once the dataset has been preprocessed, decompose it into smaller data sets of equal size subsets. Then apply a data mining approach to each subset, with the identical data mining method stated for each subset. The findings of the data mining approach on all subsets are pooled and aggregated for the final output. The performance of the suggested algorithm is assessed using a number of criteria, including accuracy, precision, recall, F-score, and execution time. Social advertising and bank marketing are the two datasets on which the proposed method is tested. The suggested approach's performance is compared with non-scale data mining methods, and it is found that the scaling method outperformed non-scale data mining methods on all measures.**

Keywords*—*Cascade, Data Mining, Scalability, Accuracy, Data Parallelism, Pooling

I. INTRODUCTION

When the average dataset size was hundreds or thousands of samples, numerous frequently used data mining algorithms were designed. Working with datasets containing hundreds of millions of cases and thousands of features is becoming commonplace. Data mining algorithms become less successful when presented with these challenges unless they are scaled up adequately. Humans and machine learning algorithms both struggle with very huge datasets [1]. Given a large amount of data, scalability refers to the ability to accurately develop a classifier or prediction. Scalability, as a broad phrase, refers to a method that may execute successfully on a little quantity of data while still being compatible with large amounts of data. In general, scalability is an issue in data mining algorithms that must be addressed in order to produce high-performance, efficient, and scalable data mining algorithms. After dealing with the scalability issue, data mining algorithms can readily extract information from large datasets [2].

Because real-world data is enormous, scalability is essential. The goal of this paper is to overcome the scalability problem by developing data mining methods and testing them on four scalable datasets. The first step is to train an algorithm on a dataset and produce good results, but then it must be computationally possible to execute the algorithm on a system that will analyze terabytes of data. Most deep learning-based data mining methods necessitate a large amount of computer power, but they must be clever in how they use approximations in their algorithms in order to perform at higher levels [3].

In this work, cascade approach is applied on data mining methods for scalability. Time complexity, Training/Testing accuracy, and a set of additional performance characteristics including recall, precision, and F1 Score were employed as performance measures in this work. Scalable and non-scalable methodologies are used in the comparative analysis. Then we can conclude that scalable data mining algorithms are capable of processing large amounts of data in a short amount of time.

II. LITERATURE REVIEW

Lot of research papers are published in the field of scalability in data mining. Breiman advocated pasting votes to create a large number of classifiers from short training sets or data "bits." He presented two vote-pasting strategies: Ivote and Rvote. In Ivote, each consecutive classifier's short training set (bite) is based on the collective hypothesis of the preceding classifiers, and sampling is done with substitution. Ivote is alike of boosting in that the "bite" sizes are significantly smaller than the creative data set. Ivote uses significance sampling to build training sets (and consequently classifiers) in a sequential manner. Rvote is a quick and easy way to make a lot of random bites. Rvote was not comparable with Ivote or Adaboost in terms of accuracy, according to Breiman. Furthermore, sampling from the group of training data may necessitate many haphazard disc requests, putting a strain on the CPU. As a result, Breiman presented a different approach: a sequential traversal of the data set. An instance is read and checked in this scheme to see if it will be included in the training set for the next classifier in the aggregate. The sequential pass through the dataset technique, on the other hand, resulted in a loss of accuracy for the vast majority of

Authorized licensed use limited to: AMITY University. Downloaded on September 06,2022 at 13:13:14 UTC from IEEE Xplore. Restrictions apply.

datasets. Breiman further highlighted that for extremely skewed datasets, this strategy of sequentially reading instances from disc will not work [4].

Chawla et al. dispersed Breiman's techniques to deal with these issues by separating the original data set into T distinct portions and assigned each disjoint subset to a separate processor. They pasted little votes on each of the randomly picked disjoint partitions, following Breiman's method. Chawla et al. used a majority vote to integrate the predictions of all the classifiers. Using the aforementioned memory requirement methodology, dividing the data set into T distinct subsets reduces the memory required by a factor of 1/T, which is significant. As a result, DIvote has the potential to be more memory scalable than Ivote. A data set can be divided into subsets that can be managed simply by the computer's main memory. The technique for pasting DRvotes is similar to that of DIvotes. Each bite is a bootstrap duplicate of size N, which is the only difference. Throughout all iterations, each instance has the same chance of being chosen. DRvote, on the other hand, has lower accuracies than DIvote. Breiman's views on Rvote and Ivote are supported by this finding [5]. Fan et al. created boosting for distributed learning and scalable, in which all classifiers was trained with only a miniature segment of the training set. [6]

Bondi et al. attempted to describe various types of scalability, including structural and load scalability. Structural scalability refers to a system's ability to expand in a given dimension without requiring major architectural changes. Load scalability refers to a system's capacity to scale gracefully as the amount of traffic it can handle grows [7]. It is suggested that systems with low load scalability do so because they constantly engage in unproductive activity, are hampered by bad scheduling algorithms, are unable to properly use parallelism, or are algorithmically inefficient. They can distinguish between those characteristics that limit development due to space and/or structural considerations alone and those that affect performance by distinguishing between structural and load scalability [1].

Grandvalet and Canu proposed a technique for determining the importance of input variables in kernelized Support Vector Machines automatically. Scale factors define the input space metric for determining relevance, and feature selection is done by assigning zero weights to irrelevant variables [8]. The metric is automatically calibrated by minimizing the standard SVM empirical risk, which adds scale factors to the standard set of classifier parameters. Constraints that encourage the scarcity of scale factors are used to choose features [9]. The resulting approach outperforms state-of-the-art feature selection algorithms and is demonstrated to be effective on a difficult face expression recognition challenge [10]

Liu et al. offered thorough features for a collection of exemplary data mining applications from both the hardware and software viewpoints. They started with MineBench, a benchmarking package that includes two association rule mining, two classification, and four clustering applications from a variety of areas. On an 8-way Shared Memory Parallel (SMP) system, they test the MineBench apps and examine their key performance features. The input datasets and number of processors used during the evaluation are adjusted to test the scalability of the applications in our benchmark suite. The findings were based on scalability, I/O complexity, the percentage of time spent in the OS mode, and communication/synchronization overheads. This data can help future system designers as well as programmers of new data mining algorithms improve system and algorithmic performance [11] [12].

III. Proposed Methodology

In order to build high-performance, efficient, and scalable data mining algorithms, scalability is an issue that must be addressed in data mining algorithms. Data mining methods can easily extract information from enormous datasets after dealing with the scalability issue. A new scalability algorithm is proposed and used to several data mining algorithms in the proposed work. The cascade approach is used to develop the proposed scalability methodology. Figure 1 depicts the intended work procedure. The procedure begins with the collecting of a huge dataset from several sources, followed by pre-processing of the dataset. Decompose the dataset into smaller datasets of equal size subsets once it has been pre-processed. Then, on each subset, use a data mining methodology, with the same data mining method specified for each subset. Data parallelism and multiple data mining system are the two scalability ideas used in this research. Data parallelism is another term for data decomposition and multiple data mining system is the use of data mining method on every subset in parallel manner. For the final output, the results of the data mining approach on all subsets are pooled and aggregated. Overview of proposed cascade method is shown in figure 2. The suggested algorithm's performance is measured using a variety of metrics, including accuracy, precision, recall, f-score, and execution time.

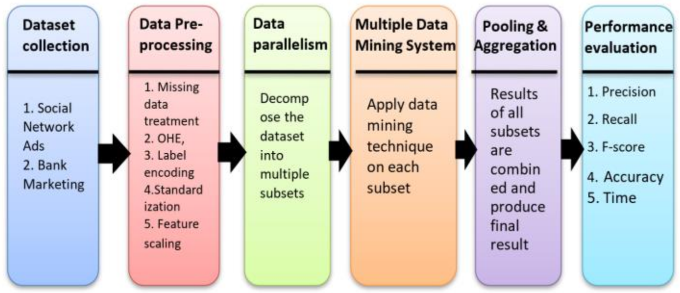

Fig. 1. Intended workflow of proposed cascade scalable approach

Data partitioning methods are strongly related to data parallelization. A data mining algorithm can be scaled up by dividing the dataset into small subsets. The next stage would be to parallelize the data mining algorithm's application to small subsets. Decompose a large data collection into smaller chunks and apply data mining classifiers to each one. If each smaller problem involves s samples, the complexity of solving all of the problems is on the order of $(n/s)\ s^2 = ns$, which is much less than n^2 if n is significantly greater than s. Following the decomposition of the entire dataset into subsets, each

Authorized licensed use limited to: AMITY University. Downloaded on September 06,2022 at 13:13:14 UTC from IEEE Xplore. Restrictions apply.

subset's data is divided into train and test portions. In the proposed system, 80% of the data is used for training and 20% is used for testing. The same division is applied to each subset, after which a data mining classifier is applied to the training set and the model is predicted. This predicted model is used on test data to determine the model's accuracy on each subset of the test dataset. Precision, recall, f-score, and execution time are also used to identify each subset.

After getting the results from each subset, combine them. These combined results are summed up to produce a final output for each metric. The cascade scaling method separates the dataset S into numerous little disjoint datasets Si,j for each step i. Each of these subsets is subjected to a data mining technique with no alterations, and a result Ci,j is generated from each subset of the data. The combination technique creates the final result C as the data mining process's output after all have been applied and are independent of one another. The algorithm for proposed approach is also mentioned below.

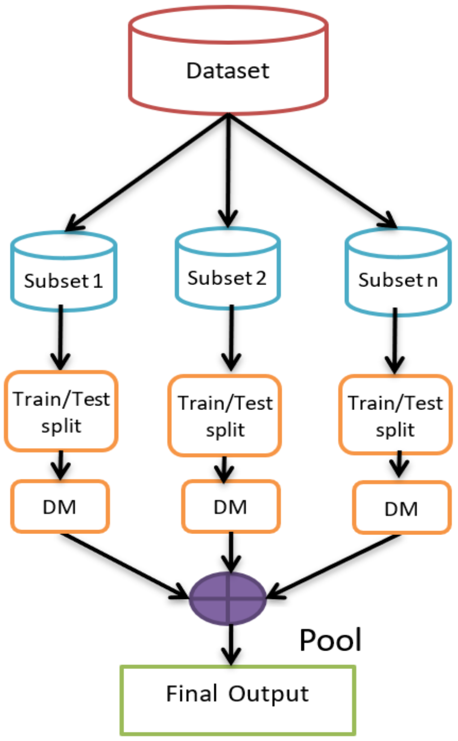

Fig. 2. Overview of proposed cascade method

The following Table is the innovative scalability approach for the proposed cascade method applied for the Data Mining Algorithms.

TABLE I. INNOVATIVE CASCADE SCALABILITY APPROACH

Algorithm: Proposed Cascade scalability approach for Data Mining Algorithms

Inputs: *Raw dataset*

```
Output: Best algorithm, Accuracy, Precision, Recall, F-Score, Execution Time
1.Dataset collection
  Df →load dataset()
2. Identify count, mean, standard deviation, min, max, 25th percentile, 50th percentile and
  75th percentile
  Df.describe()
3. Pre-process dataset
4. if (missing data exists):
      treat_missingData()
  Else:
      cleaned_data()
5. Decompose dataset into n number of equal sized subsets
  Split_dataset(n)
6. identify the max value of n to reduce overfitting
  n = math.sqrt(len(dataset))
7. check for overfitting
  If (subset <= n):
     No overfitting
  Else:
     overfitting exists
     Reduce the value of n
8. split each subset into train and test data with randomization
9. Apply same data mining method on each subset
10. identify accuracy, precision, recall, f-score, execution time of each subset
11. perform pooling and aggregation on results of subsets
12. for N in split (dataset, n):
     x_train, x_test, y_train, y_test = train_test_split (x,y,test_size = 20%, random_state=1)
     model = DecisionTreeClassifier() / RandomForestClassifier() / AdaBoostClassifier() /

SupportVectorClassifier(kernel = 'rbf', random_state = 2) / LogisticRegression() /

KNeighborsClassifier(n_neighbors=5, metric='minkowski') / GaussianNB()
     accuracy = accuracy_score (y_test, model.predict(x_test))
     precision = precision_score (y_test, model.predict(x_test))
     recall = recall_score (y_test, model.predict(x_test))
     execution_time = endtime – starttime
end for
13. final_acc = (sum(accuracies) / len(accuracies))
   final_pre = (sum(p_score) / len(p_score))
   final_rec = (sum(r_score) / len(r_score))
   final_f1 = (sum(f1_scor) / len(f1_scor))
   final_time = (execution_time)
```

Authorized licensed use limited to: AMITY University. Downloaded on September 06,2022 at 13:13:14 UTC from IEEE Xplore. Restrictions apply.

14. Perform comparison between results of different data mining algorithms
15. Perform comparison between proposed scalable and non-scalable method
16. Perform comparison between proposed scalable with existing scalable methods.

IV. Experimental Setup And Evaluation

The following hardware and software configurations were used to conduct the research. 1. Processor: AMD E1-2500, 1.4 GHz; 2. RAM: 4 GB; 3. System: 64-bit OS / Ubuntu Linux OS; Hard disc: 500 GB are the hardware specifications. The software used is: 1. Microsoft Excel in CSV; 2. Anaconda for python jupyter notebook; and 3. Python.

A. Datasets

In this research work, two datasets are used for experiment purpose, which is mentioned below. 80% data is used for training the data mining model and 20% is used for test the model.

1) Social Networks Ads Dataset: The dataset contains information on users who bought or didn't buy a product (1/0). It has five qualities, with the purchased column acting as a dependent variable and the others acting as independent variables. [13]

2) Bank Marketing Dataset: This dataset contains the information about the account holders in bank. [14].

In proposed research work, initially collect the datasets from various online platforms and quantity of dataset are described in Table II.

TABLE II. Dataset Description

Dataset Name	Quantity	Training Data	Testing Data
Social Networks Ads	401	321	80
Bank Marketing	45212	36170	9042

B. Performance Evaluation Parameters

Following performance metrices are used as the evaluation parameters [15][16].

1) Accuracy: It shows the model's overall accuracy, which is the percentage of total samples correctly identified by the classifier.

2) Precision: It shows what percentage of predictions that were labelled as positive classes were in fact positive.

3) Recall: It shows the percentage of all positive samples the classifier accurately predicted as positive.

4) F-score: F-scaore or F1-score is a measure that unites recall and precision. It's the harmonic mean of precision and recall in mathematics. Thus any significance change in any one of these two can reflect the change in the Fscore

5) Execution Time: Time is already frequently utilized to assess performance in a variety of contexts. In general, time-based measurement refers to the measurement of time during classification in both scalable and non-scalable systems. Basically it is calculating by difference of start process time of method with end process time of method. The execution time is measured in terms of seconds.

C. Data Mining Appoarches

In this work, Decision tree, Random-forest, AdaBoost, SVM and Logistic Regression is used for experiments on both the datasets [17] . The two data sets under consideration supports the binary classification of their target variables. The scalability issue is integrated by the machine learning approach by using the 80:20 ratio of training and testing of the two data sets [18]. Thus, the overfitting 90:10 and under estimates of 60:40 and 70:30 ratios were overcome by using the 80:20 ratio.

V. Results And Discussion

The proposed scaling method is compared with non-scaling for different data mining methods.. Table III presents the performance evaluation of Social Network Ads dataset with different data mining methods using proposed cascade approach. In this table non scaling and with scaling results are shown for accuracy, precision, recall, f-score and execution time.

TABLE III. Performance Evaluation Of Social Network Ads Dataset With Different Data Mining Methods

Non-Scaling					
Data Mining Methods	***Accuracy***	***Precision***	***Recall***	***F-Score***	***Execution Time***
Decision Tree	78.75	78.75	77.60	78.69	0.001263
Random Forest	85	85	85.41	85.11	0.138214
AdaBoost	88.75	88.75	89.58	88.84	0.0756383
SVM	70	70	65.62	68.20	0.0057478
Logistic Regression	60	60	50	45	0.0066910
Scaling using Cascade method					
Data Mining Methods	***Accuracy***	***Precision***	***Recall***	***F-Score***	***Execution Time***
Decision Tree	97.5	97.5	98.33	97.57	0.00107
Random Forest	96.25	98.75	99.16	98.78	0.15192
AdaBoost	97.5	99.16	99.44	99.18	0.07332
SVM	98.75	99.37	99.58	99.39	0.00167
Logistic Regression	97.5	99.6	99.73	99.61	0.00128

It is observed that accuracy, precision and recall is greatly improved using scaling method as compared to non-scaling. Accuracy achieved by SVM is 98.7% which is best using scaling method and precision, recall achieved best by KNN and logistic regression which is around 99%.

Authorized licensed use limited to: AMITY University. Downloaded on September 06,2022 at 13:13:14 UTC from IEEE Xplore. Restrictions apply.

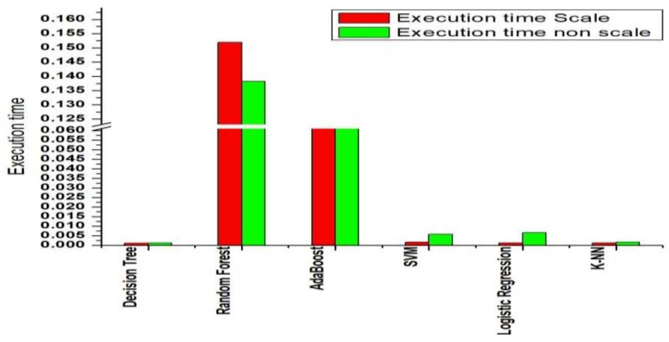

Fig. 3. Comparison of Execution time of proposed scaling method with non- scaling Social Ads dataset

From the above figure the processing time taken by scaling method is too less than non-scaling which means in all measures scaling method performed outstanding on non-scaling methods of data mining. Random Forest and AdaBoost took much more time as compared to the others.

Table IV presents the performance evaluation of Bank Marketing with different data mining methods using proposed cascade approach.

TABLE IV. PERFORMANCE EVALUATION OF BANK MARKETING DATASET WITH DIFFERENT DATA MINING METHODS

Non-Scaling					
Data Mining Methods	***Accuracy***	***Precision***	***Recall***	***F-Score***	***Execution Time***
Decision Tree	93.34	93.28	47.93	93.91	0.02007
Random Forest	97.04	97.04	49.87	95.84	0.49935
AdaBoost	97.3	97.29	50	95.96	0.21585
SVM	97.3	97.29	50	95.96	0.15122
Logistic Regression	97.3	97.29	50	95.96	0.1297
Scaling using Cascade method					
Data Mining Methods	***Accuracy***	***Precision***	***Recall***	***F-Score***	***Execution Time***
Decision Tree	97.99	97.99	94.75	97.53	0.00173
Random Forest	95.61	98.99	97.37	98.76	0.1397
AdaBoost	97.89	99.33	98.25	99.17	0.00205
SVM	88.87	99.49	98.68	99.38	0.00093
Logistic Regression	95.95	95.95	98.95	99.5	0.0017

It is observed that precision and recall is greatly improved using scaling method as compared to non-scaling. Processing time taken by scaling method is too less than non-scaling which means in all measures scaling method performed outstanding on non-scaling data mining methods. The comparative analysis of the execution time is reflected in the following figure.

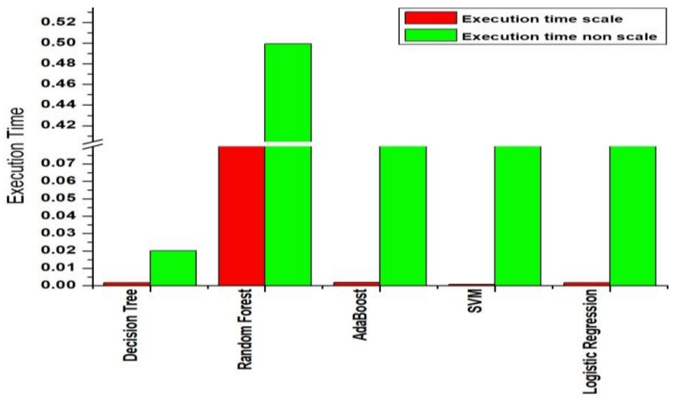

Fig. 4. Comparison of execution time of proposed scaling method with non -scaling Bank Marketing dataset

Random Forest in this case also took much more time as compared to the others. Table V presents the performance evaluation of Social Networking Add with different scaling methods using proposed cascade approach. It is observed that except normalization method other methods such as min max, standard, max absolute and robust scaling methods performed better on all data mining techniques. Table VI presents the performance evaluation of Bank Marketing with different scaling methods using proposed cascade approach. It is observed that proposed method performed better against existing scaling methods on all data mining methods except SVM. On SVM proposed method gave 88.87% accuracy which is lesser than existing scaling methods, so it has scope for improvement.

Table V. Performance Evaluation of Social Add Dataset With existing scaling and proposed Method

Scaling Methods	Decision Tree	Random Forest	AdaBoost	SVM	Logistic Regression	K-NN
MinMax	0.8125	0.85	0.8875	0.8625	0.825	0.8625
StandardScaling	0.8	0.85	0.8875	0.8625	0.8375	0.8625
MaxAbsScaling	0.8	0.8375	0.8875	0.8625	0.8125	0.85
RobustScaling	0.8	0.8375	0.8875	0.8625	0.85	0.85
Normalization	0.525	0.575	0.5875	0.6	0.6	0.575
Proposed Cascade Method	97.5	96.25	97.5	98.75	97.5	96.25

One advantage of the cascade method is that it prevents MMS "memory management systems" from thrashing when algorithms attempt to fill large datasets into central memory. Furthermore, if the time complexity of a learning algorithm is

Authorized licensed use limited to: AMITY University. Downloaded on September 06,2022 at 13:13:14 UTC from IEEE Xplore. Restrictions apply.

less than linear in the quantity of instances, processing minute, permanent extent of data subsets consecutively can make it linear, with the invariable phrase reliant on the subset size. Subsets of instances or subsets of features can be used in decomposition.

Table VI. Performance Evaluation Of Bank Marketing Dataset With existing scaling and proposed Method

Scaling Methods	Decision Tree	Random Forest	AdaBoost	SVM	Logistic Regression
MinMax	0.934673	0.970477	0.97299	0.97299	0.97299
StandardScaling	0.934673	0.970477	0.97299	0.97299	0.97299
MaxAbsScaling	0.936558	0.970477	0.97299	0.97299	0.97299
RobustScaling	0.935302	0.970477	0.97299	0.97299	0.97299
Normalization	0.935302	0.972362	0.972362	0.97299	0.97299
Proposed Cascade Method	0.9799	0.9561	0.9789	0.8887	0.9595

VI. Conclusion

A cascade technique is used to build the proposed scaling mechanism. The method starts with gathering the datasets from the available resources, which is then pre-processed. Decompose the dataset into smaller datasets of equal size subsets once it has been pre-processed. Then, for each subset, apply a data mining strategy, utilising the same data mining method mentioned for each subset. For the final result, the findings of the data mining approach on all subsets are pooled and aggregated. A number of metrics are used to evaluate the suggested algorithm's performance, including accuracy, precision, recall, f-score, and execution time. When compared to non-scaling, the accuracy, precision, and recall of the scaling method are significantly enhanced. SVM has the best accuracy (98.7%) when employing the scaling approach, whereas KNN and logistic regression have the best recall and precision (about 99%). In addition, the most significant observation is the comparison of non-scaling and the proposed cascade in bank marketing data for logistic regression. Scaling approaches take significantly less time to process data than non-scaling methods, implying that scaling methods outperformed non-scaling data mining methods on all measures. In future, proposed method can apply on more datasets also including the Non-stationary data sets.

References

[1] A. B. Bondi, "Characteristics of scalability and their impact on performance," in *Proceedings of the 2nd international workshop on Software and performance*, 2000, pp. 195-203.

[2] J. Shafer, R. Agrawal, and M. Mehta, "SPRINT: A scalable parallel classifier for data mining," in *Vldb*, 1996, pp. 544-555.

[3] M. Mehta, R. Agrawal, and J. Rissanen, "SLIQ: A fast scalable classifier for data mining," in *International conference on extending database technology*, 1996, pp. 18-32.

[4] L. Breiman, "Pasting small votes for classification in large databases and on-line," *Machine learning,* vol. 36, pp. 85-103, 1999.

[5] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "Learning ensembles from bites: A scalable and accurate approach," *The Journal of Machine Learning Research,* vol. 5, pp. 421-451, 2004.

[6] W. Fan, S. J. Stolfo, and J. Zhang, "The application of AdaBoost for distributed, scalable and on-line learning," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 362-366.

[7] D. Brain and G. I. Webb, "The need for low bias algorithms in classification learning from large data sets," in *European Conference on Principles of Data Mining and Knowledge Discovery*, 2002, pp. 62-73.

[8] A. Lazarevic and Z. Obradovic, "Boosting algorithms for parallel and distributed learning," *Distributed and parallel databases,* vol. 11, pp. 203-229, 2002.

[9] A. Mc Manus and M.-T. Kechadi, "Scalability issue in mining large data sets," *WIT Transactions on Information and Communication Technologies,* vol. 33, 2004.

[10] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs," in *NIPS*, 2002, p. 2002.

[11] Y. Liu, J. Pisharath, W.-k. Liao, G. Memik, A. Choudhary, and P. Dubey, "Performance evaluation and characterization of scalable data mining algorithms," in *16th IASTED international conference on parallel and distributed computing and systems (PDCS). MIT, Cambridge*, 2004, pp. 620-625.

[12] N. O. Andrews and E. A. Fox, "Clustering for data reduction: a divide and conquer approach," Department of Computer Science, Virginia Polytechnic Institute & State …2007.

[13] https://www.kaggle.com/datasets/rakeshrau/social-network-ads

[14] https://www.kaggle.com/c/bank-marketing-uci

[15] S. Iqbal, R. Naseem, S. Jan, S. Alshmrany, M. Yasar and A. Ali, "Determining Bug Prioritization Using Feature Reduction and Clustering With Classification," in *IEEE Access*, vol. 8, pp. 215661-215678, 2020, doi: 10.1109/ACCESS.2020.3035063.

[16] S. Memiş, S. Enginoğlu and U. Erkan, "Numerical Data Classification via Distance-Based Similarity Measures of Fuzzy Parameterized Fuzzy Soft Matrices," in *IEEE Access*, vol. 9, pp. 88583-88601, 2021, doi: 10.1109/ACCESS.2021.3089849.

[17] V. Bahel, S. Pillai and M. Malhotra, "A Comparative Study on Various Binary Classification Algorithms and their Improved Variant for Optimal Performance," *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 495-498, doi: 10.1109/TENSYMP50017.2020.9230877.

[18] N. K. A. Appiah-Badu, Y. M. Missah, L. K. Amekudzi, N. Ussiph, T. Frimpong and E. Ahene, "Rainfall Prediction Using Machine Learning Algorithms for the Various Ecological Zones of Ghana," in *IEEE Access*, vol. 10, pp. 5069-5082, 2022, doi: 10.1109/ACCESS.2021.3139312.

Authorized licensed use limited to: AMITY University. Downloaded on September 06,2022 at 13:13:14 UTC from IEEE Xplore. Restrictions apply.