Assessment and Comparison of Ensemble Learning Techniques using Feature Reduction for Classification of IoT data

Vijay M. Khadse[1], Dr. Parikshit N. Mahalle[2] & Dr. Gitanjali R. Shinde[3]
[1]*Assistant Professor, College Of Engineering Pune (COEP), Pune, India*
[2]*Senior Member IEEE, Professor & Head, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune India. Post Doc Researcher, Center for Communication, Media and Information Technologies (CMI), Aalborg University, Copenhagen, Denmark*
[3]*Assistant Professor, Smt. Kashibai Navale College of Engineering, Pune India*
[1]*vmk.comp@coep.ac.in,*[2]aalborg.pnm@gmail.com, [3]*gr83gita@gmail.com*

Abstract: *In recent years, the growth of Internet of Things (IoT) as a prominent technology has been incredible. The number of network and sensor enabled devices in IoT domains is growing extremely, leading to the huge production of data. These data contain important information which can be used in various areas, such as science, industry, medical and even social life. To make IoT system smart, the only solution is entering the world of machine learning. Many machine learning algorithms are introduced and for handling such vast amount of IoT data. It is very difficult to find the best suited algorithm for problem in IoT domain. This study explores bagging, boosting and stacking ensemble learning models. This study combined three ensemble models and proposed a hybrid model. A set of features extracted from the raw IoT datasets using Principal component analysis (PCA), Linear discriminant analysis (LDA) and Isomap are used for classification. A performance comparison of the classifiers is provided in terms of their accuracy, area under the curve (AUC) and F1 score. The experimental results of this comparative study show that Hybrid with PCA and Stacking with PCA have better overall performance than other ensemble models for binary and multivariate datasets respectively.*

Keywords*- Ensembles; Bagging; Boosting; Stacking; Random Forest; Classification; UCI and Kaggle datasets.*

1. Introduction

Machine learning (ML) algorithms deal with the problem of learning prediction models of datasets. The datasets contain a collection of instances. Every instance can be described regarding values of variables, which are referred to as features and attributes. The main task of ML is to learn a model using already learn data and predicting the values for unknown instances [1].

Ensemble Learning is the state-of-the-art for different ML problem. This method enhances the predictive performance of a single model by training more models and merging their predictions. In ensemble learning there are a group of base learners (on average 5 to 6) that means a group of models for processing. The main aim of ensemble learning is to combine these models, make the one strong learner. Therefore, the obtained result will be much better than the single base learner [2].

This paper is about the comparison of ensemble learning techniques on IoT sensor datasets. This study is valuable for both research and industry, as the data sets used is from Multi-domain IoT system. Section 2 contains the basics about feature reduction techniques and algorithms used and literature survey conducted of the study. Sections 3 contains the gap analysis. Section 4 describes our proposed methodology. Section 5 shows the performance of models and result. Finally study ends with section 6 having observations and conclusion.

2. Preliminary

2.1 Feature reduction techniques

The following contain the brief summary, merits and demerits of feature reduction techniques used in this study.

2.1.1 Principal Component Analysis (PCA): Principal component analysis (PCA) allows us to summarize and to visualize the information in a data set containing observations described by multiple inter-correlated quantitative variables. Each variable could be considered as a different dimension. It is used to extract the important information from a multivariate data table and to express this information as a set of few new variables called principal components [3].

2.1.2 Linear Discriminants Analysis (LDA): Linear Discriminant Analysis (LDA) is a generalization of Fisher's linear discriminant, a method used in Statistics, pattern recognition and ML to identify a linear combination of features that characterizes or separates two or more classes of objects or events. This method projects a dataset onto a lower-dimensional space with good class separability [4].

2.1.3 Isometric mapping (Isomap): Isomap is a non-linear feature reduction technique. Instead of Euclidean distance, this technique uses geodesic distance to and out distance between two data point and preserves this information in low dimension [5].

2.1.4 T- Distributed Stochastic Neighbour Embedding (T-SNE): T-SNE is a non-linear technique primarily used for data exploration and visualizing high-dimensional data. In simpler terms, t-SNE gives you a feel or intuition of how the data is arranged in a high-dimensional space [6].

2.2 Algorithms

The following describes in brief algorithms used in this study.

2.2.1 Decision tree classifier (DT): This classifier constructs the tree structure model and builds the classification model. Each of the root and internal node describe a test on an attribute and each terminal node specifies the possible values of that attribute [7].

2.2.2 AdaBoost: AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm. It can be used in conjunction with many other types of learning algorithms to improve performance [8].

2.2.3 Random Forest (RF): RF is based on a simple idea: "the wisdom of the crowd". Aggregate of the results of multiple predictors gives a better prediction than the best individual predictor. A group of predictors is called an ensemble [9].

2.2.4 Bernoulli's Naïve Bayes (BNB): It assume that features has only two values either 0 or 1. Analyzing this assumption, this algorithm classifies the data [10].

2.2.5 Gaussian Naïve Bayes (GNB): This classifier constructs the tree structure model and builds the classification model. Each of the root and internal node describe a test on an attribute and each terminal node specifies the possible values of that attribute [11].

2.2.6 Multilayer Perceptron Classifier (MLP): MLP belongs to artificial neural network family. In consist of three layers: input layer, hidden layer and output layer. The neurons use an activation function to produce an output based on weighted input signals [12].

2.2.7 K-Nearest neighbour (KNN): This is supervised algorithm used for classification as well as regression. It uses 'feature similarity' means assigning value of new data point based on closer training set [13].

2.2.8 Support Vector Machine (SVM): SVM is a learning algorithm used for classification and regression. This algorithm calculates maximum marginal hyperplane based on margin of data points and classify the labels in multidimensional space [14].

2.2.9 Gradient Boosting (GB): It is also used for regression and classification problem. It is a boosting ensemble method. Working of gradient boosting and adaboost is similar but instead of using high weight data points to identify the shortcoming, it uses gradients in the loss function [15].

2.2.10 Logistic Regression (LR): Logistic regression is the supervised machine learning algorithm. It predicts the probability of output variable in form of binary regression [16].

2.3 Literature Survey

The number of component classifiers of an ensemble technique and extracted "n" components from original features using feature reduction technique has a great impact on prediction and classification performance.

Authors of [17] has compared feature selection and dimensionality reduction techniques on gesture recognition sensor data to increase the performance. They have selected feature using wrapper forward technique and reduces the features using eight dimensionality reduction techniques. Linear Discriminant Analysis, Manifold Charting, Autoencoder, t-distributed Stochastic Neighbor Embedding, Principal Component Analysis, Large Margin Nearest Neighbor (LMNN) and Isomap are the eight techniques used in this study. They have also used seven different classifiers. They observed that 87% to 90% accu-racy achieved ELM and SVM RBF classifiers in feature selection and 95% accuracy achieved the combination of LMNN and SVM in the dimensionality reduction process. This study showed dimensionality reduction improves the performance of hand gesture dataset.

Authors of [18] compared performance of Bagging (random forest), Boosting (extreme gradient boosting machine XGB) and Stacking ensemble techniques on agriculture business time series dataset. In the experimental study, 2 case studies are examined. Soybean price and wheat price are the case study 1 and 2 respectively. Least absolute shrinkage and selection operator (LASSO), Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and XGB were used for level-0 layer in Stack ensemble technique. As per Authors, XGB boosting model performed well in both the cases, with high accuracy and small prediction rate.

Authors of [19] combined MIWrapper and SimpleMI algorithm and proposed new ensemble based multiple instance multiple instance learning algorithm. They compared multiple instance learning ensemble algorithm with MIWrapper and SimpleMI individual base learners (Naïve Bayes, SVM, C4.5, Multilayer perceptron and Decision tree). Experimental results stated that the ensemble-based model provides high performance than traditional solutions.

Authors of [20] proposed a stacking strong ensemble methodology using Logistic Model Trees and three well know ensembles namely extra tree, random forest and gradient boosting. They concluded that Stacking methodology gave remarkable performance than individual classifiers performance.

Authors of [21] used Cuttlefish optimization algorithm for data point reduction. This optimally extracted subset of data points and a reduced set of features provided by PCA are providing almost the same accuracy, the false positive rate that they had obtained from the original dataset.

Authors of [22] compared seven base classifier and have ensemble methods applied to financial domain dataset. LogR, MLP, C4.5, CDT, CART, SVM and Pegasos are the seven base classifiers and AdaBoost, Bagging, Random Subspace, Decorate and Rotation Forest are the ensemble methods. It has used four evaluation metrics, Area Under the Curve, Accuracy, False positive rate and Time taken to build the model, for performance evaluation for each classifier. It is observed that Pegasos algorithm performed better than other methods for AdaBoost. C4.5, CART and CDT models performed better than other methods analyzed for bagging, Random Subspace, Decorate and Rotation Forest.

Authors of [23] compared feature extraction and feature selection technique on 6 IoT datasets. They have used Principal Component Analysis (PCA), Generalized Hebbian Algorithm (GHA), Independent component analysis (ICA), Singular Value Decomposition (SVD) and Self Organizing Map (SOM) for feature extraction. For feature selection, filter and wrapper techniques are used. For performance evaluation, compactness, accuracy and computational time are used. They concluded that feature extraction techniques performed well on low dimensional data and feature selection perform better on high dimensional data.

Authors of [24] used bagging, boosting and stacking to predict the defected modules from the modules of software. For this experimentation, the have used 11 base classifiers. The 11 base classifiers are Naïve Bayes, Bayes net, Sequential minimal optimization (SMO), PART, J48, RF, Random tree, IB1, VF1, Decision table and NB tree. They used four software products which are developed by NASA. In terms of defect detection accuracy, boosting method demonstrates better efficiency in comparison with bagging method. Result shows that random forest with stacking performed excellent to improve fault detection rate.

Authors of [25] compared Naïve Bayes, Decision Tree, Multilayer Perceptron and K-Nearest Neighbour classification methods on variety of UCI dataset. Multilayer Perceptron achieves high accuracy. They made two ensemble model namely 3-ensemble and 4-Ensemble. Using majority voting technique on ensemble model, experimental results shows that 3-ensemble model achieve high accuracy with 83.13%. Ensemble learning approach proves astonishing result that regular individual classifiers in various domain.

Authors of [26] compared accuracy of single classifiers (KNN, SVM, C4.5, Random Tree, Random Vector Functional Linking) on multiple UCI variety datasets. They compare the accuracy of Ensemble classifiers (PSEL, AdaBoost, Random-Forest, Multiboost, Bagging, RTboost etc.) on UCI datasets. Results show that Progressive Subspace Ensemble Learning (PSEL) performed better as compared to other ensemble classifiers.

Authors of [27] compared 19 ensemble algorithms including Bagging, Boosting, Random forest, Rotation forest and its variants. They have used Decision tree base learners for meta classifiers. For this study, 19 various binary UCI datasets were considered. Accuracy, Area Under the Curve (AUC), Root Mean Square (RMS) were used as performance matrix. They concluded that family of Rotation forest ensemble technique outperformed than other ensemble techniques.

Authors of [28] applied ensemble classifiers on student performance i.e. educational domain dataset. These classifiers are bagging, boosting, Random Forest, Rotation Forest and AdaBoost. It has been observed that Rotation forest is performing very well and the Random Forest algorithm performing very low. AdaBoost and Bagging performed well than Random Forest and close to Rotation Forest. Accuracy of Rotation Forest is 75.95%.

Authors of [29] compared four popular ensemble methods namely Bagging, Boosting, Stacking and Random forest on 31 UCI datasets and justified that, depends upon the dataset domain the result varies means accuracy varies. Therefore no one was the winner from ensemble learning methods.

Authors of [30] discussed four bagging based ensemble classifiers, namely, the ensemble ANFSI, the ensemble SVM, the ensemble ELM and random forest. Ensemble classifiers evaluated on thirteen UCI binary dataset with different bagging numbers (20, 50 and 80). Out of four ensemble classifiers, the ensemble SVM has been identified to be the most favorable ensemble classifier and random forest tree identified second most favorable ensemble classifier.

Authors of [31] investigated network intrusion detection system by applying three ensemble methods (bagging, boosting and stacking). Results show that only stacking method was able to reduce false positive rate as compared to other ensemble methods. Among the four classifiers (J48, naive Bayes, JRip and iBK-nearest neighbor), J48 performed better than the three other methods by achieving the highest accuracy rates and lowest false positive rate.

Authors of [32] compared Bagging, Boosting and Stacking ensemble techniques with Decision tree, Artificial neural network, Support vector machine and logistic regression as a base learner on credit scoring datasets. In this study, Accuracy, type I and type II error were considered for performance measurement of models. They concluded that Stacking and Bagging with decision tree performed better than all ensemble models in terms of accuracy, type I and type II error.

Authors of [33] examined six distinct ML classifiers to three ensemble techniques i.e. Bagging, Stacking and Additive Regression. According to these researchers, most of the cases, model obtained by Stacking were characterized by the lowest prediction error, but outcome tended to vary giving once better. Bagging approach seemed to be more stable but gave worse result than stacking and additive regression. With Additive regression all multilayer perceptron (MLP) and Radial Basis Function (RBF) ensembles yielded significant error reduction compared to original model.

3. Gap Analysis

Most of the researchers previously compared several base classifiers for the ensemble. It is observed that three main ensemble methods i.e. Bagging, Boosting and Stacking are used for the comparison on multiple datasets. For feature reduction many researchers have worked on PCA to achieve high accuracy rate and low computational time. Based on literature survey, it is observed that researchers while using ensemble methods for comparison on multi-domain datasets do not consider feature reduction techniques. Similarly, while comparing feature reduction techniques on multiple domain datasets they do not use ensemble methods.

Unlike from existing studies, this study not only compares the performance of Bagging, Boosting and Stacking models but also the proposed hybrid ensemble model. This study also uses feature reduction techniques on IoT binary and multiclass multi-domain datasets. This study considers PCA, LDA and Isomap as feature reduction technique to improve the performance of the model on diverse multi-domain IoT datasets.

4. Proposed Methodology

4.1 Design

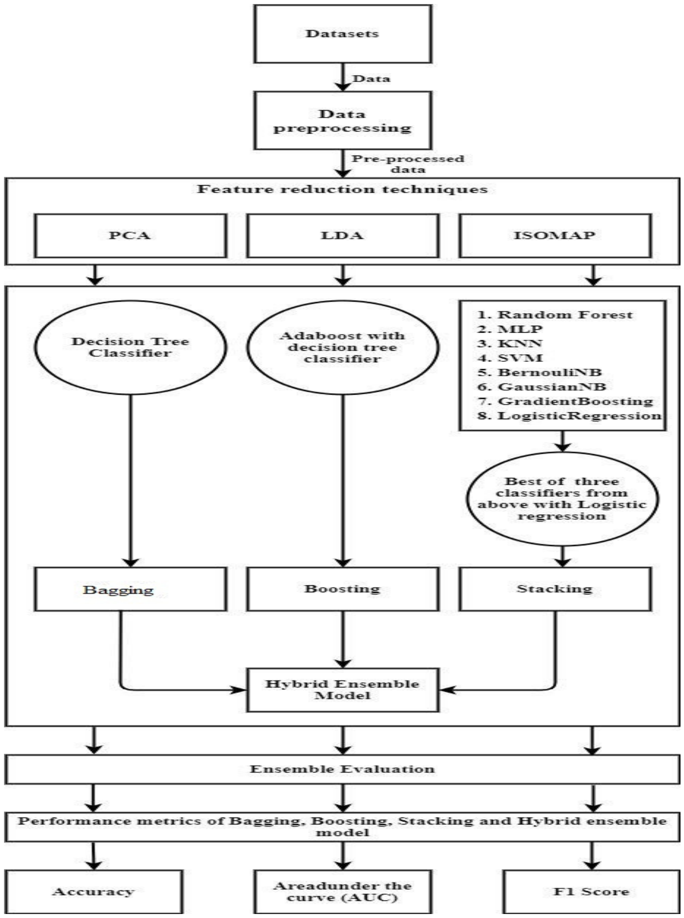

Figure 1. Proposed Methodology

The above figure 1 is proposed methodology for this study. It involves the ensemble learning of individual classifiers and feature reduction techniques which are mentioned above. This methodology divided into five stages. Following are the stages included for performing the comparative study of feature reduction techniques and ensemble learning methods.

4.1.1 Datasets used: This study has used dataset from UCI ML storehouse [34] and Kaggle. From both repositories 10 binary and 10 multivariate IoT sensor datasets of variety of domains were collected. These datasets contain the values provided by the sensors. Table 1 contains the information about features, classes, class types and instances of datasets.

Table 1: Dataset detail

Sr No.	Datasets	Total features	Total classes	Class types	Instances
1	Electric grid	14	2	Binary	10000
2	Extra sensory - B	277	2	Binary	2686
3	Football sensor	9	2	Binary	945
4	Pulsar star	9	2	Binary	9652
5	EEG signal	15	2	Binary	8123
6	Power system - A	129	2	Binary	5161
7	Hand gesture recognition	65	2	Binary	5811
8	Watch sensor	13	2	Binary	7386
9	Power system - B	99	2	Binary	4966

10	Machine sensor	75	2	Binary	10616
11	Cardiotocography sensor	41	10	Multivariate	2126
12	Extra sensory - A	277	6	Multivariate	2686
13	Mode detection	33	5	Multivariate	5894
14	Sky server	18	3	Multivariate	10000
15	Movement recognition	563	6	Multivariate	2948
16	Air quality sensor	16	5	Multivariate	9358
17	Energy prediction	29	8	Multivariate	19736
18	Big sensors	561	6	Multivariate	6373
19	Transport detection	38	5	Multivariate	5894
20	Direction sensor	25	4	Multivariate	5455

4.1.2 Data Pre-processing: Out of range values, missing values, impossible data combination etc lead to undesirable effect on ML prediction model. Gathered data from the resources is not in a standardized form. This data contains a lot of null and missing values. This study removes the null or missing values in data. Large numerical values require normalize data for feature reduction. Column normalization technique is applied on IoT multi-domain datasets. Data values are rescaled between ranges of 0 to 1. Also checked the data for positive or negative infinity and the problem to return the result in Boolean array using numpy logic function.

4.1.3 Feature Reduction: Due to large number of features in the dataset, it gets complex to visualize the data and work on it. Many of the features or dimensions are correlated, therefore it becomes redundant. Using feature reduction techniques, higher dimensions of dataset gets converted into new set of synthetic dimensions and extracted lower dimentions to avoid overfitting problem and improve the performance of the model.

This research study used two linear and two nonlinear feature reduction techniques. Principal component analysis (PCA) is linear technique for reducing the dimensionality as well as minimizing information loss. Using this method, the study creates new uncorrelated data by minimizing variance. In proposed methodology, PCA's "Explained variance ratio" function used to reduce the features from the original set of features. It is mostly used tool in ML and data analysis research. Linear Discriminant Analysis (LDA) is also linear technique for reducing the dimensionality based on the classes of dataset. Using this technique, this study finds the dimensions which can maximize the separability between the classes to make good decision to classify data. This proposed methodology applied LDA's "Explained variance ratio" function to minimize the features. Isomap is nonlinear technique for visualization of data and compute low dimensional embedding of high dimensional data. To reduce the set of features, the study utilizes Isomap's "Reconstruction error" function. The number of neighbours depend on the number of instances in datasets. Isomap is very efficient and used for high number of dimensionalities. For the visualisation of the data, the study applied T - distributed stochastic neighbour embedding(T-SNE) technique on dataset. To observe how the data is separated, study visualises the data in 2 dimensions.

4.1.4 Ensemble Method: This work considers four ensemble learning techniques. They are Bagging, Boosting, Stacking and Hybrid. The details of the four ensemble methods are given below.

Bagging [35] also called Bootstrap aggregation, is to build some subsets(trees) of data which is selected randomly with replacement from training data. [47] in its study has set number of trees to 50 for experimentation. Aggregating all the 50 prediction of tree which is better than considering prediction of single tree and it makes the model more robust. Proposed methodology used decision tree classifier as a base learner for bagging ensemble method.

Boosting is sequential learner technique [36]. Learner fits the model to the data with previous learner and inspects data for errors. If an input is misclassified by previous prediction model then the study model increases the weight of the input for the next learner to correctly classify. Because of that,it reduces bias of decision tree classifier. In this technique, decision tree classifier is used as a base learner and Adaboost classifier as a meta classifier. For experimentation, [47] in its study sets number of trees to 50 for boosting ensemble method.

Stacking is to combining multiple models, make the training data of their predictions and apply meta classifier on the training data [37]. Because of this approach, the performance of model gets improved. For the first layer in the stacking model, this study has chosen eight classifiers from several families of algorithms. Random forest classifier from tree family [38], Multilayer perceptron classifier (MLP) from neural network family [39], Gradient boosting classifier from ensemble family [40], Bernoulli and gaussian classifiers from Naïve Bayes family [41], K-nearest neighbour classifier (KNN) from instance-based family [42], Logistic regression classifier from regression family [43] and Support vector machine (SVM) classifier belong to the generalised linear classifier [44]. Out of eight classifiers, this study separates three high-performance classifiers for binary datasets as well as multivariate datasets for making a stack. There are two methods to build a stack of classifiers in stacking ensemble model i.e. using prediction values of the classifier and using probabilities of prediction values of the classifiers. This research work chose probabilities of prediction values of the classifiers method for both binary and multivariate datasets to build the stacking model. This technique helps to boost the performance of the stacking model. For the second layer in the stacking model, since the data in this study is derived from the original dataset with complex transformation, it is not necessary to select complex classifiers in the output layer. Logistic regression is good choice and it also prevents overfitting. That is why, Logistic Regression was selected as meta classifier.

This study merged predictions of bagging, boosting, stacking ensemble models and proposed new “Hybrid Ensemble Model” (HEM). For binary datasets, this study combined the prediction values (values which are predicted on training dataset) of bagging, boosting, stacking ensemble models and made new predicted values. For Multivariate datasets, this study combined the prediction values as well as the probability of prediction values (probability of predicted values made on the training dataset) of bagging, boosting, stacking ensemble models and made our new predicted values. Finally, the study compares predicted value to test dataset (contained actual value of the inputs) to measure the performance. The HEM aims to improve stability of ensemble learning model. Evenif training data is slightly modified, the prediction will not change.

4.1.5 Performance Metrics: For this comparative study, performance of models is compared using three technique. Accuracy is most common and essential technique to measure the prediction rate of the model. For multiclass classification problem comes into the picture, Area Under the Curve (AUC) is must for use. This technique shows how much your model can classify between labels. Higher the score of AUC, well predicted the classes. F1 score shows the balance between precision and recall. However, it does not consider true negative in measurement.

5. Experimentation and Result

This study selected some high dimensional datasets and some low dimensional datasets as shown in Table 1. The idea behind this approach is to how PCA, LDA and Isomap reduces the dimensions on a dataset.

Below Table 2 contain the information of total features and reduced features of each IoT binary and multivariate datasets.

Table 2: Dimensions reduce by PCA, LDA and Isomap

Sr No.	Datasets	Total Features	Reduced dimension		
			PCA	LDA	Isomap
1	Electric grid	14	10	1	11
2	Extra sensory - B	277	18	1	21
3	Football sensor	9	7	1	8
4	Pulsar star	9	4	1	3
5	EEG signal	15	5	1	9
6	Power system - A	129	22	1	27

7	Hand gesture recognition	65	42	1	23
8	Watch sensor	13	6	1	4
9	Power system - B	99	18	1	22
10	Machine sensor	75	13	1	16
11	Cardiotocography sensor	41	12	1	15
12	Extra sensory - A	277	11	4	23
13	Mode detection	33	5	3	11
14	Sky server	18	8	2	10
15	Movement recognition	563	110	3	195
16	Air quality sensor	16	4	4	5
17	Energy prediction	29	13	1	14
18	Big sensors	561	190	3	209
19	Transport detection	38	7	3	13
20	Direction sensor	25	18	3	18

Scikit learn is a useful library for machine learning [45]. It offers a number of supervised and unsupervised learning algorithms via simple python framework. In the scikit learn, PCA, LDA and Isomap include common one parameter i.e. “n-components”.

This parameter indicates the number of features to be returned for further processing. To figure out the value of parameter, corresponding function of its feature reduction technique is used. Before applying feature reduction technique, data is converted into standardize form using “standard scalar” function.

5.1 Selection of “n” Features

In PCA, it creates its own new “n” features from original “n” features of the dataset. “Explained variance ratio” function is used to extract new features. “Explained variance ratio” function returns a vector explained by each feature. A cumulative sum of variance ratio of each feature is returned in ascending order and the cumulative variance is plotted. This tells how many n-components are required to cover the whole variance. For the research study, it has set the threshold at 95% and choose first “n" features to achieve threshold value.

LDA creates its own new components based on the labels of the dataset. Suppose a dataset has ‘x’ features then LDA creates its own new ‘x-1’ features. Due to this strategy, this study observes the reduced dimensions of LDA is far lesser than the PCA and Isomap. To extract the features from the LDA’s new features set, this study applied same function used in PCA i.e. “Explained variance ratio”. This study built a function which consecutively adds the explained variance of features until the threshold can’t fit any more features. Then it returns the number of features added. The threshold was set to be 95%.

Isomap is far more different than the PCA and LDA. It needs to determine the number of neighbors. Selecting a large number of neighbors makes it computationally expensive. Due to this problem, this studies takes the square root of total instances of dataset and initialize the number of neighbors for the Isomap. To extract the features, “Reconstruction error” function of Isomap is applied. This function signifies distance between original data point and its projection point onto a lower-dimensional space. This study provides error rate of each feature. This increases the number of features, decreases the reconstruction error rate. After a certain number of features, the reconstruction error rate stabilizes. Number of features were chosen where the reconstruction error rate gets stabilized.

5.2 Execution and Result

To evaluate the efficiency of model and reduce the overfitting and underfitting problems,cross-validation technique is used. Datasets utilized have not enough instances to construct optimal model and results are fluctuated for different splits of the data. Due to these problems, K-fold cross validation technique is applied. Both widely used 10-fold and 5-fold cross validation techniques is applied on binary and multivariate datasets. As far as performance is concerned, there is not much difference between 5-fold and 10-fold cross validation techniques. In K-fold cross validation, there is bias-variance trade-off correlated with decision of K [46]. Generally, despite these criteria, one performs K-fold cross validation with K=5 or K=10, since values have been experimentally shown to provide test error rate estimates that do not suffer from extreme bias or extremely high variance. For experimentation, this study has used 5-fold cross validation technique. First, the study used bagging model with decision tree classifier working as a base learner and evaluated on each dataset. Table 3 provides details about Accuracy, AUC and F1 score of bagging with PCA, LDA and Isomap.

Table 3: Accuracy, AUC and F1 score of PCA, LDA and Isomap with bagging model on each dataset.

No.	Dataset type	Dataset	Feature Reduction Technique	Accuracy	AUC	F1 score
1	Binary	Electric grid	PCA	88.885	0.875	0.913
			LDA	97.699	0.975	0.981
			Isomap	89.775	0.887	0.920
2		Extra sensory - B	PCA	96.713	0.967	0.969
			LDA	97.281	0.972	0.974
			Isomap	96.582	0.965	0.967
3		Football sensor	PCA	90.106	0.899	0.882
			LDA	89.707	0.894	0.876
			Isomap	89.734	0.894	0.877
4		Pulsar star	PCA	97.041	0.908	0.875
			LDA	97.064	0.919	0.879
			Isomap	97.111	0.907	0.877
5		EEG signal	PCA	98.155	0.981	0.981
			LDA	55.778	0.557	0.564
			Isomap	92.635	0.926	0.927
6		Power system - A	PCA	91.056	0.866	0.830
			LDA	72.834	0.640	0.477
			Isomap	90.377	0.852	0.813
7		Hand gesture recognition	PCA	95.636	0.956	0.956
			LDA	66.088	0.660	0.670
			Isomap	95.933	0.959	0.959
8		Watch sensor	PCA	99.251	0.992	0.992
			LDA	73.422	0.734	0.720
			Isomap	99.380	0.993	0.993
9		Power system - B	PCA	90.996	0.828	0.770
			LDA	75.961	0.622	0.408
			Isomap	91.273	0.836	0.780
10		Machine sensor	PCA	90.984	0.884	0.859
			LDA	65.303	0.601	0.463
			Isomap	90.400	0.880	0.851
11	Multivariate	Cardiotocography Sensor	PCA	99.800	0.999	0.998
			LDA	100	1.0	1.0
			Isomap	99.729	1.0	0.997
12		Extra sensory – A	PCA	75.940	0.496	0.759
			LDA	79.310	0.608	0.793
			Isomap	74.022	0.516	0.740
13		Mode detection	PCA	87.033	0.941	0.870
			LDA	71.549	0.921	0.715
			Isomap	85.301	0.923	0.853
14		Sky server	PCA	88.482	0.886	0.884
			LDA	94.385	0.947	0.943
			Isomap	84.9557	0.852	0.849
15		Movement recognition	PCA	89.711	0.976	0.897
			LDA	85.118	0.954	0.851

			Isomap	89.575	0.982	0.895
16		Air quality sensor	PCA	90.721	0.999	0.907
			LDA	95.726	1.0	0.957
			Isomap	90.721	0.997	0.893
17		Energy prediction	PCA	97.215	0.997	0.972
			LDA	99.989	0.999	0.999
			Isomap	75.350	0.744	0.753
18		Big Sensors	PCA	85.851	0.991	0.858
			LDA	84.364	1.0	0.843
			Isomap	82.362	0.989	0.823
19		Transport detection	PCA	89.524	0.937	0.895
			LDA	77.113	0.923	0.771
			Isomap	89.061	0.890	0.936
20		Direction sensor	PCA	85.353	0.817	0.853
			LDA	75.215	0.722	0.752
			Isomap	84.064	0.817	0.840

Next, Adaboost boosting model is used with decision tree classifier as a base learner and evaluated each dataset. Table 4 shows Accuracy, AUC and F1 score of boosting ensemble model with PCA, LDA and Isomap.

Table 4: Accuracy, AUC and F1 score of PCA, LDA and Isomap with boosting model on each dataset.

No	Dataset Type	Dataset	Feature Reduction Technique	Accuracy	AUC	F1 score
1	Binary	Electric grid	PCA	84.215	0.828	0.876
			LDA	97.567	0.973	0.980
			Isomap	85.817	0.844	0.889
2		Extra sensory - B	PCA	95.418	0.954	0.957
			LDA	96.145	0.961	0.964
			Isomap	95.027	0.950	0.953
3		Football sensor	PCA	85.159	0.848	0.823
			LDA	86.276	0.857	0.832
			Isomap	85.010	0.857	0.832
4		Pulsar star	PCA	95.352	0.895	0.815
			LDA	95.686	0.902	0.828
			Isomap	95.183	0.892	0.809
5		EEG signal	PCA	95.812	0.958	0.958
			LDA	53.750	0.537	0.544
			Isomap	88.285	0.882	0.885
6		Power system - A	PCA	88.638	0.859	0.800
			LDA	69.060	0.620	0.457
			Isomap	88.299	0.857	0.795
7		Hand gesture recognition	PCA	91.105	0.911	0.911
			LDA	62.874	0.628	0.628
			Isomap	93.265	0.932	0.932
8		Watch sensor	PCA	98.879	0.988	0.988
			LDA	70.294	0.702	0.703
			Isomap	99.102	0.991	0.991
9		Power system - B	PCA	89.214	0.846	0.758
			LDA	74.798	0.634	0.430
			Isomap	89.290	0.845	0.758
10		Machine sensor	PCA	88.064	0.867	0.825
			LDA	63.848	0.596	0.467

			Isomap	86.146	0.869	0.827
11	Multivariate	Cardiotocography sensor	PCA	99.051	0.997	0.990
			LDA	100	1	1
			Isomap	99.360	0.990	0.993
12		Extra sensory - A	PCA	68.864	0.557	0.688
			LDA	73.175	0.640	0.731
			Isomap	65.968	0.541	0.659
13		Mode detection	PCA	82.075	0.912	0.820
			LDA	66.146	0.897	0.661
			Isomap	79.796	0.891	0.797
14		Sky server	PCA	84.090	0.854	0.840
			LDA	92.582	0.927	0.925
			Isomap	78.954	0.813	0.789
15		Movement recognition	PCA	84.974	0.971	0.849
			LDA	80.492	0.942	0.804
			Isomap	85.280	0.975	0.852
16		Air quality sensor	PCA	87.704	0.999	0.877
			LDA	94.294	1.0	0.942
			Isomap	88.508	0.999	0.885
17		Energy prediction	PCA	95.929	0.996	0.959
			LDA	99.989	0.999	0.999
			Isomap	67.433	0.668	0.674
18		Big Sensors	PCA	78.642	0.985	0.786
			LDA	81.962	1.0	0.819
			Isomap	75.003	0.984	0.750
19		Transport detection	PCA	85.161	0.904	0.851
			LDA	71.192	0.900	0.711
			Isomap	84.384	0.896	0.843
20		Direction sensor	PCA	78.747	0.797	0.787
			LDA	69.876	0.739	0.698
			Isomap	78.009	0.800	0.780

Next, for stacking ensemble model, the study applied Random forest (RF), SVM, KNN, Bernoulli's Naïve Bayes (BNB), Gaussian Naïve Bayes (GNB), Gradient boosting (GBM), MLP and Logistic regression (LR) on each dataset. This study has visualized the data to see how the predictions from all eight models are different. For that, this study has used TSNE technique and created scatter plot which shows the predictions of different models. Then created a heatplot to compare the correlation of their prediction. Finally, frequencies of predicted classes is visualized using countplot of all classifiers. Detail information of Accuracy of eight classifiers of stacking model with PCA, LDA and Isomap is shown in Table 5. In the below Table 5, First 10 datasets are binary dataset and next 10 datasets are multiclass dataset.

Table 5: Accuracy of eight classifiers used for stacking model with PCA, LDA and Isomap.

No	Dataset	Feature Reduction Technique	RF	BNB	GNB	MLP	KNN	SVM	GBM	LR
1	Electric grid	PCA	86.40	80.15	88.05	93.70	90.40	95.05	88.65	86.85
		LDA	98.10	88.30	98.10	98.10	98.10	98.05	98.10	98
		Isomap	87.20	79.60	88.35	95.15	90.50	95.95	89.90	87.75
2	Extra sensory B	PCA	95.53	87.52	94.22	95.71	96.83	95.34	96.46	94.78
		LDA	96.46	96.27	96.08	96.46	97.20	96.46	95.71	96.64
		Isomap	95.53	80.44	91.24	95.15	97.02	95.34	95.15	94.59
3	Football sensor	PCA	89.94	88.88	90.47	89.94	90.47	89.41	89.41	89.41
		LDA	89.41	91.53	89.41	91.53	88.35	89.41	86.24	89.41
		Isomap	90.47	89.41	88.88	90.47	89.41	89.41	90.47	89.94

4	Pulsar star	PCA	95.90	92.59	92.95	96.73	96.94	97.15	96.99	96.58
		LDA	96.73	86.89	97.20	97.30	97.46	97.51	96.78	97.46
		Isomap	95.90	92.69	92.74	97.15	97.09	97.30	97.04	96.37
5	EEG signal	PCA	74.76	57.66	58.52	72.24	90.83	74.27	87.63	56.49
		LDA	61.35	61.41	61.29	61.41	58.21	61.41	61.16	61.47
		Isomap	78.15	60.06	62.58	86.58	93.35	88.80	82.33	65.10
6	Power system - A	PCA	78.89	71.73	67.57	78.60	88.09	77.44	84.12	76.08
		LDA	69.31	72.31	78.70	78.89	77.92	79.09	78.21	78.89
		Isomap	79.67	73.28	69.69	75.21	86.25	75.79	83.05	71.53
7	Hand gesture recognition	PCA	90.97	55.97	92.26	98.62	92.86	98.02	97.85	59.15
		LDA	71.28	65.09	71.79	70.67	66.80	71.88	71.45	63.62
		Isomap	93.72	84.35	94.32	96.99	94.58	98.28	97.42	91.31
8	Watch sensor	PCA	99.25	55.34	98.71	98.78	99.05	99.05	99.52	44.31
		LDA	77.74	52.43	78.28	78.01	76.79	77.53	77.80	48.44
		Isomap	99.52	62.44	96.61	98.98	99.45	99.25	99.45	58.52
9	Power system - B	PCA	74.94	75.25	71.42	77.16	89.33	77.66	86.01	76.55
		LDA	74.34	77.16	77.46	78.16	76.25	78.57	77.26	78.26
		Isomap	73.74	77.06	74.64	77.16	88.22	77.46	85.11	76.35
10	Machine sensor	PCA	63.41	65.53	57.76	67.93	87.05	67.93	75.18	64.54
		LDA	66.14	65.53	66.66	69.63	67.13	69.44	69.86	69.63
		Isomap	72.03	65.53	64.83	67.79	87.66	68.22	76.12	65.30
11	Cardiotocography sensor	PCA	99.53	96.71	100	99.76	100	100	98.82	100
		LDA	100	41.54	100	39.43	100	100	100	18.30
		Isomap	99.53	94.36	99.76	99.53	100	100	99.76	100
12	Extra sensory - A	PCA	69.27	72.62	70.01	72.43	72.25	71.88	72.43	73.37
		LDA	64.99	74.30	73.55	74.48	76.90	78.02	78.77	78.02
		Isomap	65.36	70.20	63.68	71.50	72.43	72.06	70.01	71.69
13	Mode detection	PCA	73.19	54.36	52.24	70.05	85.92	75.48	84.05	58.35
		LDA	64.46	52.07	60.55	63.95	71.84	64.46	70.56	62.34
		Isomap	68.10	49.36	54.36	74.72	85.07	75.65	81.34	57.42
14	Sky server	PCA	87.85	72.70	86	88.25	86.25	88.20	88.55	87.55
		LDA	94.80	91	93.75	92.35	93.95	94.85	94.75	94.10
		Isomap	82.45	75.55	79.65	85.15	84.85	85.80	85	82.95
15	Movement recognition	PCA	90.33	89.66	90	96.77	95.93	97.45	96.77	97.62
		LDA	83.38	70.50	81.69	79.49	82.88	83.38	83.22	84.06
		Isomap	89.32	88.47	91.18	96.27	92.88	97.28	95.08	96.77
16	Air quality sensor	PCA	79.86	81.35	85.04	88.94	90.33	89.69	90.38	88.62
		LDA	90.49	86.75	91.13	90.33	95.40	81.19	95.99	90.59
		Isomap	80.50	83.01	83.22	89.69	90.27	90.38	90.91	85.09
17	Energy prediction	PCA	85.38	81.63	86.59	99.34	85	99.41	97.87	97.28
		LDA	100	88.47	100	88.47	100	100	100	78.16
		Isomap	54.25	69.30	70.05	72.10	74.90	73.05	73.60	69.95
18	Big Sensors	PCA	82.11	82.43	87.05	94.19	87.92	95.05	89.88	95.52
		LDA	84.78	67.76	83.13	84.54	83.21	84.31	83.13	84.70
		Isomap	79.76	80.15	77.14	88.78	81.49	89.33	85.88	87.60
19	Transport detection	PCA	73.11	57.50	54.02	75.57	88.97	82.35	86.17	63.95
		LDA	71.58	55.89	63.18	67.93	77.26	70.73	76.08	66.75
		Isomap	76.33	48.17	58.86	82.18	87.53	82.27	85.83	61.74
20	Direction sensor	PCA	73.87	62.32	69.56	89.09	86.06	90.65	86.70	68.01
		LDA	67.09	63.24	66.17	68.46	76.07	71.58	73.14	66.54
		Isomap	67.73	60.86	65.81	76.16	83.95	85.42	83.86	67.18

To select three best performing classifiers among the eight classifiers, average value of accuracy rate is calculated for all eight classifiers for binary and multivariate datasets. Table 6 shows the information of average accuracy value of all eight classifiers on binary and multivariate datasets. As shown in Table 6, KNN, SVM and GBM are the three top performer classifiers in case of binary and multivariate dataset. Finally, the study selects these three classifiers for further processes.

Table 6: Average Accuracy value of all eight classifiers used for stacking model for binary and multivariate datasets.

No.	Dataset type	RF	BNB	GNB	MLP	KNN	SVM	GBM	LR
1	Binary	83.89	75.08	81.69	85.67	88.31	85.88	87.01	77.44
2	Multivariate	79.98	72.07	77.91	81.99	86.31	85.66	86.61	77.80

After selecting three classifiers, study builds one level prediction set for stacking classifier. This work created a level-1 train dataset using 5-fold cross-validation. Level-1 test dataset is created by selected models on complete original train dataset and predicted on test dataset. Finally,LR is trained as meta-classifier on level-1 train data and predicted on level-1test dataset. Table 7 shows Accuracy, AUC and F1 score of PCA, LDA and Isomap with stacking model to each dataset.

Table 7: Average Accuracy value of all eight classifiers used for stacking model for binary and multivariate datasets.

No.	Dataset type	Dataset	Feature Reduction Technique	Accuracy	AUC	F1 score
1	Binary	Electric grid	PCA	94.9	0.943	0.96
			LDA	98.05	0.977	0.984
			Isomap	96	0.954	0.968
2		Extra sensory - B	PCA	96.834	0.967	0.97
			LDA	96.664	0.965	0.968
			Isomap	96.275	0.961	0.964
3		Football sensor	PCA	89.947	0.897	0.886
			LDA	89.417	0.891	0.879
			Isomap	90.476	0.903	0.892
4		Pulsar star	PCA	96.996	0.903	0.876
			LDA	97.514	0.926	0.9
			Isomap	97.048	0.907	0.879
5		EEG signal	PCA	94.584	0.945	0.947
			LDA	60.984	0.611	0.58
			Isomap	93.6	0.935	0.937
6		Power system - A	PCA	88.189	0.825	0.766
			LDA	78.315	0.639	0.448
			Isomap	87.124	0.811	0.744
7		Hand-gesture recognition	PCA	99.14	0.991	0.991
			LDA	71.023	0.708	0.743
			Isomap	98.624	0.986	0.986
8		Watch sensor	PCA	99.526	0.995	0.995
			LDA	77.943	0.781	0.756
			Isomap	99.594	0.995	0.996
9		Power system - B	PCA	90.342	0.824	0.762
			LDA	77.364	0.561	0.257
			Isomap	88.631	0.81	0.729
10		Machine sensor	PCA	87.711	0.857	0.816
			LDA	69.585	0.624	0.473
			Isomap	88.182	0.86	0.822
11	Multivariate	Cardiotocography sensor	PCA	100	1	1
			LDA	100	1	1

			Isomap	100	1	1
12		Extra sensory - A	PCA	74.115	0.537	0.741
			LDA	73.929	0.981	0.739
			Isomap	62.569	0.609	0.625
13		Mode detection	PCA	87.955	0.955	0.879
			LDA	72.688	0.928	0.726
			Isomap	84.987	0.945	0.849
14		Sky server	PCA	89.2	0.89	0.892
			LDA	94.85	0.953	0.948
			Isomap	85.5	0.855	0.855
15		Movement recognition	PCA	97.796	1	0.977
			LDA	83.728	0.947	0.837
			Isomap	97.288	1	0.972
16		Air quality sensor	PCA	90.758	1	0.907
			LDA	95.779	1	0.957
			Isomap	90.705	1	0.907
17		Energy prediction	PCA	99.771	0.989	0.997
			LDA	100	1	1
			Isomap	75.4	0.935	0.754
18		Big Sensors	PCA	95.137	1	0.951
			LDA	83.686	1	0.836
			Isomap	89.647	0.995	0.896
19		Transport detection	PCA	91.009	0.97	0.91
			LDA	77.184	0.939	0.771
			Isomap	88.379	0.956	0.883
20		Direction sensor	PCA	91.384	0.905	0.913
			LDA	76.81	0.757	0.768
			Isomap	6.434	0.867	0.864

Next, for hybrid ensemble model, this work merges the prediction of bagging, boosting and stacking models were merged and created data of predicted classes. Here the concept of majority voting is used, that is if ‘yes’ is predicted more times than ‘no’ then ‘yes’ is chosen and vice versa. In binary datasets, it is certain to have a majority vote for either 0 or 1 class. But in multivariate datasets, which has 3 or more classes, a 3 ensemble models can predict different classes. In this scenario, there is no clear winner. Therefore, a class which has highest probability value is selected. Finally, it is tested data on original test dataset. The pseudocode of Hybrid ensemble model is shown below.

1. Initialise the My predict empty list
2. Initialise the Tmp pre and Tmp prob list contain 0 equal to number of classes present
3. For i in 0 to size of the test dataset
4. Increment the index of bagging, boosting and stacking predicted class by 1 inside the Tmp pred list
5. If all three models predicted different classes
6. Then assign probability values of bagging, boosting and stacking to Tmp prob list
7. Append the index of highest probability value to the My predict list
8. Else
9. Append the index of highest predicted class to the My predict list

10. End if

11. Clear the Tmp pre and Tmp prob list and initialise againt with 0

12. End For

13. Count how many times the test data and My predict list have the same value

Table 8 contain Accuracy, AUC and Prediction of hybrid ensemble model with PCA, LDA and Isomap.

Table 8: Accuracy, AUC and F1 score of PCA, LDA and Isomap with hybrid ensemble model to each dataset.

No.	Class types	Dataset	Feature Reduction Technique	Accuracy	AUC	F1 score
1	Binary	Electric grid	PCA	91	0.9	0.929
			LDA	98.05	0.977	0.984
			Isomap	91.95	0.909	0.937
2		Extra sensory - B	PCA	96.275	0.963	0.964
			LDA	95.53	0.954	0.957
			Isomap	96.461	0.964	0.966
3		Football sensor	PCA	89.417	0.891	0.879
			LDA	85.714	0.851	0.832
			Isomap	90.476	0.902	0.891
4		Pulsar star	PCA	97.203	0.911	0.886
			LDA	96.685	0.923	0.872
			Isomap	97.203	0.913	0.886
5		EEG signal	PCA	98.338	0.983	0.983
			LDA	54.892	0.548	0.556
			Isomap	94.523	0.945	0.946
6		Power system - A	PCA	93.61	0.906	0.879
			LDA	74.056	0.649	0.486
			Isomap	91.287	0.868	0.83
7		Hand-gesture recognition	PCA	97.076	0.97	0.971
			LDA	66.809	0.667	0.675
			Isomap	96.474	0.964	0.965
8		Watch sensor	PCA	99.458	0.994	0.994
			LDA	73.68	0.737	0.729
			Isomap	99.526	0.995	0.995
9		Power system - B	PCA	92.253	0.861	0.815
			LDA	74.647	0.612	0.397
			Isomap	92.857	0.873	0.831
10		Machine sensor	PCA	91.854	0.897	0.875
			LDA	65.301	0.595	0.447
			Isomap	91.384	0.892	0.868
11	Multivariate	Cardiotocography sensor	PCA	100	1	1
			LDA	100	1	1
			Isomap	100	1	1
12		Extra sensory - A	PCA	75.232	0.44	0.752
			LDA	78.584	0.527	0.785
			Isomap	71.322	0.522	0.713
13		Mode detection	PCA	89.228	0.959	0.892
			LDA	71.586	0.925	0.715
			Isomap	85.581	0.948	0.855

14		Sky server	PCA	89.4	0.895	0.894
			LDA	94.35	0.948	0.943
			Isomap	84.65	0.854	0.846
15		Movement recognition	PCA	93.22	0.989	0.932
			LDA	84.406	0.95	0.844
			Isomap	92.542	0.985	0.925
16		Air quality sensor	PCA	90.17	1	0.901
			LDA	95.673	1	0.957
			Isomap	90.384	1	0.903
17		Energy prediction	PCA	98.378	0.998	0.983
			LDA	100	1	1
			Isomap	75.95	0.929	0.759
18		Big Sensors	PCA	89.725	0.996	0.897
			LDA	81.803	1	0.818
			Isomap	83.764	0.993	0.837
19		Transport detection	PCA	91.687	0.962	0.916
			LDA	78.71	0.936	0.787
			Isomap	89.737	0.959	0.897
20		Direction sensor	PCA	1r|89.459	0.88	0.894
			LDA	77.451	0.82	0.774
			Isomap	88.267	0.876	0.882

For measuring the performance of models, this study has calculated average value Accuracy, AUC and F1 score of PCA, LDA and Isomap with all ensemble models on all binary and multivariate datasets. Below Table 9 shows average accuracy rate of PCA, LDA and Isomap with Bagging, Boosting, Stacking and hybrid ensemble model on all binary and multivariate IoT datasets.

Table 9: Average accuracy using PCA, LDA and Isomap with all ensemble models on binary and multivariate datasets

No.	Dataset Type	Feature Reduction Technique	Bagging Model	Boosting Model	Stacking Model	Hybrid Model
1	Binary	PCA average	93.879	91.185	93.816	94.648
		LDA average	79.117	77.029	81.685	78.536
		Isomap average	93.320	90.542	93.555	94.215
2	Multivariate	PCA average	88.963	84.523	91.712	90.649
		LDA average	86.276	82.970	85.865	86.256
		Isomap average	85.586	80.269	86.090	86.219

Below, Table 10 shows average AUC score of PCA, LDA and Isomap with Bagging, Boosting, Stacking and hybrid ensemble model on binary and multivariate IoT datasets.

Table 10: Average AUC using PCA, LDA and Isomap with all ensemble models on binary and multivariate datasets.

No.	Dataset Type	Feature Reduction Technique	Bagging Model	Boosting Model	Stacking Model	Hybrid Model
1	Binary	PCA average	0.915	0.895	0.914	0.927
		LDA average	0.757	0.741	0.768	0.751
		Isomap average	0.909	0.891	0.912	0.922
2	Multivariate	PCA average	0.903	0.897	0.924	0.911
		LDA average	0.907	0.904	0.950	0.910
		Isomap average	0.871	0.855	0.916	0.906

Below Table 11 shows average F1 score of PCA, LDA and Isomap with Bagging, Boosting, Stacking and hybrid ensemble model on all binary and multivariate IoT datasets.

Table 11: Average F1 Score using PCA, LDA and Isomap with all ensemble models on binary and multivariate datasets.

No.	Dataset Type	Feature Reduction Technique	Bagging Model	Boosting Model	Stacking Model	Hybrid Model
1	Binary	PCA average	0.902	0.871	0.896	0.917
		LDA average	0.701	0.683	0.698	0.693
		Isomap average	0.896	0.867	0.891	0.911
2	Multivariate	PCA average	0.889	0.844	0.916	0.906
		LDA average	0.861	0.829	0.858	0.862
		Isomap average	0.860	0.802	0.860	0.861

For better understanding, average value of Accuracy, AUC and F1 score is visualized. Below graphical representation shows visualisation of Table 9. This Figure 2 describes the average accuracy rate of PCA, LDA and Isomap with Bagging, Boosting, Stacking and Hybrid ensemble models on all binary and multivariate IoT datasets.

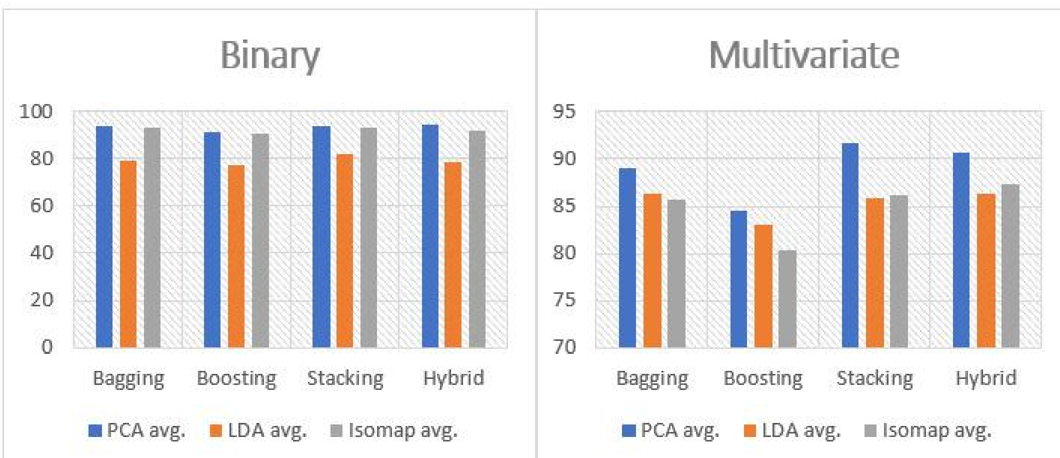

Figure 2: Average Accuracy for PCA, LDA and Isomap with all ensemble models on all binary and multivariate datasets.

Below graphical representation shows visualisation of Table 10. This Figure 3 describes the average AUC rate of PCA, LDA and Isomap with Bagging, Boosting, Stacking and Hybrid ensemble models on all binary and multivariate IoT datasets.

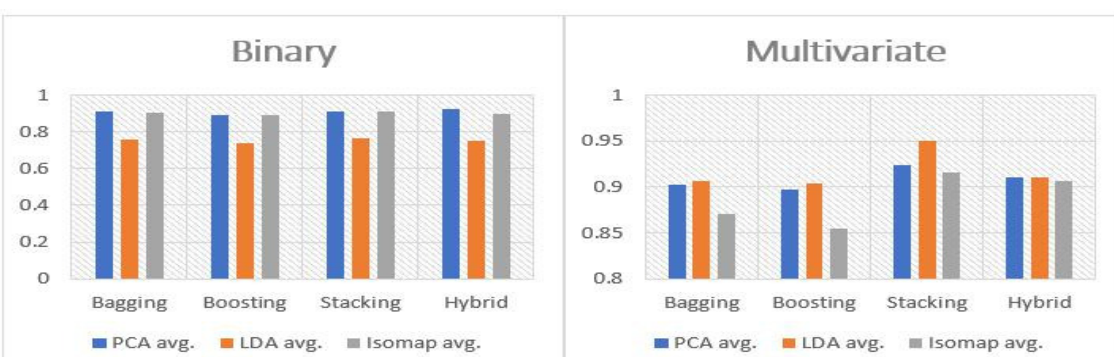

Figure 3: Average AUC for PCA, LDA and Isomap with all ensemble models on all binary and multivariate datasets.

Below graphical representation shows visualisation of Table 11. This Figure 4 describes average F1 score of PCA, LDA and Isomap with Bagging, Boosting, Stacking and Hybrid ensemble models on all binary and multivariate IoT datasets.

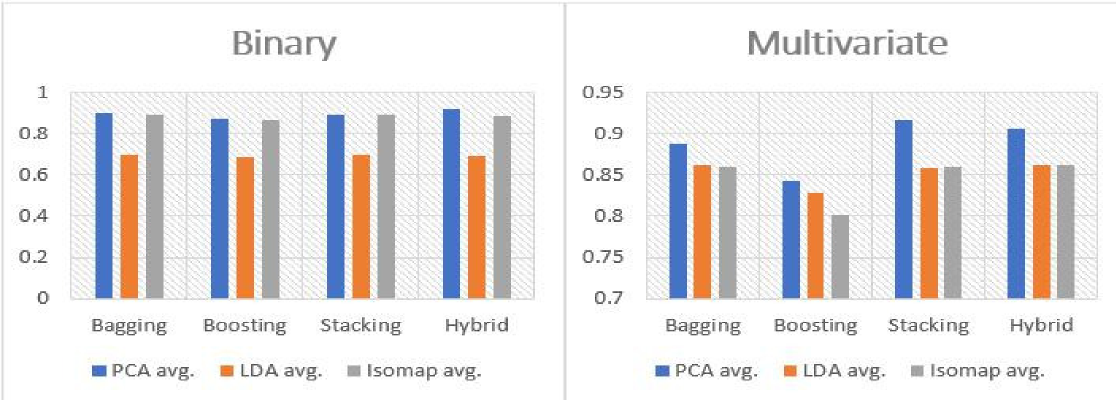

Figure 4: Average F1score for PCA, LDA and Isomap with all ensemble models on all binary and multivariate datasets

6. Observations and Conclusion

6.1 Observations

It is observed from the Table 9 that, for binary datasets, hybrid with PCA model achieved highest accuracy score with 94.468% and boosting with LDA model achieved lowest accuracy score with 77.029%. LDA average accuracy scores of all ensemble models are very less as compared to PCA and Isomap. For multivariate dataset, stacking with PCA model achieved top score with 91.712% while boosting with Isomap model get low score with 80.269%.

From Table 10 it is observed that, for binary datasets, hybrid model with PCA obtained the best average AUC score of 0.927 and boosting with LDA earned the lowest AUC average score of 0.683. Compared to LDA and Isomap average AUC scores of all ensemble models, PCA average scores of all ensemble models are relatively high. For multivariate datasets, Stacking with LDA performs the best and with 0.950 average AUC score while boosting with Isomap model received low average AUC score of 0.855.

It is futher observed from Table 11, for binary datasets, the hybrid model with PCA performed excellent and obtained 0.917 average F1 score while boosting with LDA model performed very poor and obtained a average F1 score of 0.683. For multivariate datasets, stacking model with PCA received highest mean F1 score of 0.916 and the boosting with Isomap model obtained the lowest mean F1 score of 0.802.

6.2 Conclusions

This comparative study investigated the possibility of applying bagging, boosting, stacking and hybrid ensemble algorithms with PCA, LDA and Isomap to improve the performance on IoT sensor datasets. In both binary and multivariate cases, PCA ts perfectly to all ensemble models compared to LDA and Isomap. For binary datasets, Hybrid with PCA works the best against other models. Boosting with LDA performed ineffectively compared to other models. For multivariate datasets, Stacking with PCA performed better than other models in question and close runner up is Hybrid with PCA. Boosting with Isomap worked very poorly in case multivariate datasets. Bagging performed average in binary as well as multivariate datasets.

References

*[1] Michie, D., Spiegelhalter, D. J., & Taylor, C. C, "Machine learning", Neural and Statistical Classification, 13(**1994**), 1-298.*

[2] Zhou, Zhi-Hua, "Ensemble Learning", Encyclopedia of biometrics 1 **(2009)**: 270-273.

[3] Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." Chemometrics and intelligent laboratory systems 2, no. 1-3 **(1987)**: 37-52.

[4] Balakrishnama, S., & Ganapathiraju, A. “Linear discriminant analysis-a brief tutorial” In Institute for Signal and information processing Vol. 18**,(1998)**, pp. 1-8).

[5] Balasubramanian, Mukund, Eric L. Schwartz, Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. "The isomap algorithm and topological stability." Science 295, no. 5552 **(2002)**: 7-7.

[6] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9, no. Nov **(2008)**: 2579-2605.

.

[7] Ranzato, Francesco, and Marco Zanella. “Abstract interpretation of decision tree ensemble classifiers", In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, (**2020**).

[8] Dinakaran, S., and P. Ranjit Jeba Thangaiah, "Ensemble method of effective AdaBoost algorithm for decision tree classifiers." International Journal on Artificial Intelligence Tools 26, no. 03 **(2017**): 1750007.

[9] O’Brien, R., & Ishwaran, H. ,“A random forests quantile classifier for class imbalanced data”, Pattern recognition, 90 **(2019)**, pp232-249.

[10] Sayfullina, L., Eirola, E., Komashinsky, D., Palumbo, P., Miche, Y., Lendasse, A., & Karhunen, J. “Efficient detection of zero-day android malware using normalized bernoulli naive bayes”, In 2015 IEEE Trustcom/BigDataSE/ISPA (Vol. 1, pp. 198-205). IEEE. (**2015)**

[11] Ontivero-Ortega, M., Lage-Castellanos, A., Valente, G., Goebel, R., & Valdes-Sosa, M. , “Fast Gaussian Naïve Bayes for searchlight classification analysis”, Neuroimage, 163 (**2017),** 471-479.

[12] Pal, S. K., & Mitra, S., “Multilayer perceptron, fuzzy sets, classification”,**(1992**)

[13] Zhang, Y., Cao, G., Wang, B., & Li, X. ,“A novel ensemble method for k-nearest neighbour. Pattern Recognition”, 85(2019), 13-25.

[14] Pradhan, Ashis, "Support vector machine-a survey." International Journal of Emerging Technology and Advanced Engineering 2, no. 8 **(2012)**: 82-85

[15] He, Zhiyuan, Danchen Lin, Thomas Lau, and Mike Wu, "Gradient Boosting Machine: A Survey", arXiv preprint arXiv:1908.06951 **(2019)**.

[16] Kleinbaum, David G., K. Dietz, M. Gail, Mitchel Klein, and Mitchell Klein, “Logistic regression”, New York: Springer-Verlag, **(2002)**

[17] Junior, J. J. A. M., Freitas, M. L., Siqueira, H. V., Lazzaretti, A. E., Pichorim, S. F., & Stevan Jr, S. L., “Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by sEMG in eight channels armband approach”, Biomedical Signal Processing and Control, 59 **(2020)**, 101920.

[18] Ribeiro, M. H. D. M., & dos Santos Coelho, L., “Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series”, Applied Soft Computing, 86 **(2020)**, 105837.

[19] Taşer, Pelin Yildinm, Kökten Ulaş Birant, and Derya Birant, "Comparison of Ensemble-Based Multiple Instance Learning Approaches." In 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1-5. IEEE, **(2019)**

[20] Alexandropoulos, Stamatios-Aggelos N., Christos K. Aridas, Sotiris B. Kotsiantis, and Michael N. Vrahatis, "Stacking strong ensembles of classifier." In IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 545-556. Springer, Cham, **(2019)**

[21] Suganthi, M., and V. Karunakaran, "Instance selection and feature extraction using cuttlefish optimization algorithm and principal component analysis using decision tree", Cluster Computing 22, no. 1 **(2019)**: 89-101.

[22] Tounsi, Youssef, Larbi Hassouni, and Houda Anoun. "An Enhanced Comparative Assessment of Ensemble Learning for Credit Scoring." International Journal of Machine Learning and Computing vol 8, no. 5 **(2018)**.

[23] Vijai, Praveen, "Performance comparison of feature reduction techniques in-terms of compactness, computation time and accuracy", In 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 374-380. IEEE, **(2018)**.

*[24] Rojarath, Artittayapron, Wararat Songpan, and Chakrit Pong-inwong, "Improved ensemble learning for classification techniques based on majority voting", In 2016 7th IEEE international conference on software engineering and service science (ICSESS), pp. 107-110. IEEE, **(2016)**.*

[25] Yu, Zhiwen, Daxing Wang, Jane You, Hau-San Wong, Si Wu, Jun Zhang, and Guoqiang Han, "Progressive subspace ensemble learning", Pattern Recognition 60 (2016): 692-705.

*[26] Narassiguin, Anil, Mohamed Bibimoune, Haytham Elghazel, and Alex Aussem, "An extensive empirical comparison of ensemble learning methods for binary classification ", Pattern Analysis and Applications 19, no. 4 **(2016).***

*[27] Pandey, Mrinal, and S. Taruna, "A comparative study of ensemble methods for students' performance modeling" , International Journal of Computer Applications 103, no. 8 **(2014).***

*[28] Wan, S., & Yang, H. (2013, July). Comparison among methods of ensemble learning. In 2013 International Symposium on Biometrics and Security Technologies (pp. 286-290). IEEE,**(2013)**.*

*[29] Ye, Ren, and Ponnuthurai N. Suganthan, "Empirical comparison of bagging-based ensemble classifiers", In 2012 15th International Conference on Information Fusion, pp. 917-924. IEEE, **(2012)**.*

*[30] Syarif, Iwan, Ed Zaluska, Adam Prugel-Bennett, and Gary Wills, "Application of bagging, boosting and stacking to intrusion detection", In International Workshop on Machine Learning and Data Mining in Pattern Recognition, pp. 593-602. Springer, Berlin, Heidelberg, **(2012)**.*

*[31] Wang, Gang, Jinxing Hao, Jian Ma, and Hongbing Jiang. "A comparative assessment of ensemble learning for credit scoring", Expert systems with applications 38, no. 1 **(2011)**.*

*[32] Graczyk, Magdalena, Tadeusz Lasota, Bogdan Trawiński, and Krzysztof Trawiński, "Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal", In Asian conference on intelligent information and database systems, Springer, Berlin, Heidelberg, **(2010)**.*

*[33] Dong, Yan-Shi, and Ke-Song Han, "A comparison of several ensemble methods for text categorization", In IEEE International Conference onServices Computing, 2004.(SCC 2004). Proceedings. IEEE,**(2004)**.*

*[34] Asuncion, Arthur, and David Newman, "UCI machine learning repository", **(2007)**.*

*[35] Breiman, Leo. "Bagging predictors", Machine learning 24, no. 2 **(1996)**: 123-140.*

*[36] Freund, Yoav, and Robert E. Schapire. "Experiments with a new boosting algorithm." In ICML, vol. 96, **(1996)**.*

*[37] Wolpert, David H, "Stacked generalization", Neural networks 5, no. 2 **(1992):** 241-259.*

*[38] Amasyali, M. Fatih, and Okan Ersoy. "Cline: A new decision-tree family." IEEE Transactions on Neural Networks 19, no. 2 **(2008)**: 356-363.*

*[39] Wilamowski, Bogdan M. "Neural network architectures and learning algorithms." IEEE Industrial Electronics Magazine 3, no. 4 **(2009)**: 56-63.*

*[40] Mason, Llew, Jonathan Baxter, Peter L. Bartlett, and Marcus R. Frean. "Boosting algorithms as gradient descent." In Advances in neural information processing systems, pp. 512-518. **(2000)**.*

*[41] Majka, Michal, and Maintainer Michal Majka. "Package 'naivebayes' ", **(2018)**.*

*[42] Aha, David W., and Dennis F. Kibler. "Noise-Tolerant Instance-Based Learning Algorithms." In IJCAI, **(1989).***

*[43] Gay, David M., and Roy E. Welsch. "Maximum likelihood and quasi-likelihood for nonlinear exponential family regression models." Journal of the American Statistical Association 83, no. 404 **(1988)**: 990-998.*

*[44] Tyagi, Kanishka, and Michael Manry. "Multi-step training of a generalized linear classifier." Neural Processing Letters 50, no. 2 **(2019):** 1341-1360.*

*[45] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." The Journal of machine Learning research 12 **(2011)**: 2825-2830.*

*[46] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An introduction to statistical learning. Vol. 112. New York: springer, **(2013)**.*

*[47] Sultana, Naznin, and Mohammad Mohaiminul Islam. "Meta Classifier-Based Ensemble Learning For Sentiment Classification." In Proceedings of International Joint Conference on Computational Intelligence, pp. 73-84. Springer, Singapore, **(2020)**.*