**Noname manuscript No.**
(will be inserted by the editor)

# Enhancing the performance of classification using Super Learning

**Md Faisal Kabir and Simone A. Ludwig**

Received: date / Accepted: date

**Abstract** Classification is one of the supervised learning models, and enhancing the performance of a classification model has been a challenging research problem in the fields of Machine Learning (ML) and data mining. The goal of ML is to produce or build a model that can be used to perform classification. It is important to achieve superior performance of the classification model. Obtaining a better performance is important for almost all fields including healthcare. Researchers have been using different ML techniques to obtain better performance of their models; ensemble techniques are also used to combine multiple base learner models. The ML technique called super learning or stacked-ensemble is an ensemble method that finds the optimal weighted average of diverse learning models. In this paper, we have used super learning or stacked-ensemble achieving better performance on four benchmark data sets that are related to healthcare. Experimental results show that super learning has a better performance compared to the individual base learners and the baseline ensemble.

**Keywords** Super Learning / Learner · Stacked Ensemble · Classification

## 1 Introduction

Machine Learning (ML) or Data Mining (DM) algorithms [1] [2] can be classified into supervised or unsupervised learning depending on the goal of the data mining task. Supervised methods are used when there is a variable whose value has to be predicted. Such a variable is referred to as a response or output

Md Faisal Kabir and Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, ND, USA
E-mail: {mdfaisal.kabir,simone.ludwig}@ndsu.edu

variable. For an unsupervised method, the data is not labeled and there is no value to predict or classify.

The goal of classification, which is a supervised learning technique, is to predict quantitative or categorical outputs that assume values in a finite set of classes (e.g. Yes/No or Green/Red/Blue etc.) without an explicit order. Classification is the task of learning a target function $f(x)$ that maps each attribute set $x$ into one of the pre-defined class labels $y$ [3] [4]. The target function is also informally known as the classification model.

There are several classification models or learning algorithms available, and researchers are using these algorithms in different fields such as healthcare, network security, and business. Researchers are trying to find which algorithm will perform well for a particular research problem and the available data at hand. The main objective of ML techniques is to produce a model that can be used to perform classification, prediction, estimation, or any other similar task [2]. The most common task in the learning process is classification. It is important to estimate the classifier's performance from the classification model. The performance analysis of the model is generally measured in terms of sensitivity, specificity, overall accuracy, and Area Under the Curve (AUC) [5]. Achieving better performance using the model is the key for unseen data. To achieve a better performance for the available data sets, researchers are using an appropriate single classifier. However, selecting the best ML model for a specific problem is a complex task and there is no direct or effortless solution for addressing various issues simultaneously. Indeed, even if multiple models could be very well suited for a particular problem, it may be very difficult to find one which performs optimally for different distributions. The ensemble learning model permits to combine more than one model or classifier to form a better model. Ensemble machine learning methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms. However, ensemble learning has a higher computational cost and complexity than single base learning approaches. This constraint is no more problematic due to the availability of current big data platforms and off-the-shelf data processing technology that is mature enough to allow for fast and parallel operation of multiple algorithms [6].

Many of the popular modern machine learning algorithms are actually ensembles. Researchers have been using Bagging (Random Forest) and Boosting (Gradient Boosting Machine) ensemble techniques in different fields including the medical domain to get better performance [7] [8] [9]. A super learning or stacking method that ensembles a group of base learners are also used by researchers to obtain a better predictive performance [10] [11]. The super learning algorithm is a loss-based supervised learning method that finds the optimal combination of a collection of prediction algorithms. The method performs asymptotically as well as the best possible weighted combination of the base learners, thus, providing a very powerful approach to tackle multiple problems with the same technique. In addition, it defines an approach to minimize the likelihood of over-fitting during training, using a variant of cross-validation [10] [11].

In this paper, we present two different forms of super learner or stacked ensemble. First one uses two base learners namely Gradient Boosting Machine (GBM) and Random Forest (RF), and the second one uses three base learners namely GBM, RF and Deep Neural Network (DNN); and for both cases a meta-learner called Generalized Linear Model (GLM) is used [12] [13]. We use four well-known benchmark data sets related to the healthcare area and compare the performance of both super learners with the individual base learners, baseline ensemble and the state-of-the-art classifiers. Our evaluations confirm that the super learner method has the ability to perform better compared to individual base learners, baseline ensemble approach, and some of the state-of-the-art techniques on four benchmark data sets.

The rest of the paper is organized as follows. Section 2 describes state-of-the-art techniques; Section 3 presents background of the algorithms used. Section 4 is our proposed solution section, in which our proposed framework is discussed in detail. Section 5 shows the experimental results; the proposed techniques are evaluated using four benchmark data sets and their results are presented. Section 6 is the summary section of this paper; we conclude our paper and suggest possible future research directions.

## 2 Related Work

In this research, a ML technique called super learning or stacked ensemble [10] [15] [16] has been used to improve the performance of four benchmark data sets related to healthcare. Stacked generalization in the context of neural net ensembles used leave-one-out Cross-Validation (CV) to generate level-one data [17], which is the cross-validated predicted values generated from cross-validating base learners on the training data. The authors extended the previous stacking framework [17] to regression problems [18] and proposed to use k-fold CV to generate level-one data. In this work, the authors also suggested non-negativity constraints for the meta-learner. It was proposed combining estimates in regression and classification that provided a general framework for stacking and compared CV-generated level-one data to bootstrapped level-one data [19]. Ensemble or combining learners in various methods showed better performance over a single candidate learner, but there is a concern that these methods may over-fit the data and may not be the optimal way to combine the candidate learners [10]. Researchers suggest a solution to this problem in the form of a new learner and named it super learner. In the context of prediction, a super learner is itself a prediction algorithm, which applies a set of candidate learners to observed or training data, and chooses the optimal learner for a given prediction problem based on the cross-validated risk. Theoretical results show that the super learner will perform asymptotically as well as or better than any other candidate learners [10] [20].

Using super learning for dynamic accuracy prediction in various domains is becoming popular. Researchers have used a super learning model to enhance anomaly detection in cellular networks [21]. It was also used in predicting violence among inmates from the 2005 census of state and federal adult correctional facilities [22]. Researchers investigated different ensemble learning methods including super learning for network security and anomaly detection. In their paper, they showed that the super learner provides better results than any of the single models like Nave Bayes (NB), Decision Tree (DT), Neural Network (NN), Support Vector Machine (SVM), K-nearest Neighbors (KNN) and RF [11].

Different ML and DM techniques have been developed and used in various data sets in healthcare. Researcher used ensemble-based techniques with 10 fold cross-validation on Messidor data for enhancing the performance [23]. Classifier methods like multi-layer perceptron (MLP), and NB have been used to assess the performance of the Wisconsin breast cancer (WBC) data sets [24]. Sequential minimal optimization (SMO) technique, which is an optimization algorithm widely used for training SVM, has also been used to assess the performance of the WBC data set [24]. In addition, bagging and boosting methods have been used to compare the performance of the WBC data set [7]. The NB classifier has been used on the Pima Indian Diabetes Dataset (PIDD). In order to get superior performance over the NB classifier, researchers used a Genetic Algorithm (GA) approach for attribute or feature selection [25]. For the Indian Liver Patient Dataset (ILPD) data set, authors used an ensemble classifier with 5 fold cross-validation and obtained acceptable results [26]. Researchers showed the comparative analysis of diverse ML algorithms like NB, SVM, MLP, random forest (RF) for various data sets including ILPD with the best accuracy for ILPD using SVM [27].

In this paper, we used the super learner or stacked ensemble approach that is discussed in the following section applied to the four benchmark data sets.

## 3 Methodology

Super learning or stacked ensemble is a ML method that uses two or more learning algorithms. It is a loss-based supervised learning method that finds the optimal combination of a collection of prediction algorithms. It is a cross-validation-based approach for combining machine learning algorithms that produce predictions that are at least as good as those of the best input algorithm [10] [11].

### 3.1 Super Learning or Stacking

Stacking is a broad class of algorithms that involves training a second-level meta-learner of an ensemble. Super learning or stacking [10] is a procedure for ensemble learning in which a meta-learner is trained on the output of a collection of base learners. The output from the base learners, also called the level-one data, can be generated using cross-validation. Construction of level-one data is discussed in the following section. The original training data set is often referred to as the level-zero data. The pseudo-code of the super learning or stacking is shown in Algorithm 1 [15] [16], and the concept diagram of the super learning method is illustrated in Fig. 1.

**Algorithm 1** Super learning Algorithm

```
1: Input: data set $D$ with set of $X$ examples, and response column $Y$.
2: Output: ensemble-model.

3: Set up the ensemble
     – Specify a list of $L$ base algorithms (with a specific set of model parameters).
     – Specify a meta-learning algorithm.

4: Train the ensemble.
     – Train each of the L base algorithms on the training set.
     – Perform k-fold cross-validation on each of the $L$ learners, and collect the cross-
       validated predicted values from k-fold CV that was performed on each of the $L$ base
       learners.
     – The N cross-validated predicted values from each of the $L$ algorithms can be com-
       bined to form a new matrix, $Z(NXL)$. This matrix $Z$, along with the original re-
       sponse vector is called the "level-one" data (N = number of instances in the training
       set).
     – Train the meta-learning algorithm on the level-one data ($Z$,$Y$). The ensemble model
       consists of the $L$ base learning models, and the meta-learning model, which can then
       be used to generate predictions on a test set.

5: Predict new data.
     – To generate ensemble predictions, first generate predictions from the base learners.
     – Feed those predictions into the meta-learner to generate the ensemble predictions.
```

### *3.1.1 Constructing level-one data*

The super learner theory requires cross-validation to generate the level-one data. Assume that the training set is comprised of $n$ independent, and identical distributed observations, $\{O_1, O_2, O_3\}$ where $O_i = (X_i, Y_i)$ here $X_i$ is the feature value, and $Y_i$ is the outcome or class value [15] [16]. Consider an ensemble comprised of a set of $L$ base learning algorithms, $\{B_1, B_2, ..., B_L\}$ each of which is indexed by an algorithm class, and a specific set of model parameters. Then, the process of constructing the level-one data will involve generating a $n \times L$ matrix, referred to as $Z$ of the k-fold cross-validated predicted values as follows:

1. The original training set $X$ is divided at random into $k = V$ roughly-equal pieces $X(1), X(2), ..., X(V)$.
2. For each base learner in the ensemble, $B_L$ V-fold cross-validation is used to generate $n$ cross-validated predicted values associated with the $l^{th}$ learner.

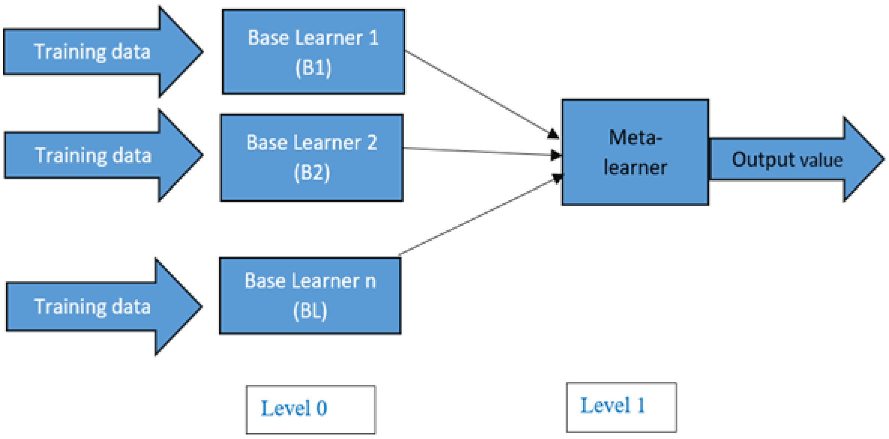

Fig. 1: Concept Diagram of Super Learner

These n-dimensional vectors of cross-validated predicted values become the $L$ columns of $Z$.

The level-one data set $Z$, along with the original outcome vector $\{Y_1, Y_2, ..., Y_n\}$, is used to train the meta-learning algorithm. Finally, each of the $L$ base learners are fitted to the full training set and these fits are saved. The final ensemble fit is comprised of the $L$ base learner fits, along with the meta-learner fit. To generate a prediction for new data using the ensemble, the algorithm first generates the predicted values from each of the $L$ base learner fits, and then passes those predicted values as input to the meta-learner fit, which returns the final predicted value for the ensemble.

### *3.1.2 Base learners*

It is recommended that the base learners should include a diverse set of learners, for example, linear model, SVM, RF, GBM, Neural Net, etc., however, the super learner theory does not require any specific level of diversity among the set of base learners [15] [16]. It is also allowable to include the same algorithm multiple times as a base learner by different sets of parameters. For example, the user could specify multiple Distributed Random Forest (DRF) method, each with a different splitting criterion, tree depth, number of folds, or number of trees. Typically, in stacking-based ensemble methods, the prediction functions are fit by training each base learning algorithm on the whole training data set and then combining these fits using a meta-learning algorithm. In this paper, we first used two base learners namely Gradient Boosting Machine (GBM) and Distributed Random Forest (RF). In addition, we used another base learner called Deep Neural Network (DNN) with GBM and RF that are briefly discussed below.

**Gradient Boosting Machine (GBM)** [12] produces a prediction model in the form of an ensemble of weak prediction models. It builds the model in a stage-wise fashion and is generalized by allowing an arbitrary differentiable loss function. It is one of the most powerful methods available today. GBM for regression and classification is a forward learning ensemble method. The guiding heuristic is that good predictive results can be obtained through increasingly refined approximations. H2O is an open source, in-memory, ML, and predictive analytics platform [12] which is used in this work. GBM is available in H2O, which is built upon the distributed, open source, Java-based machine learning platform for big data [12]. H2O's GBM sequentially builds regression trees on all the features of the data set in a fully distributed way - each tree is built in parallel. Additional features have been incorporated into the new version of H2O like the per-row observation weights, per-row offsets, N-fold cross-validation, and support for more distribution functions (such as Gamma, Poisson, and Tweedie).

**Distributed Random Forest (DRF)** [12] is a powerful classification and regression tool. When given a set of data, Random Forest (RF) generates a forest of classification (or regression) trees, rather than a single classification (or regression) tree. Each of these trees is a weak learner built on a subset of rows and columns. More trees will reduce the variance. Both classification and regression take the average prediction over all of their trees to make a final prediction, whether predicting a class or numeric value. For a categorical response column, DRF maps factors (e.g. 'dog', 'cat', 'mouse') in lexicographic order to a name lookup array with integer indices (e.g. 'cat' - 0, 'dog' - 1, 'mouse' - 2).

**Deep Neural Network (DNN)** [13] is an architecture of deep learning based on an Artificial Neural Network (ANN) that is inspired by biological neural networks. A DNN has basically many connected units arranged in layers of varying sizes with information being fed forward through the network. DNNs have been successfully applied to fields such as computer vision and natural language processing system and achieved better or similar accuracy rates compared to humans in classification tasks[14]. H2O's deep learning is based on a multi-layer feedforward ANN that is trained with stochastic gradient descent using back-propagation. The network can contain a large number of hidden layers consisting of neurons with activation functions such as tanh, rectifier, and maxout. Advanced features such as dropout, L1 or L2 regularization, grid search, etc. enable high predictive accuracy.

#### *3.1.3 Meta-learning algorithm*

The meta-learner is used to find the optimal combination of the $L$ base learners. The $Z$ matrix of cross-validated predicted values, described previously, is used as the input for the meta-learning algorithm along with the original outcome from level-zero training data $\{Y_1, Y_2, ..., Y_n\}$. In the super learning algorithm, the meta-learning method is specified as the minimizer of the cross-validated risk of a loss function of interest, such as squared error loss or rank loss. Historically, in stacking implementations, the meta-learning algorithm is often some sort of regularized linear model, however, a variety of parametric and non-parametric methods can be used as a meta-learner to combine the output from the base fits [15] [16]. For this paper, we used Generalized Linear Models (GLM) as the meta-learner, which is described briefly as follows.

**Generalized Linear Models (GLMs)** are an extension of traditional linear models. They have gained popularity in statistical data analysis due to the following three characteristics [13]. Firstly, the flexibility of the model structure unifying the typical regression methods (such as linear regression, and logistic regression for binary classification). Secondly, the recent availability of model-fitting software, and finally, the ability to scale well with large data sets.

GLM provides flexible generalization of ordinary linear regression for response variables with error distribution models other than a Gaussian (normal) distribution. GLM's estimate regression models for outcomes follow exponential distributions. In addition to the Gaussian (i.e. normal) distribution, these include Poisson, binomial, and gamma distributions. Each serves a different purpose, and depending on the distribution and link function choice, either can be used for prediction or classification [12].

### 3.2 Proposed Approach

To obtain better performance, we selected three base learners from H2O namely Gradient Boosting Machine (GBM), Random Forest (RF), and Deep Neural Network (DNN) [12]. For the meta-learner, we used Generalized Linear Model (GLM) [12] [13]. It is a particular implementation of the Super Learner, using a probability-based weighting function to combine the outputs of the first level learners. In a nutshell, we used the probabilities of success of each class to build exponentially decayed weighting functions, adding a control variable to reduce the overall influence of low accuracy models in the final prediction.

Our proposed method has the following main steps:

1. Classification Model Data and Sample Data for Classification
   - We construct the classification model data and sample data for classification whereby for the training data set the class information is known whereas the class information is unknown for the testing data set. The data sets are referred to as level-0 data, which is shown in Fig. 2 where $X$ is the training data set with $n$ rows, and $m$ columns; the class value column is separated from the training data, which is referred to as $Y$.
2. Classifiers and Model Selection
   - To set up the stacked ensemble or super learner, we need to specify the base learners and a meta-learner algorithm. For this research, we first selected two base learners namely GBM and RF. We also selected

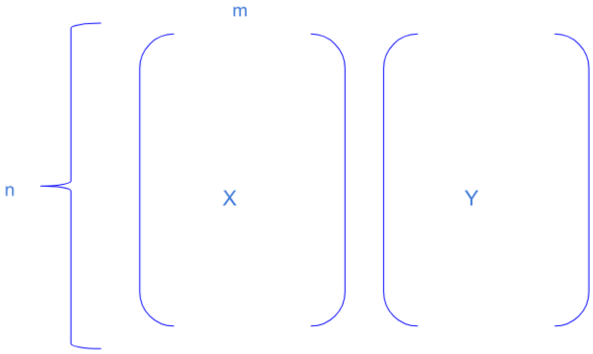

Fig. 2: Level-0 data

another base learner DNN with the previous two base learners and for the meta-learner we specified GLM.
For the model selection process, we used the cartesian grid search and specified a set of values for particular parameters to search over each base learner. The parameters that underwent a model selection phase are shown in Table 1 with the corresponding range of values. After preliminary experiments, parameters were set as fixed values are also shown in Table 1. For the meta learner algorithm, we used the default parameters available in H2O. The training of the ensemble has the following two steps:

Table 1: Classifiers with the corresponding hyper-parameter values.

| **Classification algorithm** | **Hyper-parameters in grid search with the corresponding range of values** | **Hyper-parameters fixed values** |
|---|---|---|
| GBM | learn_rate: [0.01, 0.03, 0.05, 0.1]<br>sample_rate: [0.5, 0.6, 0.7]<br>col_sample_rate_per_tree: [0.7,0.8, 0.9]<br>max_depth: [6,8, 10] | ntrees = 80<br>nfolds = 10<br>fold_assignment = Modulo<br>keep_cross_validation_predictions=True |
| RF | sample_rate: [0.5, 0.6, 0.7]<br>col_sample_rate_per_tree: [0.7,0.8, 0.9]<br>max_depth: [6,8, 10] | ntrees = 100<br>nfolds = 10<br>fold_assignment = Modulo<br>keep_cross_validation_predictions=True |
| DNN | activation: [tanh,rectifier, maxout]<br>hidden: [50]<br>l1: [0,1e-3, 1e-5]<br>l2: [0,1e-3, 1e-5] | epochs = 20<br>nfolds = nfolds<br>fold_assignment = Modulo<br>keep_cross_validation_predictions=True |

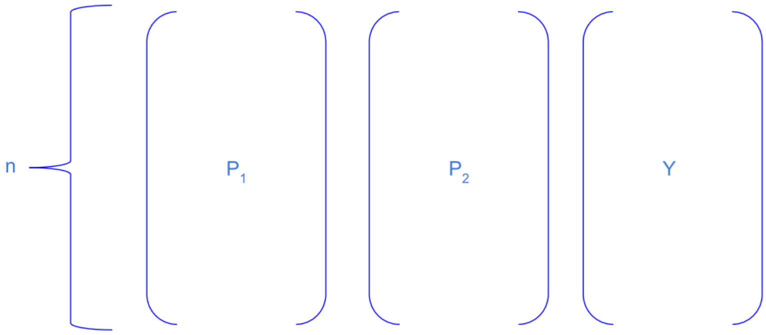

Fig. 3: Level-1 data for two base learners (GBM and RF).

(a) Base learners

- We trained GBM, RF, and DNN individually on the training data set with the specific parameters obtained using the grid search. Here, 10-fold cross-validation is performed on each of these learners and we kept the cross-validation prediction parameter specified as True. For all three base learners, the Bernoulli distribution was specified since the response column is of type categorical with two classes. In addition, for the base learners the fold-assignment modulo was selected which is a simple deterministic way to evenly split the data set into the folds. It is important to note that in our experiments we first used two base learners (GBM and RF) and then three base learners (GBM, RF, and DNN). The $N$ cross-validated predicted values of the three base learners GBM, RF, and DNN are defined as P1, P2, and P3 respectively. For the ensemble consisting of two base learners (GBM and RF), the predicted values P1 and P2 are combined to form a $n \times 2$ matrix. This matrix along with the class value $(Y)$ of the training data is called the level-1 data for the ensemble having two base learners, which is shown in Fig. 3.

  For the stacked ensemble consisting of three base learners, level-1 data is constructed similarly. However, instead of using the cross-validated predicted values P1 and P2, we used P1, P2, and P3, which are combined to form a $n \times 3$ matrix. This matrix along with the class value $(Y)$ of the training data is called the level-1 data for the ensemble having three base learners, which is shown in Fig. 4.

  Please note that for each of the base learners the best model was selected based on the mean squared error (MSE) which is the average squared difference between the estimated values and the actual values. This was done once the grid search on the training data was complete, and then we queried the grid

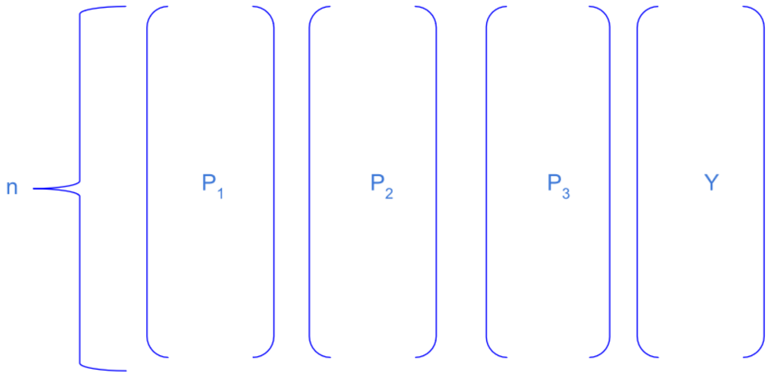

Fig. 4: Level-1 data for three base learners (GBM, RF, and DNN).

object and sorted the models by the performance metric MSE. Finally, for each base learner the model having the minimum MSE was selected.

(b) Meta-learner

– In the stacked ensemble, for the meta-learner we have used GLM available in H2O. We trained the level-1 data using GLM with default parameters to get the prediction values for the training data set. Firstly, for the stacked ensemble having two base learners GBM and RF were used for the parameter named base model with the other specified default parameters discussed in *Step (a)*. Secondly, for the stacked ensemble having three base learners GBM, RF and DNN were used for the parameter named as base model.

It is important for the stacked ensemble that all base models must have been cross-validated and they all must use the same cross-validation folds. Also, a parameter named ‘keep cross-validation prediction’ was set to True. In our case, we considered that by using 10 fold cross-validation and setting the ‘keep cross-validation prediction’ parameter as True for all the base learners.

3. Output generation / results stage
   – The last part of our approach was to use the super learner or ensemble-model to generate predictions on the test data.

## 4 Experiments and Results

This section presents the experimental results and performance evaluation of our model. For our experiment we used H2O. We chose Python as the programming language for the implementation using H2O.

### 4.1 Benchmark Data Sets

To evaluate the performance of our model, we used four benchmark data sets related to healthcare. The data sets were chosen from the UCI Machine Learning repository [23], [28], and Kaggle [29]. The first data set named Diabetic Retinopathy Debrecen data, also called Messidor data set, contains features extracted from the Messidor image set to predict whether an image contains signs of Diabetic Retinopathy (DR) or not. It has a total of 1151 instances, 19 attributes, and a class label with binary outcome 1 or 0, where 1 represents 'sign of DR' and 0 represents 'no sign of DR'. The second data set that we used is the original Wisconsin Breast Cancer (WBC) data set. The goal of this data set is to predict breast cancer. There are 699 records in this database. Each record in the database has nine attributes. In this database, there are a total of 699 instances, among them 241 (65.5%) records are malignant and 458 (34.5%) records are benign. We also used the Pima Indian Diabetes Database (PIDD) and the objective of this data set is to predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the data set. Various constraints were placed on the selection of these instances from a large database. For example, all patients should include female patients who are at least 21 years old and of Pima Indian heritage. There is a total of 768 records with 268 (34.9%) diabetes patients and 500 (65.1%) non-diabetes patients. The final data set that we used in our evaluation process is the Indian Liver Patient data set (ILPD) that contains 10 variables and a binary variable as output (liver patients or not). The data set contains 441 male and 142 female patient records. There are a total of 583 records with 416 (71.4%) liver patients and 167 (28.6%) non-liver patients. The summary of these four data sets is shown in Table 2.

Table 2: Data sets description.

| Name | Number of instances | Number of attributes | Class label with number of instances |
|---|---:|---:|---|
| Messidor | 1151 | 9 | Class 0: no sign of DR (540); Class 1: contains sign of DR (611) |
| WBC | 699 | 19 | Class 2: benign (458); Class 4: malignant (241) |
| Pima Indian Diabetes | 768 | 8 | Class 0: non-diabetes patients (500); Class 1: diabetes patients (268) |
| ILPD (Indian Liver Patient Dataset) | 583 | 10 | Class 1: liver patients (416); Class 2: non-liver patients (167) |

We constructed the training data and the test data for all the data sets that we used in this research. The training set contains 80% of the data while the test set contains the remaining 20%. The Stratified shuffle split technique available in scikit-learn (sklearn), a machine learning library for the Python programming language, was used since it preserves the percentage of samples for each class.

### 4.2 Evaluation Measures

To measure the performance of our model, several evaluation measures were used such as Sensitivity, Specificity, and Accuracy [5]. These were derived from the confusion matrix, and applied to the classifier evaluation, and are shown in Equation (1) through (3).

$$\mathit{Accuracy} = (TP + TN)/(TP + FP + TN + FN) \tag{1}$$

$$Sensitivity = TP/(TP + FN) \tag{2}$$

$$Specificity = TN/(TN + FP) \tag{3}$$

where:

$TP$ = number of positive examples correctly classified
$TN$ = number of negative samples correctly classified
$FN$ = number of positive observations incorrectly classified
$FP$ = number of negative samples incorrectly classified

In addition, the Area under the Receiver Operating Characteristic curve (ROC) were also measured [5]. This is because almost all data sets used in this paper can be considered as imbalanced data sets. This metric has been widely used as the standard measure for comparison of the performance. The ROC curve is a representation of the best decision boundaries for the cost between the True Positive Rate (TPR), and the False Positive Rate (FPR) that are defined in Equation (4) and (5). The ROC curve plots TPR against FPR.

$$TPR = TP/(TP + FN) \tag{4}$$

$$FPR = FP/(FP + TN) \tag{5}$$

The area below the ROC curve is called AUC and is widely utilized for weighing classifier performance. The value of AUC ranges from 0.0 to 1.0, where a value of AUC equals 1.0 means perfect prediction, a value of 0.5 means random prediction, and a value less than 0.5 is considered as a poor prediction.

### 4.3 Results

In this paper, we compared the performance of our method with the individual base learners used in this research, baseline ensemble, and best results available so far in the literature. We applied the stacked ensemble or super learner (SL) methods on the training data. For the evaluation of the model, we used the test data set. Table 3 shows the performance (different evaluation metrics) of the proposed technique (SL having two base learners - GBM and RF) on the test data for the different data sets while Table 4 shows the performance of SL having three base learners namely GBM, RF, and DNN on test data for all the data sets used in this research.

Table 3: Performance of the proposed techniques on test data (SL consisting of two base learners - GBM and RF).

| **Data sets** | **Sensitivity (%)** | **Specificity (%)** | **Accuracy (%)** | **AUC** |
|---|---|---|---|---|
| **Messidor** | 90.24 | 45.37 | 69.26 | 0.806 |
| **WBC** | 100.00 | 97.83 | 98.57 | 0.997 |
| **PIDD** | 90.74 | 76.00 | 81.17 | 0.882 |
| **ILPD** | 94.12 | 51.81 | 64.10 | 0.733 |

Table 4: Performance of the proposed techniques on test data (SL consisting of three base learners - GBM, RF and DNN).

| **Data sets** | **Sensitivity (%)** | **Specificity (%)** | **Accuracy (%)** | **AUC** |
|---|---|---|---|---|
| **Messidor** | 78.86 | 79.63 | 79.22 | 0.847 |
| **WBC** | 100.00 | 98.91 | 99.29 | 0.998 |
| **PIDD** | 96.30 | 73.00 | 81.17 | 0.886 |
| **ILPD** | 70.59 | 72.29 | 71.80 | 0.730 |

Comparing Table 3 and Table 4, for all the data sets used in this research, best results (based on test data) were obtained using the super learner methods (either SL consisting of two base learners or SL consisting of three base learners). For the Messidor data, best AUC, specificity, and accuracy were obtained when SL consisting of three base learners applied on the test data and for sensitivity best results were reported using SL with two base learners. Interestingly, for WBC the best performance was obtained when SL consisting of three base learners applied on the test data for all the performance measures considered in this research. For the PIDD data set, best AUC, sensitivity and accuracy were obtained when SL consisting of three base learners applied on the test data and for specificity best results were reported SL with two base learners. For the ILPD, best AUC and sensitivity were obtained with SL consisting of three base learners and for accuracy and specificity best results were achieved SL consisting of two base learners.

In addition, the accuracy comparison using single base learners, baseline ensemble, the SL consisting of two base learners (GBM and RF), and the SL having three base learners (GBM, RF, and DNN) on test data are presented in Table 5. We also compare AUC using single base learners, baseline ensemble, SL consisting of two base learners (GBM and RF), and the SL that consist of three base learners (GBM, RF, and DNN) on the test data set are shown in Table 6.

Table 5: Accuracy comparison using single base learners, baseline ensemble, and super learner consisting of two base learners and three base learners on test data (**Bold** indicates the best value).

| **Data set** | **Accuracy (%) (GBM)** | **Accuracy (%) (RF)** | **Accuracy (%) (DNN)** | **Accuracy (%) (Baseline ensemble)** | **Accuracy (%) (SL with 2 base learners)** | **Accuracy (%) (SL with 3 base learners)** |
|---|---|---|---|---|---|---|
| **Messidor** | 71.86 | 67.53 | 77.92 | 69.86 | 69.26 | **79.22** |
| **WBC** | **99.29** | 98.57 | **99.29** | 97.90 | 98.57 | **99.29** |
| **PIDD** | 79.22 | **81.17** | 74.68 | 75.33 | **81.17** | **81.17** |
| **ILPD** | 63.32 | 64.10 | 65.81 | 70.16 | 64.10 | **71.80** |

Table 6: AUC comparison using single base learners, baseline ensemble, and super learner of having two base learners and three base learners (**Bold** indicates the best value).

| **Data set** | **AUC (GBM)** | **AUC (RF)** | **AUC (DNN)** | **AUC (Baseline ensemble)** | **AUC (SL with 2 base learners)** | **AUC (SL with 3 base learners)** |
|---|---|---|---|---|---|---|
| **Messidor** | 0.815 | 0.765 | 0.838 | 0.740 | 0.806 | **0.847** |
| **WBC** | 0.997 | 0.997 | **0.998** | 0.996 | **0.998** | **0.998** |
| **PIDD** | 0.876 | 0.882 | 0.872 | 0.808 | 0.882 | **0.886** |
| **ILPD** | 0.718 | 0.727 | 0.733 | 0.730 | 0.727 | **0.734** |

From the Table 5, it is explicit that our proposed method SL having three base learners performs slightly better (or equal in few cases) than other methods for all the data sets used in this research. For the Messidor data set, SL

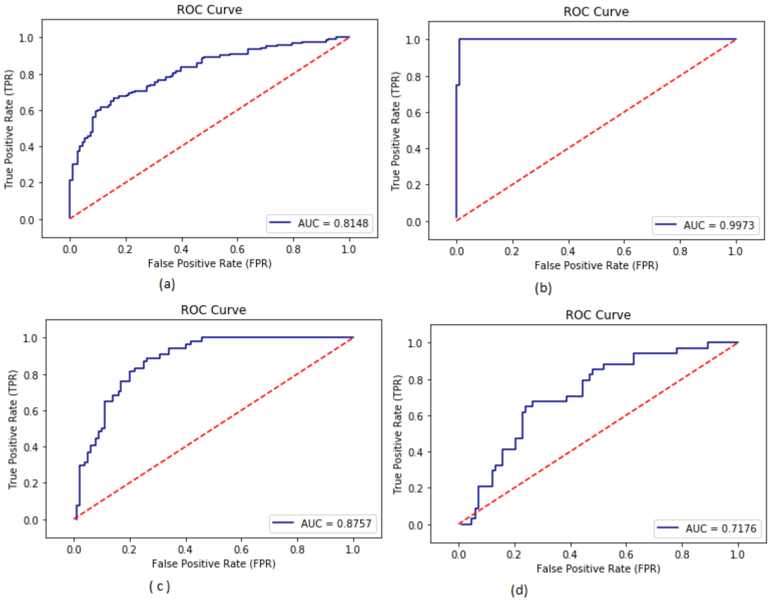

Fig. 5: ROC analysis using GBM for different data sets used: (a) Messidor / Diabetic Retinopathy (DR), (b) Wisconsin Breast cancer Diagnostics, (c) Pima Indian diabetes, and (d) ILPD (Indian Liver Patient data set).

with three base learners has the best accuracy (79.22%) followed by the individual learner DNN (77.92%). For PIDD, the best accuracy (81.17%) was obtained with both SL methods (having two and three base learners) and with an individual learner named RF. For ILPD, the best accuracy (71.80%) was obtained when the SL method with three base learners was applied on the test data followed by the baseline ensemble (70.16%).

Similar trends are also observed in Table 6, the best AUC value was obtained using the super learner having three base learners for all the data sets used in this research. For the Messidor data set, the best AUC value (0.847) was reported with SL consisting of three base learners followed by individual base learner DNN (0.838). For WBC, the best AUC score (0.998) was reported with both SL methods (using two and three base learners) and an individual learner named DNN. For PIDD, the best AUC (0.886) was attained with the SL method consisting of three base learners followed by SL with two base learners and an individual learner RF (0.882). For ILPD, the best AUC (0.734) was obtained with the SL method having three base learners followed by an individual learner named DNN (0.733).

We also present the ROC analysis for all data sets that have been used in this paper using all the base learners and the super learner. The ROC plots using the base learners namely GBM, RF, and DNN for all data sets (test) are shown in Figure 5, Figure 6, and Figure 7, whereas the ROC plots using the super learner or stacked-ensemble for the data sets are shown in Figure 8.

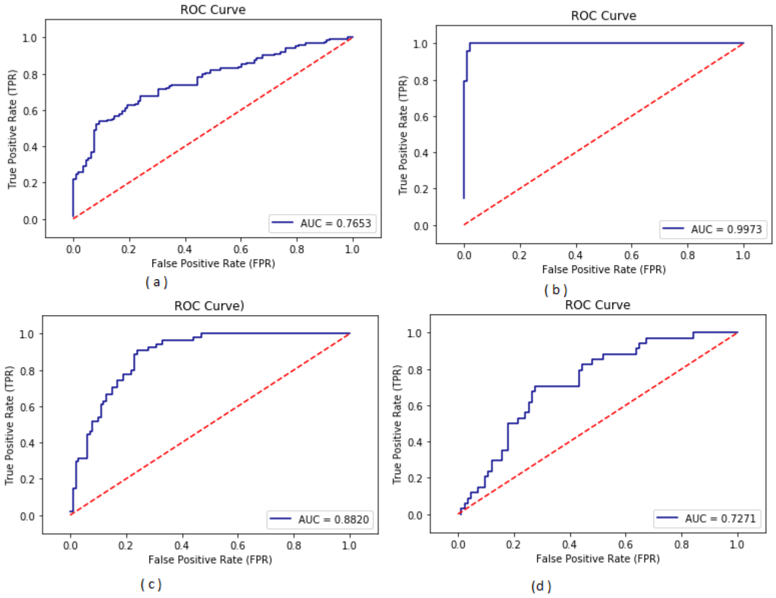

Fig. 6: ROC analysis using RF for different data sets used: (a) Messidor / Diabetic Retinopathy (DR), (b) Wisconsin Breast cancer Diagnostics, (c) Pima Indian diabetes, and (d) ILPD (Indian Liver Patient data set).

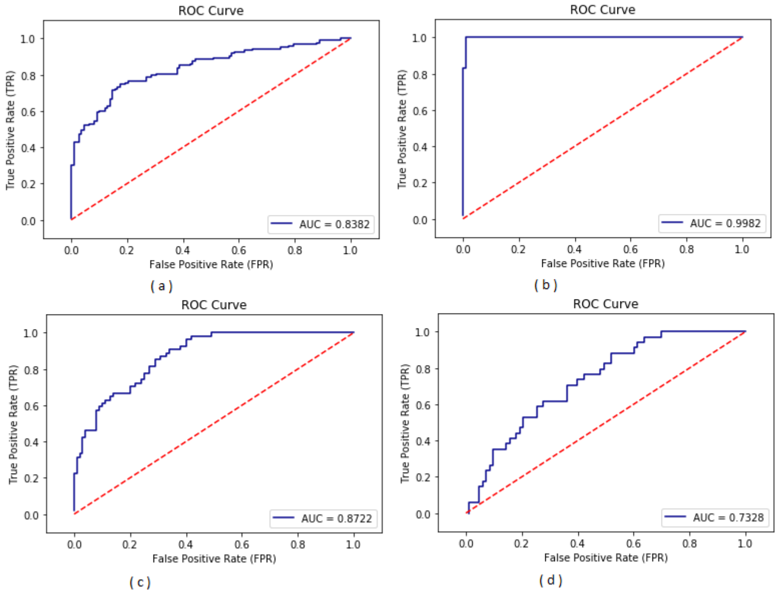

Fig. 7: ROC analysis using DNN for different data sets used: (a) Messidor / Diabetic Retinopathy (DR), (b) Wisconsin Breast cancer Diagnostics, (c) Pima Indian diabetes, and (d) ILPD (Indian Liver Patient data set).

Table 7: Comparison of super learner (SL) methods, and state-of-the-art (SA) best results for the four benchmark data sets (*Italics* indicates that result is obtained using the SL methods having two base learners).

| **Data set** | **Sensitivity (%)** | | **Specificity (%)** | | **Accuracy (%)** | | **AUC** | |
|---|---|---|---|---|---|---|---|---|
| | **SA** | **SL** | **SA** | **SL** | **SA** | **SL** | **SA** | **SL** |
| **Messidor** | 90.00 | *90.24* | 91.00 | 79.63 | 90.00 | 79.22 | 0.99 | 0.847 |
| **WBC** | - | 100.00 | - | 98.91 | 97.57 | 99.29 | - | 0.998 |
| **PIDD** | - | 96.30 | - | 76.00 | 76.95 | 81.17 | 0.846 | 0.886 |
| **ILPD** | - | 94.12 | - | 72.29 | 79.66 | 71.80 | - | 0.733 |

### 4.4 Performance comparison of four benchmark data sets with other methods

Several ML techniques have been used for the four benchmark data sets that we used for the evaluation of the performance. Authors in [23] used an ensemble-based technique on the Messidor data set with 10-fold cross validation; they obtained 90% sensitivity, 91% specificity, 90% accuracy, and 0.989 AUC. Authors in [24] showed the comparison of five different classifiers based on 10-fold cross validation on the WBC data sets. Among these classifiers, the best accuracy (about 97%) was obtained by SMO. The authors also used feature selection method named Principal Component Analysis (PCA) on the WBC data set with the J48, an open source java implementation of the C4.5 decision tree algorithm and MLP classifiers, and the best accuracy achieved was 97.57%. In [7], authors compared the performance in terms of accuracy of bagging, and boosting with a hybrid approach of a Hierarchical and Progressive Combination of Classifiers (HPCC). They found a 83.34% accuracy for HPCC, and 82.39% for bagging with GLM. The authors did not explicitly mention the number of cross-validation they used in their experiments. In [25], the authors used GA for attribute or feature selection methods, and a NB classifier has been used for classification on PIDD. For PIDD, the authors partitioned the data set with a split of 70% / 30% for training and testing, respectively. They obtained an accuracy of 77.3%, and 76.95% for training and testing, and an AUC of 0.816 and 0.846, respectively. For the ILPD data set, the best accuracy (79.38%) was found using an ensemble classifier with 5 fold cross-validation [26]. In [27], the authors provided a comparative analysis of different ML algorithms for the diagnosis of different data sets. For ILPD, the best accuracy (79.66%) was obtained by SVM.

We summarized and compared the results that we obtained using the SL methods with the state-of-the-art (SA) best results based on the four benchmark data sets outlined in Table 7. From the table, for the SL methods all the values were obtained using three base learners except the sensitivity for the Messidor data (indicates as *italics*), which were achieved using two base learners. It is important to note that in our experiments, we used 80% for training and 20% for testing for all data sets used and the results were evaluated on the test data.

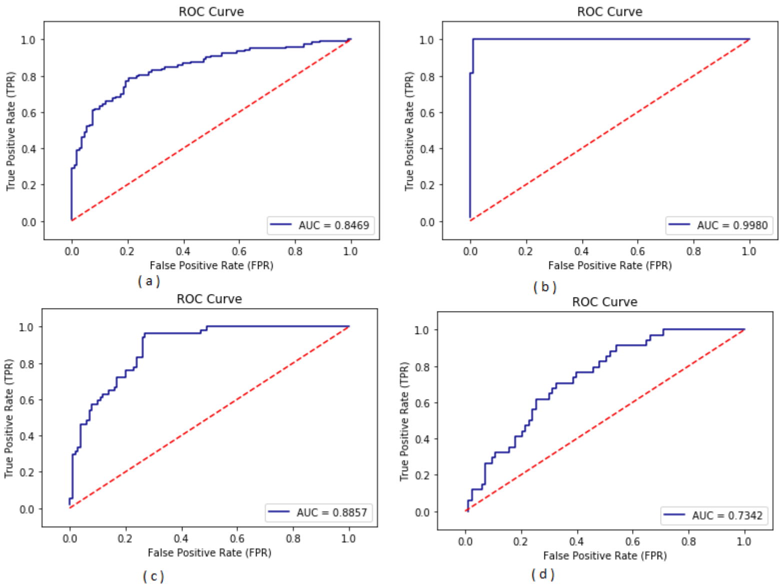

Fig. 8: ROC analysis using the super learner (using three base learners) for different data sets used: (a) Messidor / Diabetic Retinopathy (DR), (b) Wisconsin Breast cancer Diagnostics, (c) Pima Indian diabetes, and (d) ILPD data sets.

## 5 Conclusions

Classification is one of the important tasks of machine learning that predicts the target class for each example in the data. To achieve good performance on the available data sets, researchers are using appropriate single classifiers. However, selecting the best data mining or machine learning model for a specific problem is complex. Due to this researchers are using multiple different models for a particular problem to obtain good performance. In this paper, we focused on the improvement of the classification performance in terms of sensitivity, specificity, accuracy, and AUC for four benchmark data sets related to healthcare. To do so, we used the super learning or stacked-ensemble method that finds the optimal weighted average of diverse learning models. For the base learners we first used GBM and RF and then used another base learner DNN along with the previous two - GBM and RF. To find the optimal combination of the base learner models used in this research, Generalized Linear Models (GLM) was used as the meta-learner.

From our experimental results, we showed that super learning has a better performance compared to individual base learners, baseline ensemble approach, and some of the state-of-the-art techniques for these four benchmark data sets. Using the stacked ensemble or super learner methods (using two base learners or three base learners), we achieved better or equal performance compared to the individual base learners and the baseline ensemble for all the evaluation metrics considered in this research.

In our future work, we plan to apply this technique to other health related big data problems. In addition, we will investigate research problems by including more diverse base learners and other meta-learner. Finally, this technique could be applied to other real world problem domains such as cyber security, Geographic Information System, transportation, and agriculture.

## References

1. Han, Jiawe, and Micheline Kamber. Data mining concepts and techniques San Francisco Moraga Kaufman. (2001).
2. Kourou, Konstantina, et al. Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal 13 (2015): 8-17.
3. Agrawal, Rakesh, et al. An interval classifier for database mining applications. Proc. of the VLDB Conference. 1992.
4. Rahman, SM Monzurur, Md Faisal Kabir, and Muhammad Mushfiqur Rahman. Integrated Data Mining and Business Intelligence. Encyclopedia of Business Analytics and Optimization. IGI Global, 2014. 1234-1253.
5. Fawcett, Tom. An introduction to ROC analysis. Pattern recognition letters 27.8 (2006): 861-874.
6. Casas, Pedro, et al. Big-DAMA: big data analytics for network traffic monitoring and analysis. Proceedings of the 2016 workshop on Fostering Latin-American Research in Data Communication Networks. ACM, 2016.
7. Kaur, Harnoor, and Shalini Batra. HPCC: An ensembled framework for the prediction of the onset of diabetes. Signal Processing, Computing and Control (ISPCC), 2017 4th International Conference on. IEEE, 2017.
8. Gibbons, Chris, et al. Supervised machine learning algorithms can classify open-text feedback of doctor performance with human-level accuracy. Journal of medical Internet research 19.3 (2017).
9. Silwattananusarn, Tipawan, Wanida Kanarkard, and Kulthida Tuamsuk. Enhanced classification accuracy for cardiotocogram data with ensemble feature selection and classifier ensemble. Journal of Computer and Communications 4.04 (2016): 20.
10. van der Laan, Mark J., Eric C Polley and Alan E. Hubbard. Super Learner Statistical Applications in Genetics and Molecular Biology, 6.1 (2007): -. Retrieved 19 Mar. 2018, from doi:10.2202/1544-6115.1309.
11. Van der Laan, Mark J., and Sherri Rose. Targeted learning: causal inference for observational and experimental data. Springer Science & Business Media, 2011.
12. Vanerio, Juan, and Pedro Casas. Ensemble-learning approaches for network security and anomaly detection. Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks. ACM, 2017.
13. Aiello, Spencer, et al. Machine Learning with Python and H20. H2O. ai Inc (2016).
14. Cirean, Dan, Ueli Meier, and Jrgen Schmidhuber. Multi-column deep neural networks for image classification. arXiv preprint arXiv:1202.2745 (2012).
15. Nykodym, Tomas, et al. Generalized Linear Modeling with H2O. Published by H2O. ai, Inc (2016).
16. LeDell, Erin. Scalable Super Learning. Handbook of Big Data 339 (2016).
17. LeDell, Erin E. Scalable Ensemble Learning and Computationally Efficient Variance Estimation. University of California, Berkeley, 2015.
18. Wolpert, David H. Stacked generalization. Neural networks 5.2 (1992): 241-259.
19. Breiman, Leo. Stacked regressions. Machine learning 24.1 (1996): 49-64.
20. LeBlanc, Michael, and Robert Tibshirani. Combining estimates in regression and classification. Journal of the American Statistical Association 91.436 (1996): 1641-1650.
21. van der Laan, Mark J., Sandrine Dudoit, and Aad W. van der Vaart. The cross-validated adaptive epsilon-net estimator. Statistics & Decisions 24.3 (2006): 373-395.
22. Casas, Pedro, and Juan Vanerio. Super learning for anomaly detection in cellular networks. Wireless and Mobile Computing, Networking and Communications (WiMob),. IEEE, 2017.

23. Baak, Valerio, and Edward H. Kennedy. Principled Machine Learning Using the Super Learner: An Application to Predicting Prison Violence. Sociological Methods & Research (2018): 0049124117747301.
24. Antal, Blint, and Andrs Hajdu. An ensemble-based system for automatic screening of diabetic retinopathy. Knowledge-based systems 60 (2014): 20-27.
25. Salama, Gouda I., M. Abdelhalim, and Magdy Abd-elghany Zeid. Breast cancer diagnosis on three different datasets using multi-classifiers. Breast Cancer (WDBC) 32.569 (2012): 2.
26. Choubey, Dilip Kumar, et al. Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection. Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing System (ICCCS 2016). 2017.
27. Abdar, Moloud, et al. Performance analysis of classification algorithms on early detection of liver disease. Expert Systems with Applications 67 (2017): 239-251.
28. Fatima, Meherwar, and Maruf Pasha. Survey of Machine Learning Algorithms for Disease Diagnostic. Journal of Intelligent Learning Systems and Applications 9.01 (2017): 1-16.
29. Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
30. Smith, Jack W., et al. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. Proceedings of the Annual Symposium on Computer Application in Medical Care. American Medical Informatics Association, 1988.