International Journal of Artificial Intelligence and Machine Learning
Volume 11 • Issue 2 • July-December 2021

Three-Layer Stacked Generalization Architecture With Simulated Annealing for Optimum Results in Data Mining

K. T. Sanvitha Kasthuriarachchi, University of Kelaniya, Kelaniya, Sri Lanka
Sidath R. Liyanage, Univeristy of Kelaniya, Kelaniya, Sri Lanka

ABSTRACT

The combination of different machine learning models to a single prediction model usually improves the performance of the data analysis. Stacking ensembles are one of such approaches to build a high-performance classifier that can be applied to various contexts of data mining. This study proposes an enhanced stacking ensemble by collating a few machine learning algorithms with two-layered meta classifications to address the limitations of existing stacking architecture to utilize simulated annealing algorithm to optimize the classifier configuration in order to reach the best prediction accuracy. The proposed method significantly outperformed three general stacking ensembles of two layers that have been executed using the meta classifiers utilized in the proposed architecture. These assessments have been statistically proven at a 95% confidence level. The novel stacking ensemble has also outperformed the existing ensembles named Adaboost algorithm, gradient boosting algorithm, XGBoost classifier, and bagging classifiers as well.

KEYWORDS

Classifier, Ensemble, Hyperparameter, Simulated Annealing, Stacked Generalization

1. INTRODUCTION

Research trends in Machine Learning include investigations on the most promising algorithm for a given data set. Most prediction tasks can be implemented using diverse set of algorithms. These can be arranged based on their prescient tasks. Decision Tree algorithm, Random Forest algorithm, Naïve Bayes analyzer, Artificial Neural Network, Linear Regression, Logistic Regression, Support Vector algorithm and K-Nearest Neighbor algorithm are a few of them to perform classification, clustering, regression, association rule mining etc.. A substantial research effort has been exerted on these algorithms to make better decisions related to the choice of algorithms (Li Congcong et. al, 2013).

In practice, researchers would analyze the presentation of the chosen algorithms on a test data set and select the algorithm that actually outperforms the others in a significant manner (P.K. Douglas et. al, 2016 and Ladds et. al, 2017). However, there is still the inherent uncertainty of whether a chosen algorithm will be the most suitable for all real- world datasets. As expressed in the "No free lunch theorem" the computational expense of finding an answer, arrived at the midpoint of overall issues in the class, is the equivalent for any arrangement strategy (Wolpert David & Macready William, 1996). Classifier combination strategies such as, boosting and bagging have outperformed solitary best classifiers on many real-world datasets (Syarif, Iwan et. al, 2012). Hence, when none of the

DOI: 10.4018/IJAIML.20210701.oa10

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

classification algorithms fundamentally beats different techniques, it is pragmatic to choose a couple of algorithms and to decide the best during runtime (Dietterich T.G, 2000).

From a mathematical perspective, a classification algorithm is a sophisticated fit to a non-linear function, and a solitary machine learning model may fit well to a certain dataset. However it may overfit or underfit to some different datasets. Thus, the prediction accuracy of a solitary model may arrive at the upper limit even with ideal parameters. One potential technique to overcome the limitation of a single algorithm is to join a few algorithms to break through the upper limit of a single learning algorithm which is called as an ensemble. Bagging, Boosting and Stacking are three types of ensembles. Stacking/ Stacked generalization is an ensemble strategy that utilizes a higher-level model to join lower-level sub-models to accomplish higher prediction accuracy. Unlike bagging and boosting approaches that consolidate classifiers of a similar kind, the stacked generalization can join diverse algorithms through a meta- learning model to expand the accuracy (Ting, K. M, 1999). It is an ensemble learning approach where the ensemble model could yield superior predictive performance than any of the constituent lower-level sub-models.

Stack generalization is of two types; named, stacking regression and stacking classification. Stacking regression is consolidating various regression models through meta-regressor. The stacking classification, collates individual classification models and the meta-classifier is fitting dependent on the outcome of individual classification models in the ensemble (Y. Ren et. al, 2016). There have been numerous studies on stacking ensembles that show that their accuracies are higher than the individual algorithms in the prediction of prescient tasks (Ladds et. al., 2017). However, there exists few limitations of this predominant methodology as well. The base learner parameters must be tweaked intensely since they affect the accuracy of the final prediction model. Stack generalization follows a "black box" algorithm, so the specific commitment of each covariate to the prediction cannot be quantified (Naimi AI and Balzer LB, 2018). The prediction accuracy of the stacking ensemble is still uncertain and enough exertion has not been made to build the robustness of the stacking ensemble (Kuncheva, L.I, 2014).

In this study, each of these deficiencies of the stack generalization is evaluated by implementing a priori-specified hyperparameterized stack ensemble machine learning approach. It consolidates several algorithms as base classifiers; namely, K-Nearest Neighbor (KNN), Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT) and Artificial Neural Network (ANN) into a single predictive model and returns the best prediction through the proposed approach (Brown G, 2017). It consists of three (3) layers in the ensemble. The base learner models are chosen through the best cross-validated Log Loss Error (CLLE). The individual modeling techniques of approach-based sensitivity estimation were utilized. Hyperparameters, which influence the whole ensemble's performance structure and intricacy, are optimized (Wong Jenna et. al, 2019). Two (2) layers of Meta learners are proposed to fit the outcomes of their previous layers. The proposed stacking ensemble has been tested on fifteen datasets to demonstrate the evidence of its accuracy. It can be asserted that this novel approach is an optimal classifier that infers a high performing predictive ensemble.

The rest of this paper is arranged as follows. In section 2, the background study is presented including a discussion about the stacked generalization with the approaches followed by previous studies and their deficiencies. In section 3, the methodology followed in proposing the novel stacking ensemble is presented. In section 4, the evaluation procedure and the results are discussed. In section 5, a comprehensive discussion is presented. The section 6 presents research contribution and the implications. Finally, a conclusion and future work is presented in section 6.

2. BACKGROUND

2.1 Classifier Combination for Ensembles

It is impractical most of the time to identify apriori the most suitable algorithms for regression or classification problems in data mining. This leads the data analyst to use many different algorithms to develop different models and to evaluate their performances. Once the evaluation is performed, which is often under different configurations, the best out of the all models is selected to predict the target attributes and to make decisions (Sanvitha Kasthuriarachchi et. al, 2018). A single algorithm may be unable to capture the complete underlying structure of the data to derive optimal predictions. This is where the integration of multiple models gathered into a single meta – model has been found to be effective (Vamathevan, J et. al, 2019). The main intuition behind the concept of assembling is to address the point which "Why all the prediction models are not considered and select the best model out of all for the machine learning problem".

Ensembles are of three types named Bagging, Boosting and Stacking. Bagging and boosting are two of the common ensemble techniques used in machine learning (Re Matteo & Valentini Giorgio, 2012). Bagging generates multiple versions of predictors and form an aggregated predictor by voting each version and getting the average of them (Breiman L, 1996). Bagging meta-estimator and the random forest are considered as algorithms follow bagging approach. Boosting works in a similar way to bagging by combining several poor performing base learners in an adaptive way. Experimental work showed that bagging is effective for data sets with noisy values (T.G. Dietterich, 2000). In boosting, the learning algorithms are given different distribution or weighting according to the errors of the base learners (Brown G., 2017) and (Pedregosa F et. al, 2011). AdaBoost, Gradient Boosting (GBM), eXtream Gradient Boosting (XGBM), Light GBM, and CatBoost are considered as Boosting techniques.

The third approach is stacking. It takes the output of selected classifiers on the training data and applies another learning algorithm on them to predict the response values (Large, J et. al, 2019). Usually, the stacked generalization architecture follows two layers. First, the base classification in layer 1, which uses base classifiers to construct the ensemble by training the dataset. It generates the input to the second layer. Second, the meta classification in layer 2, which combines the results of the outcome of layer 1 using a meta- classifier to produce the final predictive model. Stacking combines multiple learning algorithms L1, L2, …, LN on a single dataset S. In the layer 1, set of base classifiers $C1, C2, \ldots, Cn$ are generated where Ci = Li(S). Base learner classifiers can be any of the machine learning algorithm such as, KNN, RF, NB, SVM, ANN and DT. In the second layer, a meta-level classiðer combines the outputs of the base-level classiðers. Depend upon the prediction task, when the prediction complies with classification Logistic Regression algorithm (LR) and when regression is performed, the linear regression is used. In this layer, no learning takes place at the meta-level when combining classiðers by a voting mechanism. The voting scheme remains the same for all different training sets and sets of learning algorithms (or base-level classiðers). The simplest voting scheme is the plurality vote. According to this voting scheme, each base-level classiðer casts a vote for its prediction. The classifiers who achieved more votes are added to the meta layer.

The stacking concept was first introduced by researchers in a biological study (Yang et. al, 2010). This is an application of stacked generalization to k- fold cross validation since all the analysis models use the same k-fold splits of the data and a meta- model fits into the out-of-fold predictions of each of the models. The traditional machine learning approaches build a single hypothesis based on the training data but, the ensemble approach attempts to develop a set of hypotheses and combine them to form a new hypothesis (Sherri Rose, 2013). Different studies have been carried out based on the stacking concept. Once multiple prediction models are combined, more information could be captured in the fundamental structure of the data (Clarke B, 2003). A researcher has highlighted the importance of recognizing the uncertainty when selecting models, and the prospective role that assembling can play when combining several models to create one that outperforms single models (Varian, Hal

R, 2014). In a study about improving accuracy and reducing variance of behavior classification in accelerometer done by a researcher has shown that stacked ensembles can be easily adapted to any type of industry to achieve better accuracy in the predicted model (Ladds et. al., 2017). Also they emphasized the importance of the human intervention and the computation time required to execute the stacking ensemble for the machine learning tasks. The ensemble learning performs better than the individual algorithms (Džeroski S. and Ženko B, 2004) and (Romesburg, H.C, 2014). In another study, a researcher has pointed out that the high computational time and the memory requirement for the smooth execution of stacking approach is significant and thereby, the stacking ensemble can be potentially flawed too (Sherri Rose, 2013).

A study has proposed a Deep belief network that is a learning model to represent unknown data efficiently. They utilized Adaptive Sparse Restricted Boltzmann machines (AS-RBM) and partial least square (PLS) regression fine-tuning to increase the accuracy and the robustness of the learning model. The researchers have tested their model on Mackey-Glass time-series prediction, 2-D function approximation, and unknown system identification and obtained a better accuracy in faster learning speed (Wang G. et. al, 2019). In another study, the researchers have claimed that the model they with Sparse Deep Belief Network and Fuzzy Neural Network (SDBFNN) achieved superior performance in terms of robustness and accuracy (Wang G. et. al, 2019). The researchers who have proposed a wind power prediction using deep neural network base ensemble and transfer learning have shown better robust modeling results. They utilized deep auto-encoders as the base-regressors and the Deep Belief Network as the meta-regressor (Qureshi A.S, 2017)

2.2 Hyperparameter Optimization of Classifiers

Hyperparameter optimization is the process of identifying the best parameter values if the classifiers which derives the ideal prediction model. This is known as hyperparameter tuning as well. There are diverse hyperparameter optimization methods, namely; (1) Grid search (2) Random search (3) Bayesian optimization (4) Simulated Annealing algorithm (5) Genetic algorithm (6) Particle swarm optimization. Grid search, Random research and the Bayesian optimization are the frequent hyperparameter optimization techniques. The Grid search is the most basic method. The prediction model will be created for each possible combination of all the hyperparameter value and will evaluate each model and select the architecture which produces the best result. Random search finds better models by effectively searching a larger, less promising configuration space than grid search method (Dietterich T.G, 2000). The next method, Bayesian optimization is also called the surrogate method which keeps track of past evaluation results which are used to form a probabilistic model, maps the hyperparameters to a probability of a score on the objective function that it uses. It could find a better set of hyperparameters in less time because they study about the best set of hyperparameters to evaluate, based on past trials (James Bergstra and Y Bengio, 2012). Most of the issues identified in the above popular approaches can be overcome by using Simulated Annealing Algorithm. It finds the optimal solution in a discrete search space with many possible hyperparameter combinations. This is a probabilistic technique which is identified as a global optimum method. Genetic Algorithm is also a metaheuristic algorithm based on the evolutionary concept. It finds the individuals with the highest survival capability. One particular generation passes their capabilities to their next generation. Then the next generation inherits that feature from the parents and make better individuals. The worst individuals will gradually disappear. This concept will apply to the optimization of hyperparameters of classifiers. The population, chromosomes and genes will initialize to search space, hyperparameters and their values. The fitness value will calculate and measure the performance. The selection, crossover and mutation will apply on chromosomes to produce a new generation and measure the performance. These steps will repeatedly apply till optimal hyperparameters are derived (Yang Li, Abdallah Shami, 2020). Particle swarm optimization is another evolutionary approach for the optimization. The implementation of Particle swarm optimization is easier than Genetic algorithm. It works by allowing the group of particles to travel the search space in a semi random manner. The

optimal hyperparameters are identified by cooperating and sharing information among the individuals in the particle groups. However, it requires additional population initialization with more execution time and resources (Yang Li, Abdallah Shami, 2020).

Several studies have focused on the hyperparameter optimization in different forms (Wu J, et. al, 2016). There was a study to find a method to accelerate the search process by transferring information from previous trials to other datasets (Bardenet et. al, 2013). The key challenge they faced was the accuracy measurement. It was a relatively difficult task to maintain the accuracy of the model while maintaining the speed of the analysis through hyperparameter tuning (Yogatama D., 2014). One study has introduced a systematic framework to build ensembles with optimal weights for regression problems (Shahhosseini et. al, 2020). It was able to find the optimized ensemble weights that minimize both bias and variance of the predictions while tuning the hyperparameters of the base learners. A study about the use of Bayesian optimization to hyperparameter tuning in ensemble learning has been used as the optimized strategy to exploit trained models and improved ensembles to use as a classifier at the lower cost of regular hyperparameter optimization (Janez Demsar, 2006 and Julien-Charles, 2016). It could be observed that the existing ensemble techniques consider the base model construction and the weighted averaging to be independent steps and introduced a probabilistic ensemble weighting approach on cross-validation for hyperparameter optimization (Press, W et. al, 1992). The authors of another study have provided an extensive survey on a comparison of different hyperparameter optimization techniques (Yang Li, Abdallah Shami, 2020).

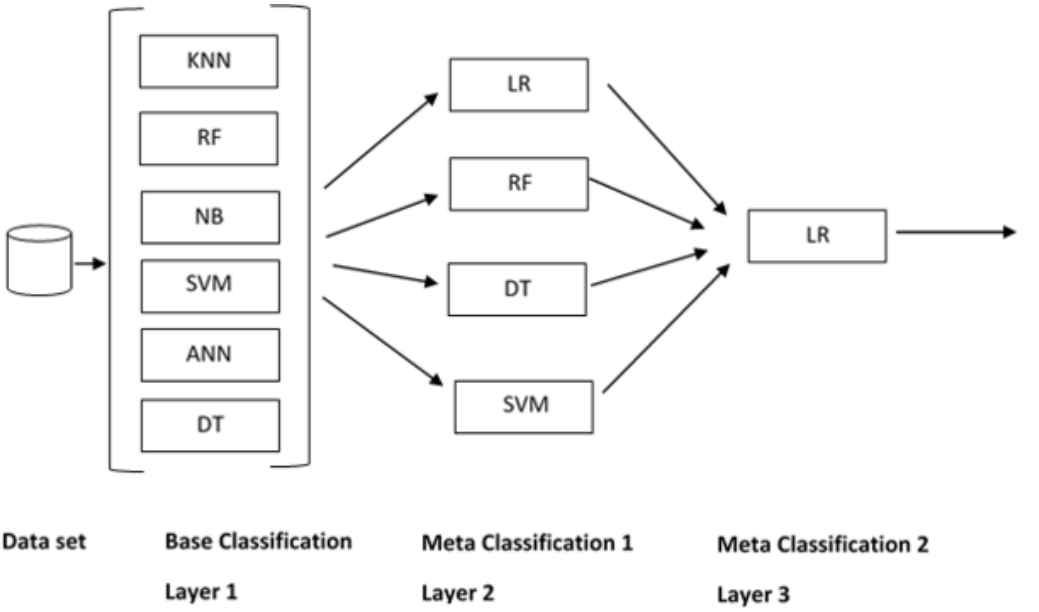
2.3 Hyperparameter Optimization Approach of Simulated Annealing Algorithm

The Simulated Annealing Algorithm is identified as a best approach for the hyperparameter optimization problem. It is a meta-heuristic optimization algorithm to take care of the optimization issue. It begins from an underlying arrangement, and afterward by a heterogeneous Markov chain moves to the neighbor arrangements until the best arrangement is found. In this chain, the progress probability from the current answer for the following arrangement relies upon an acknowledgment work. The progress is with probability 1 when the following arrangement is superior to the current arrangement. Else, it is finished with probability exp (ΔE/Θ), where ΔE is the contrast between estimations of acknowledgment work identified with the following and the current arrangement, and Θ is the temperature. The boundary Θ is balanced from significant level of degrees from the outset and it is diminished with a unique cooling (Daniel Delahaye, 2109) and (S. Kirkpatrick et. al, 1983). These changes are done until Θ gets to its most reduced temperature (S. Kirkpatrick et. al, 1983). The cooling schedule is denoted by equation (1),

$$k = (\log(T_{\min}) - \log(T)) \,/\, \log(\infty) \tag{1}$$

The Simulated Annealing Algorithm follows few steps in optimizing the hyperparameters of algorithms. They are; (1) the values for hyperparameters are selected randomly and consider it as the current state. (2) evaluate the model using the selected hyperparameters. (3) randomly update the value of a single hyperparameter of the current state and makes the neighboring state. This will be repeatedly apply till a new hyperparameter combination generates. (4) evaluate the performance of the neighboring state and accept it, if it's performance is higher than the current state. The Simulated Annealing Algorithm has demonstrated better performance in hyperparameter optimization (Press, W et. al, 1992 and Purushotham, Sanjay, 2017).

Figure 1. Architecture of the Proposed Ensemble



3. PROPOSED THREE LAYER STACKING ENSEMBLE

3.1 Architecture of the proposed Ensemble

In order to determine whether the stacking improves the accuracy of the prediction models, an extended version of two layer stacking ensemble has suggested. The proposed stacked generalization consists of three layers, named (1) layer 1: base classification, (2) layer 2: meta classification 1, and layer 3: meta classification 2. The proposed stacking classifier used six (6) base classifiers and all of them were trained using four (4) selected meta classifiers to obtain the layer 2 meta models. The four (4) meta models derived by each meta classifier have passed to next layer and produced the final prediction model using a single meta classifier. Architecture of the proposed ensemble is illustrated in figure 1.

The proposed extended stacking classifier uses KNN, RF, NB, SVM, ANN and DT algorithms for the layer 1- base classification. Since these algorithms are frequently using in many studies to derive prediction models, they have been selected as the base classifiers of the ensemble (Sanvitha Kasthuriarachchi et. al, 2018). The individual classifiers develop the prediction models with different accuracy levels. The output prediction models of layer 1 has been passed as the inputs of layer 2. The layer 2 meta- classifiers are logistic regression classifier (LR), RF classifier, DT classifier and SVM. The selection of meta- classifier should be possible to rely upon the prediction task and with proof of writing, the meta learners have chosen to construct the layer 2 output (Clarke, B., 2003). LR has utilized as the layer 3 meta classifier of this proposed procedure. The purpose of the selection of different algorithms is that they are following significantly different approaches for the model generation and focus on the data in different aspects to make a significant contribution to ensemble implementation. Different learning algorithms L_1, L_2, ..., L_N on a single dataset S, which consists of examples $s_k = (x_k, y_k)$, i.e., pairs of feature vectors (x_k) and their classiðcations (y_k). In the ðrst layer, the base classiðers C_1, C_2,...,C_N is generated, where $C_k = L_k(S)$. In the second layer, meta-level classiðers are learned to combines the outputs of base-level classiðers. To generate a training set for learning the meta-level classiðer, 10 fold cross validation procedure is applied (Romana Markovic,

2017). Third layer also works in the same manner as second layer which combines the meta model results into a single model.

3.2 Ranking the Classifiers

Since there are multiple base classifiers to form the layer 2 meta classification models, the specific commitment of each covariate to prediction is indistinct. This limitation is addressed by using a weighted scoring method of base classifier selection for the meta level classification of the ensemble. The scoring is done by assigning different weights to the base classifiers through the likelihood function known as the log loss. The Log loss is based on probabilities (Babalyan, K.,, 2018). Instead of maximizing the accuracy of the model, the error will be minimized by this technique. The lower the log loss, the higher the model accuracy. Therefore, the log loss is selected as the benchmark for comparing multiple prediction models. The log loss of each model were recorded and random weight esteem which is differing somewhere in the range of '0' and '1' is allocated to each base classifier while the prediction models are created. After an examination of the weights, the optimal base learners were chosen. The layer 2 meta models were trained and passed to the layer 3 meta classification layer to acquire the final prediction result. The proposed architecture uses to derive the best combination of base learners by measuring weighs through their log loss values and rate the models with weights. Then one aggregates the models of non-zero weights together to form the input to the meta- learner, which functions based on logistic regression. The log loss function of a machine learning model could be given as in equation (2);

$$LogLoss = -1 / n \sum_{i=1}^{n} [y_i \bullet \log_e(y_i) + (1 - y_i) \bullet \log_e(1 - y_i)] \quad (2)$$

Where, n is the number of instances in the dataset, y is the dependent variable in the dataset which will be either 0 or 1, y_i is the model probability of assigning label j to instance i. Based on the weights of each model, a random weight number is assigned to every model. Then the best classifiers are selected.

3.3 Optimizing the Hyperparameters of Classifiers

The hyperparameters of the classifiers are optimized to obtain better prediction results by the proposed ensemble. This would increases the accuracy and the robustness of the ensemble. Although there are plenty of stacking ensemble implementations, very few of them were able to illustrate a significant accuracy level than the individual machine learning algorithms (Yogatama, D. and Mann, G. , 2014). The existing stacking ensembles were having only one meta classifier. In case if classification problem is addresses LR became the meta classifier and for regression type Linear regression was used as the meta classifier. There were few studies that have looked beyond them and utilized other types of classifiers and regressors such as Support Vector (SV), Ridge Regression (RR), Multivariate Linear Regression (MLR) and so forth (Clarke B, 2003). The proposed extended stacking classifier has evaluated for many base classifiers and the best out of them were applied to meta model construction since they were ranked by the Log loss measurement. Therefore, the novelty of the proposed stacked generalization is the combination of hyperparameter tuning with weighted scoring for three layer stack with multi meta model optimization. The important hyperparameters are selected by evaluating the performance of the prediction models.

The important hyperparameters of the classifiers are determined to boost the prediction models through hyperparameter optimization. The KNN classifier has an important parameter, the number of nearest neighbors considered for each sample (*n_ neighbors*). If it is too small, the model will be underfitting, if the parameter is too large, the model will be overfitting. The RF forest has many important parameters, number of trees (*n_estimators*), maximum depth of the tree (*max_depth*). The deeper the tree, more splits and captures more information from the data, the criteria followed for

splitting (*criterion*), and minimum number of data points in a node before the node is split (*min_samples_split*). The only single hyperparameter of NB classifier that needs to tune is the smoothing parameter (alpha). SVM classifier has an important variable for penalty of the error term (*C*) in its objective function, Kernel type (*Kernel*) is another hyperparameter that has been considered as important in SVM. Maximum depth (*max_depth*) of the tree is the important hyperparameter in DT classifier. The higher the depth, more sub trees will be created with more accuracy. Finally, the ANN algorithm consists of type of the activation function (*activation*), number of epochs (*epochs*), number of hidden layers (*n_hidden_layers*), number of neurons in hidden layers (*neurons_per_layer*), the solver or the optimizer and the loss function (*loss*) as the important hyperparameters. Lastly, the LR classifier has two important hyperparameters, the penalty (*penalty*) and the coefficient (*C*). Penalty determines the regularization method used for the penalization and C determines the regularization strength of the model.

Next, the performance metrics and the evaluation methods are configured. Table 1 illustrates the configuration space for the hyperparameters of the classifiers. The 10 fold cross validation is performed to evaluate the hyperparameter optimization method. All the experiments are iterated ten times. Initially, the hyperparameter optimization is performed using Simulated Annealing Algorithm (SAA), Genetic algorithm (GA) and Particle Swamp Algorithm (PSA) to compare the performance.

Table 1. The configuration space for the hyperparameters of the classifiers.

Classifier	Hyperparameter	Search space
KNN	n_ neighbors	[10, 20]
RF	n_estimators max_depth criterian min_samples_split	[10, 100] [5, 50] [‘gini’, ‘entropy’] [2, 12]
NB	alpha	[-9, 0]
SVM	C Kernel	[0.5, 50] [‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’]
DT	max_depth	[2, 50]
ANN	activation epochs n_hidden_layers neurons_per_layer loss	[‘relu’,’tanh’] [20, 50] [2, 5] [16, 32] [‘binary_crossentropy’, ‘multiclass_crossentropy’]
LR	classifier__penalty C	[‘l1’, ‘l2’] [-4, 20]

The classifiers are evaluated using fifteen diverse datasets. The data gathered from questionnaires or surveys were recorded electronically and corrected for errors, noises, inconsistencies and outliers. Missing values in the online datasets were handled by median imputation and case deletion methods (Löw, F et. al, 2013 and Zhou G et. al, 2014). Four datasets are collected through surveys and questionnaires. The remaining datasets are chosen from publicly available data repositories. A detailed description about the datasets is available in the appendix section. A summary of the chosen datasets are depicted in table 2.

Table 2. A summary of the datasets used for the implementation of the proposed ensemble.

Dataset	Number of Features	Number of Instances	Number of Classes	probability of majority class	entropy of class probability distribution
LMSDataNew	11	799	2	89.26	49.2
ClassroomData	20	170	2	50.2	99.9
InClassSurveyData	20	171	2	51.3	99.9
InClassDBData	18	3795	2	65	93.49
xAPI-Edu-Data	14	481	2	69.57	88.6
BreastCancerData	9	286	2	63	95.06
ChronicKidneyDescies	25	400	2	54.55	99.4
WA_Fn-UseC_-Telco-Customer-Churn	21	7043	2	73	83.47
liver-disease-lab-data	11	483	2	72	86
DiabetesData	20	769	2	65	93.31
HeartData	14	303	2	84	62.85
Android_traffic	17	7846	2	57.18	98.5
FakeData	12	697	2	50	100
brain_tumorData	19	1449	2	88	53.69
hepatitisData	19	155	2	61.46	96.17

4. EXPERIMENTAL RESULTS AND DISCUSSION

Initially, the performance of selected classifiers were inspected using the datasets mentioned in the table 2. The datasets were segregated as 80:20 premise. This infers 80% of the data is in the training set and 20% of the data is in the testing set. Fifteen models were built based on the training sets by applying these base algorithms and the hold-out test sets were utilized to assess the model execution.

In the proposed approach, 10 fold cross validation is utilized to split the dataset into training and testing sets by reducing the overfitting. In this experiment, each dataset is separated to 10 equal parts and the first part is kept for testing purpose. The remaining 9 parts are used to train the model. While repeating this process 10 times, the testing dataset is kept on changing. This experiment is carried out by applying 3 fold cross validation and 5 fold cross validation as well. However, it could be noticed that the highest prediction accuracy of the model could be generated when k becomes 10 in 10 fold cross validation. This observation was common for the majority of the benchmark datasets. Therefore, the accuracies generated by 10 fold cross validation have been accepted.

4.1 Evaluating the Performance of Individual Classifiers

The performance of selected classifiers was measured using the chosen datasets as illustrated in table 3.

Some classifiers have indicated higher accuracy levels contrasted with the others. It can be clearly seen that the datasets named*; InClassSurveyData, ClassroomData, xAPI-Edu-Data*, and *hepatitisData* have involved in making lower predictive performance than the others. Be that as it may, a chosen algorithm outperformed the others, there might be an extremely little variety of the prediction accuracy of the rejected algorithms. Rejection of an algorithm for a prediction task dependent on a little variety of the exactness would not be a superior way to deal with training. In

Table 3. Prediction Performance of chosen individual Classifiers.

Dataset		RF	KNN	NB	SVM	DT	ANN
InClassSurveyData	**Accuracy**	**0.65 (+/- 0.1)**	**0.63 (+/- 0.12)**	**0.40 (+/-0.15)**	**0.65 (+/-0.03)**	**0.68 (+/-0.11)**	**0.65 (+/-0.03)**
	Precision	0.64	0.6	0.35	0.65	0.66	0.69
	Recall	0.71	0.66	0.12	0.62	0.69	0.64
InClassDBData	Accuracy	0.85 (+/- 0.04)	0.62 (+/-0.11)	0.75 (+/-0.03)	0.72 (+/-0.03)	0.80 (+/-0.0)	0.75 (+/-0.12)
	Precision	0.74	0.62	0.73	0.65	0.73	0.69
	Recall	0.81	0.54	0.74	0.72	0.69	0.93
LMSDataNew	Accuracy	0.81 (+/- 0.16)	0.75 (+/-0.14)	0.82 (+/-0.1)	0.79 (+/-0.09)	0.78 (+/-0.17)	0.81 (+/-0.0)
	Precision	0.78	0.76	0.81	0.72	0.72	0.80
	Recall	0.78	0.78	0.79	0.69	0.76	0.77
ClassroomData	Accuracy	0.70 (+/- 0.13)	0.60 (+/- 0.12)	0.62 (+/-0.16)	0.68 (+/-0.13)	0.65 (+/- 0.1)	0.63 (+/-0.16)
	Precision	0.77	0.76	0.73	0.65	0.77	0.69
	Recall	0.82	0.82	0.74	0.69	0.69	0.89
xAPI-Edu-Data	Accuracy	0.69 (+/- 0.06)	0.61 (+/-0.06)	0.59 (+/-0.07)	0.61 (0.04)	0.65 (+/-0.05)	0.53 (+/-0.01)
	Precision	0.75	0.62	0.56	0.59	0.62	0.62
	Recall	0.68	0.54	0.57	0.51	0.64	0.40
BreastCancerData	Accuracy	0.91 (+/- 0.02)	0.86 (+/-0.04)	0.90 (+/-0.03)	0.89 (+/-0.03)	0.9 (+/- 0.03)	0.63 (+/-0.01)
	Precision	0.93	0.87	0.89	0.87	0.92	0.88
	Recall	0.95	0.95	0.95	0.97	0.92	0.95
ChronicKidneyDescies	Accuracy	0.99 (+/- 0.02)	0.80 (+/-0.04)	0.97 (+/-0.02)	0.68 (+/-0.07)	0.96 (+/-0.04)	0.57 (+/-0.06)
	Precision	0.99	0.93	0.97	0.84	0.97	0.61
	Recall	0.98	0.60	0.97	0.60	0.95	0.51
WA_Fn-UseC_-Telco-Customer-Churn	Accuracy	0.79 (+/- 0.01)	0.71 (+/-0.01)	0.77 (+/-0.02)	0.73 (+/-0.0)	0.76 (+/-0.01)	0.73 (+/-0.0)
	Precision	0.63	0.58	0.54	0.72	0.68	0.69
	Recall	0.48	0.45	0.73	0.69	0.49	0.62
Liver-Disease-Lab-Data	Accuracy	0.70(+/-0.06)	0.67 (+/-0.06)	0.59 (+/-0.07)	0.72 (+/-0.01)	0.61(+/-0.07)	0.72(+/-0.01)
	Precision	0.74	0.75	0.58	0.75	0.75	0.75
	Recall	0.87	0.74	0.44	0.84	0.73	0.83
Diabetes Data	Accuracy	0.76 (+/- 0.05)	0.67 (+/-0.03)	0.75 (+/-0.06)	0.76 (+/-0.05)	0.71 (+/-0.02)	0.65 (+/-0.03)
	Precision	0.69	0.61	0.66	0.75	0.55	0.61
	Recall	0.54	0.51	0.57	0.47	0.56	0.44
HeartData	Accuracy	0.81 (+/- 0.07)	0.61 (+/-0.08)	0.83 (+/-0.06)	0.66 (+/-0.07)	0.78 (+/-0.06)	0.46 (+/-0.01)
	Precision	0.80	0.68	0.83	0.65	0.81	0.59
	Recall	0.84	0.72	0.87	0.81	0.76	0.56

continued on next page

Table 3. Continued

Dataset		RF	KNN	NB	SVM	DT	ANN
Android_Traffic	Accuracy	0.91 (+/- 0.01)	0.84 (+/-0.01)	0.43 (+/-0.01)	0.60 (+/-0.0)	0.81 (+/-0.02)	0.40 (+/-0.0)
	Precision	0.91	0.86	0.75	0.60	0.90	0.69
	Recall	0.93	0.86	0.06	0.99	0.89	0.63
FakeData	Accuracy	0.93 (+/- 0.04)	0.86 (+/-0.04)	0.68 (+/-0.05)	0.53 (+/-0.02)	0.90 (+/-0.04)	0.50 (+/- 0.0)
	Precision	0.94	0.88	0.61	0.51	0.88	0.61
	Recall	0.93	0.91	0.97	0.48	0.89	0.54
Brain_TumorData	Accuracy	0.94 (+/- 0.02)	0.92(+/-0.02)	0.89 (+/-0.02)	0.88(+/-0.0)	0.93(+/-0.01)	0.92 (+/-0.02)
	Precision	0.96	0.94	0.95	0.93	0.96	0.94
	Recall	0.98	0.92	0.92	0.96	0.96	0.98
HepatitisData	Accuracy	0.67 (+/- 0.09)	0.62 (+/-0.12)	0.75 (+/-0.17)	0.62 (+/-0.14)	0.54 (+/-0.18)	0.44 (+/-0.03)
	Precision	0.63	0.63	0.68	0.63	0.61	0.63
	Recall	0.68	0.68	0.86	0.82	0.53	0.75

this way, a combined model methodology is proposed to actualize for making forecasts as opposed to bounding to a single algorithm.

4.2 Evaluating the Proposed Ensemble

The hyperparameter optimization is performed in three approaches, as mentioned in section 3. The SAA, GA and PSO are utilized to decide the best approach which returns the highest performance. Table 4 illustrates the performance of the proposed ensemble under each approach. SAA approach has resulted in the best accuracy. Therefore, SAA was selected as the optimization approach for the proposed ensemble. The proposed stacking ensemble was implemented in Python language with the *sklearn* package and the *simulated_annealing.optimize* package (Jones E et. al, 2001) and (Pedregosa, F et. al, 2011).

Table 5 illustrates the prediction performance of layer 2 meta classifiers and layer 3 meta classifiers of the proposed ensemble. Figure 2 illustrates a comparison of the prediction behavior of individual base classifiers and the final prediction result generated by the proposed stacking ensemble in layer 3 meta classification. Datasets named; *InClassSurveyData, ClassroomData, xAPI-Edu-Data, liver-disease-lab-data* and *hepatitisData* have shown a lower predictive performance than the others. However, it is evident that the proposed novel stacking ensemble will always train a superior prediction model than the best individual base classifier.

The accuracy of the stacking ensemble prediction models is higher than the accuracies of four intermediate stacks derived in the layer 2 meta classification. In numerous datasets, the last prediction accuracy was generally higher than the layer 2 meta classification and not many of them have demonstrated comparative prediction accuracies to the most outperformed transitional stack of layer 2. All the accuracy esteems are incorporated into table 6 are with their standard deviations.

The Figure 5a, Figure 5b and Figure 5c illustrate the Area Under Curve (AUC) of Receiver Operating Characteristics (ROC) curve for all the datasets to get a visual understanding of how much the model is capable in distinguishing between classes. These graphs illustrate the AUC value comparison of each classifier with the proposed stacking ensemble for each benchmark dataset. As per the illustration of these graphs, it can be clearly seen that the AUC value of the proposed stacking ensemble is very closer to 1. This implies that the proposed ensemble is able to perfectly distinguish

Table 4. Prediction performance of the proposed ensemble under different hyperparameter optimization approaches.

Dataset	Prediction Accuracy		
	SAA	GA	PAO
InClassSurveyData	0.71(+/- 0.09)	0.68 (+/- 0.01)	0.65 (+/-0.09)
InClassDBData	1.0 (+/- 0.0)	0.96 (+/- 0.02)	0.97 (+/-0.02)
LMSDataNew	0.98 (+/- 0.02)	0.96 (+/- 0.01)	0.96 (+/-0.03)
ClassroomData	0.70 (+/- 0.01)	0.66 (+/- 0.01)	0.65 (+/-0.02)
xAPI-Edu-Data	0.67 (+/-0.02)	0.65(+/- 0.02)	0.66 (+/-0.04)
BreastCancerData	0.95 (+/-0.02)	0.90 (+/- 0.02)	0.87 (+/-0.02)
ChronicKidneyDescies	0.99(+/-0.01)	0.90 (+/- 0.02)	0.90 (+/-0.01)
WA_Fn-UseC_-Telco-Customer-Churn	0.77 (+/-0.02)	0.70 (+/- 0.04)	0.72 (+/-0.02)
Liver-Disease-Lab-Data	0.72 (+/-0.01)	0.66 (+/- 0.01)	0.68 (+/-0.09)
DiabetesData	0.75 (+/- 0.05)	0.72 (+/- 0.01)	0.73 (+/-0.01)
HeartData	0.82 (+/- 0.02)	0.79 (+/- 0.02)	0.79 (+/-0.04)
Android_Traffic	0.95 (+/-0.01)	0.91 (+/- 0.05)	0.90 (+/-0.02)
Fakedata	0.95 (+/-0.01)	0.92 (+/- 0.04)	0.93 (+/-0.03)
Brain_TumorData	0.95 (+/-0.01)	0.87(+/- 0.02)	0.85 (+/-0.01)
HepatitisData	0.72 (+/-0.09)	0.69 (+/- 0.03)	0.68 (+/-0.05)

Figure 2. Comparison of Prediction Performances of Base Classifiers and the Proposed Stacking Ensemble

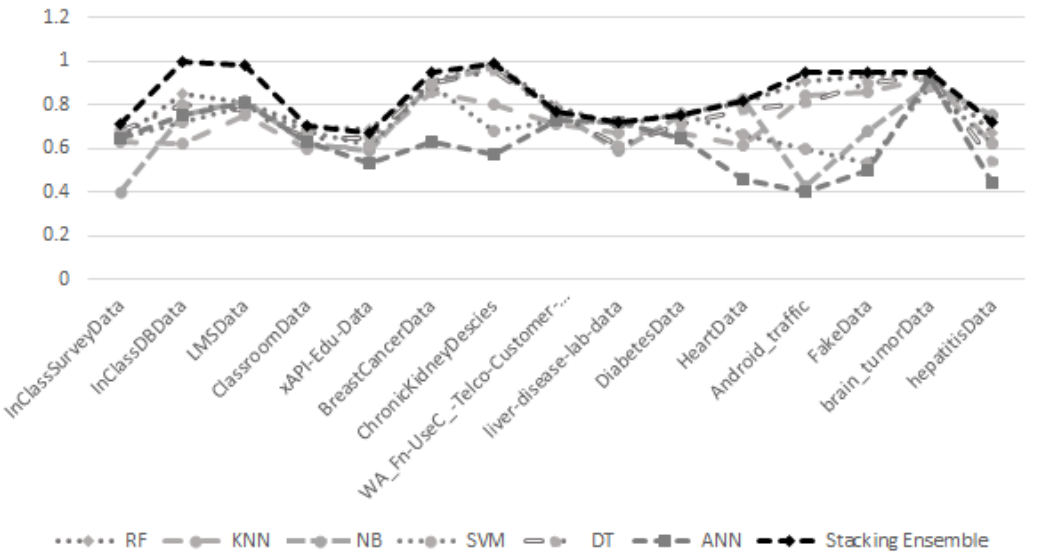


Table 5. Prediction Performance of Layer 2 Meta Classifiers and layer 3 Meta Classification

Dataset		Performance of Layer 2 Meta Classification				Layer 3
		LR	RF	DT	SVM	Accuracy
InClassSurveyData	Accuracy	0.69 (+/- 0.11)	0.68 (+/- 0.1)	0.69 (+/-0.09)	0.65 (+/-0.1)	0.71 (+/- 0.09)
	Precision	0.74	0.69	0.69	0.76	0.74
	Recall	0.77	0.66	0.80	0.66	0.77
InClassDBData	Accuracy	0.97 (+/- 0.01)	0.96 (+/- 0.02)	0.98 (+/-0.01)	0.98 (+/-0.05)	1.0 (+/- 0.00)
	Precision	0.92	0.90	0.95	0.97	1.0
	Recall	0.86	0.90	0.92	0.96	1.0
LMSDataNew	Accuracy	0.97 (+/- 0.02)	0.98 (+/- 0.02)	0.98 (+/-0.02)	0.97 (+/-0.02)	0.98 (+/- 0.02)
	Precision	0.94	0.94	0.95	0.92	0.95
	Recall	0.91	0.86	0.92	0.83	0.92
ClassroomData	Accuracy	0.70 (+/- 0.1)	0.67 (+/- 0.1)	0.68 (+/-0.01)	0.67 (+/-0.01)	0.70 (+/- 0.01)
	Precision	0.74	0.75	0.74	0.73	0.76
	Recall	0.80	0.72	0.70	0.78	0.78
xAPI-Edu-Data	Accuracy	0.65 (+/- 0.01)	0.62 (+/- 0.02)	0.64 (+/-0.05)	0.63 (+/-0.05)	0.67 (+/- 0.02)
	Precision	0.61	0.65	0.64	0.59	0.61
	Recall	0.57	0.56	0.51	0.61	0.58
BreastCancerData	Accuracy	0.95 (+/- 0.02)	0.92(+/- 0.03)	0.92 (+/-0.02)	0.94 (+/-0.02)	0.95 (+/- 0.02)
	Precision	0.87	0.86	0.86	0.86	0.86
	Recall	0.95	0.94	0.93	0.95	0.93
ChronicKidneyDescies	Accuracy	0.98 (+/- 0.01)	0.97 (+/- 0.05)	0.97 (+/-0.01)	0.95 (+/-0.02)	0.99 (+/- 0.01)
	Precision	0.94	0.95	0.94	0.94	0.95
	Recall	0.94	0.95	0.93	0.94	0.94
WA_Fn-UseC_-Telco-Customer-Churn	Accuracy	0.77 (+/- 0.01)	0.78 (0.05)	0.78 (+/-0.01)	0.77 (+/-0.05)	0.77 (+/- 0.02)
	Precision	0.68	0.66	0.66	0.67	0.65
	Recall	0.48	0.45	0.46	0.45	0.48
Liver-Disease-Lab-Data	Accuracy	0.68 (+/- 0.05)	0.68 (0.04)	0.68 (+/-0.02)	0.67 (+/-0.03)	0.72 (+/- 0.01)
	Precision	0.58	0.62	0.59	0.66	0.69
	Recall	0.49	0.64	0.62	0.59	0.85
DiabetesData	Accuracy	0.76 (+/- 0.21)	0.74 (0.09)	0.74 (+/-0.01)	0.75 (+/-0.03)	0.75 (+/- 0.05)
	Precision	0.72	0.69	0.70	0.72	0.72
	Recall	0.74	0.72	0.68	0.70	0.70
HeartData	Accuracy	0.78 (+/- 0.03)	0.78 (0.09)	0.78 (+/-0.05)	0.79 (+/-0.02)	0.82 (+/- 0.02)
	Precision	0.69	0.72	0.75	0.76	0.80
	Recall	0.68	0.70	0.73	0.72	0.76
Android_Traffic	Accuracy	0.92 (+/- 0.01)	0.90 (0.011)	0.92 (+/-0.02)	0.90 (+/-0.01)	0.95 (+/- 0.01)
	Precision	0.82	0.90	0.84	0.80	0.85
	Recall	0.76	0.84	0.80	0.76	0.86

continued on next page

Table 5. Continued

Dataset		Performance of Layer 2 Meta Classification				Layer 3
		LR	RF	DT	SVM	Accuracy
Fakedata	Accuracy	0.90 (+/- 0.01)	0.93 (0.02)	0.93 (+/-0.05)	0.95 (+/-0.01)	0.95 (+/- 0.01)
	Precision	0.90	0.86	0.90	0.85	0.90
	Recall	0.75	0.80	0.91	0.90	0.92
Brain_TumorData	Accuracy	0.93(+/- 0.02)	0.95 (0.05)	0.95 (+/-0.02)	0.92 (+/-0.02)	0.95 (+/- 0.01)
	Precision	0.89	0.92	0.90	0.92	0.92
	Recall	0.86	0.97	0.92	0.85	0.92
HepatitisData	Accuracy	0.65 (+/- 0.10)	0.62 (+/-0.03)	0.64(+/- 0.05)	0.65 (+/- 0.01)	0.72 (+/- 0.09)
	Precision	0.62	0.61	0.60	0.62	0.68
	Recall	0.59	0.48	0.54	0.55	0.67

between all the positive and the negative class points correctly. Accordingly, the proposed stacking ensemble outperforms the reset of the classifiers.

4.3 Validating the Performance of Proposed Ensemble

Statistical significance of the difference between individual base classifiers and the final prediction model of the ensemble is evaluated using paired t-test approach with significance level of 95% using 10 fold cross validation (Kuncheva L.I., 2003), (P.K. Douglas et. al, 2011) and (Shee et. al, 2014). The 1x10 t-test is performed since, the training and testing data sets are not overlapped. There are many deficiencies in the other possibilities such as ten repeats of ten- fold cross validation (10x10) and five two fold cross validation (5x2) approaches. In 10x10 t-test, the test sets and training sets are overlapping which underestimate the true variance of the algorithms. Though 5x2 t-test does not overlap the training and testing datasets, it's not sensitive to the variations of the algorithms (Sherri Rose, 2013). Accordingly, the stacking ensemble with the individual machine learning algorithms and the stacking ensemble final prediction with the intermediate prediction at level 2 are evaluated by hypotheses testing.

Statistical significance of the difference between the prediction accuracy of the proposed staking ensemble and the individual algorithms are measured by forming null and the alternative hypothesis. The null hypothesis (H_0) assumed that both models perform the same and alternative (H_1) assumed that the models perform differently. The hypotheses made for the comparison of proposed stacking ensemble and the Random Forest algorithm can be written as;

H_0: There is no difference between the performance of the proposed stacking ensemble and the Random Forest classifier.

H_1: There is a difference between the performance of the proposed stacking ensemble and the Random Forest classifier.

As this manner, the null and alternative hypotheses were built to all the algorithms for entire datasets and they have been tested using the library supported by Python for paired t-test. According to the table 5, it has been observed that except two datasets the other data sets own their p-value which were below 0.05. This implies that the null hypothesis can be rejected and it has been statistically convincing evidence that random forest and the proposed stacking ensemble perform differently. Similarly, the hypothesis test is conducted for the remaining pairs. Next, the KNN and the proposed stack is chosen for paired t-test. According to the results shown, except a single dataset, all the others are confirmed with 95% of a confidence that there exists a significant difference between the KNN algorithm performance and the novel stack. The Naïve Bayes and stack pair works in the same manner as a single dataset does not meet the criteria and the others' p- values are below the significant threshold value (0.05). Therefore, this proves that there exists a noticeable difference

Table 6. Statistical Analysis of Performance of Base Classifiers and the Stacking Ensemble using P- Values

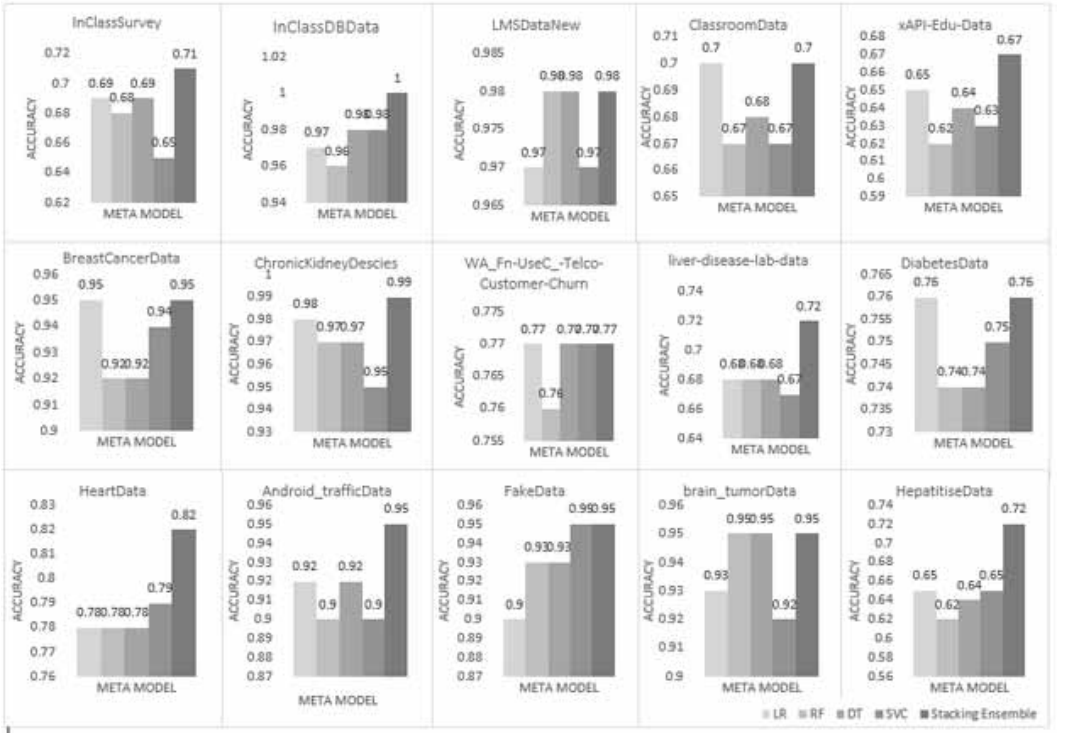
Dataset	P- Value					
	RF vs Stack	KNN vs Stack	NB vs Stack	SVM vs Stack	DT vs Stack	ANN vs Stack
InClassSurveyData	0.023	0.0101	0.0042	0.023	0.0412	0.022
InClassDBData	0.0436	0.036	0.0082	0.0428	0.0154	0.0082
LMSDataNew	0.0132	0.022	0.0015	0.003	0.002	0.0132
ClassroomData	0.0640	0.022	0.0121	0.338	0.018	0.042
xAPI-Edu-Data	0.0429	0.011	0.025	0.0066	0.028	0.036
BreastCancerData	0.010	0.010	0.020	0.040	0.0221	0.000
ChronicKidneyDescies	0.062	0.002	0.004	0.014	0.0393	0.021
WA_Fn-UseC_-Telco-Customer-Churn	0.003	0.045	0.06	0.029	0.0342	0.000
Liver-Disease-Lab-Data	0.001	0.079	0.002	0.033	0.001	0.033
DiabetesData	0.038	0.046	0.05	0.038	0.011	0.0214
HeartData	0.032	0.002	0.01	0.047	0.031	0.028
Android_Traffic	0.002	0.030	0.045	0.022	0.009	0.036
Fakedata	0.004	0.0111	0.005	0.007	0.028	0.0008
Brain_TumorData	0.028	0.0448	0.0102	0.006	0.007	0. 0448
HepatitisData	0.007	0.021	0.004	0.021	0.022	0.034

between the selected algorithm pair in terms of their prediction accuracies. The p- values for SVM and the proposed stacking ensemble were visited and all the datasets reached to 0.05 significant level. Thereby, it can be concluded that the SVM and the stacking ensemble perform differently. The DT and stacking ensemble are paired to initiate the t-test. The null hypothesis has rejected with 95% of a confidence level and implies that these algorithms performed differently in prediction tasks. Finally, the last two algorithm pair has taken for the p-value analysis. According to the results of paired t-test, the null hypothesis could be rejected with 95% of a confidence and alternative hypothesis has been accepted by proving that there exists a significant difference between the performances of them.

The statistical significance levels of the differences between the prediction accuracies of meta models derived in layer 2 and layer 3 are illustrated in the table 7. The main purpose of this evaluation is to see whether there exists a value of including an additional layer to the proposed stacking ensemble as the main contribution of this study.

It can be clearly seen that the significance level of accuracy between layer 2 logistic regression outcome and the layer 3 stack is below the threshold (< 0.05) for all the datasets. This implies that there exists a noticeable difference between them and hence, the null hypothesis was rejected. The significant level of layer 2 Random Forest Classifier outcome and the layer 3 stack is below the threshold (< 0.05) for thirteen out of fifteen datasets, nearly 86% of datasets. Therefore, it can be concluded as there exists a difference between the prediction accuracies of them. Again the null hypothesis was rejected and alternative hypothesis was accepted. The significant values given by paired t- test were below the threshold (0.05) for about 93% of datasets in testing the hypotheses for layer 2 Support Vector Classifier outcome and the layer 3 stack. Thereby, the null hypothesis was rejected and alternative hypothesis was accepted by concluding that there is a significant difference between them. Finally, the last two pairs were also applied to the paired t-test and accordingly, the

Figure 3. Comparison of Prediction Performances of Layer 2 Meta Classifiers and the Final Prediction given by Layer 3 Meta Classifier of Proposed Stacking Ensemble



null hypothesis was rejected and alternative hypothesis was accepted since the significant values are below 0.05 for all the test datasets.

This statistical figures have proven that the enhancement of the stack generalization into three layers could derive significant and noticeable accurate prediction outcome for any machine learning application.

Further, this novel approach has been compared with selected bagging and boosting algorithms to confirm the accuracy level. Accordingly, the Adaboosting algorithm, Gradient Boosting algorithm, XGB algorithm and Bagging classifier algorithm are applied to the datasets used for the experiment. Table 8 illustrates the outcome of the analysis.

The outcome of above evaluation is graphically shown in figure 4. As needs be, it is truly evident that the proposed stacking ensemble claims an essentially better performance. It contends with bagging and boosting algorithms and has demonstrated a higher or a similar accuracy measures for the chose datasets.

The results obtained in this study have shown that the lower performance occurred for some datasets due to the less number of instances with high number of dimensions. The *BreastCancerData* dataset also has lesser instance count compare with the datasets which have high performance but, it has less number of dimensions compared to the datasets of lower performance. This fact insight that the number of dimensions in the dataset has a direct impact on the prediction performance. Generally, the machine learning algorithms are applied on the preprocessed, error-free, noiseless, and non-redundant data. Then the analysis commenced by performing feature selection/ feature engineering as well. The number of dimensions of some datasets might reduce by the dimension reduction approach. Generally, if the number of instances in the dataset is high, there is high robustness and reliability with the accuracy of the prediction model in machine learning. In a situation where a dataset has 20000,

Figure 4. Illustration of Prediction Performance of Adaboosting, Gradient Boosting, XGB and Bagging Classifier with Proposed Stacking Ensemble.

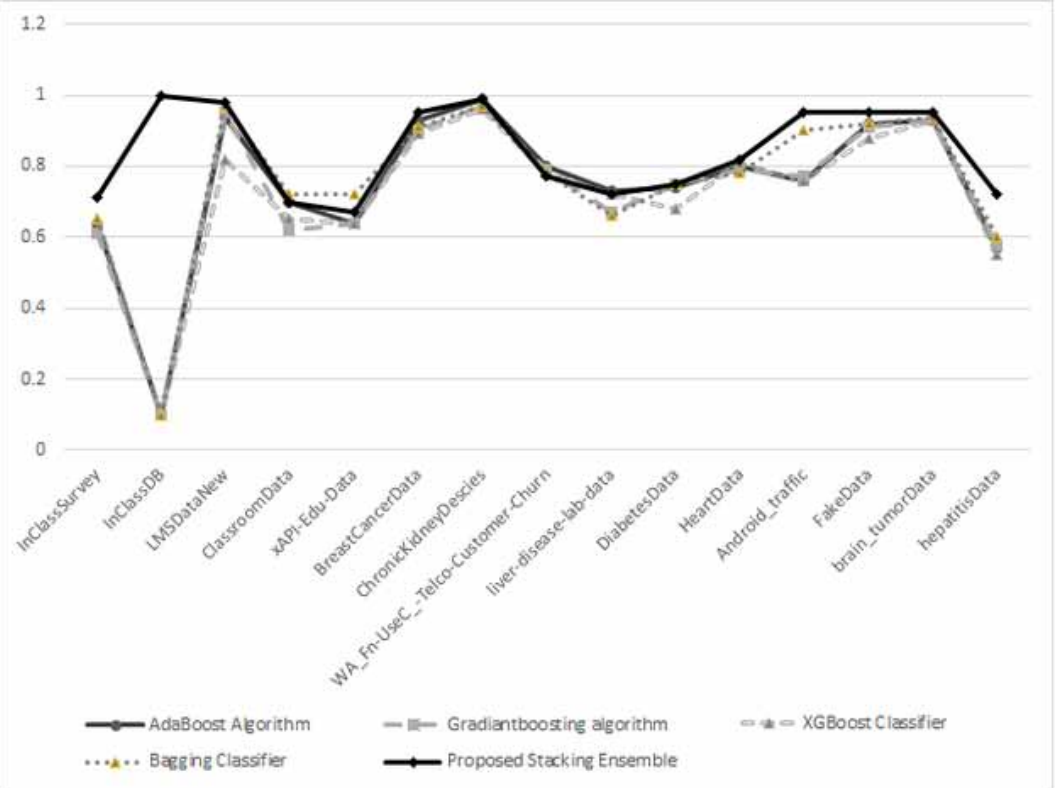


Table 7. Statistical Analysis of Performance of Layer 2 Meta Classifiers and the Layer 3 Meta Classification of Stacking Ensemble using P- Values

Dataset	P-Values of Stack Vs			
	LR Meta Classifier	RF Meta Classifier	SVM Meta Classifier	DT Meta Classifier
InClassSurveyData	0.0335	0.028	0.006	0.0045
InClassDBData	0.02	0.001	0.03	0.0410
LMSDataNew	0.0435	0.058	0.046	0.053
ClassroomData	0.012	0.006	0.047	0.0346
xAPI-Edu-Data	0.006	0.005	0.0013	0.042
BreastCancerData	0.003	0.0064	0.005	0.016
ChronicKidneyDescies	0.042	0.013	0.016	0.0242
WA_Fn-UseC_-Telco-Customer-Churn	0.044	0.048	0.056	0.031
Liver-Disease-Lab-Data	0.007	0.0078	0.020	0.015
DiabetesData	0.004	0.001	0.016	0.001
HeartData	0.012	0.013	0.0059	0.042
Android_Traffic	0.002	0.012	0.016	0.03
Fakedata	0.009	0.002	0.006	0.043
Brain_TumorData	0.02	0.064	0.001	0.041
HepatitisData	0.012	0.013	0.0005	0.0043

Table 8. Comparison of Prediction Performance of Adaboosting, Gradient Boosting, XGB and Bagging Classifier with Proposed Stacking Ensemble.

	Accuracy			
Dataset	AdaBoost Algorithm	Gradiantboosting algorithm	XGBoost Classifier	Bagging Classifier
InClassSurveyData	0.64 (+/-0.07)	0.61 (+/-0.13)	0.65 (+/-0.14)	0.65 (+/-0.13)
InClassDBData	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)
LMSDataNew	0.94 (+/-0.03)	0.95 (+/-0.03)	0.82 (+/-0.05)	0.97(0+/-0.01)
ClassroomData	0.7 (+/-0.4)	0.62 (+/-0.11)	0.65 (+/-0.14)	0.72 (+/- 0.07)
xAPI-Edu-Data	0.64 (+/-0.6)	0.64 (+/-0.06)	0.64 (+/-0.05)	0.72 (+/- 0.05)
BreastCancerData	0.93 (+/-0.02)	0.90 (+/- 0.02)	0.89 (+/-0.03)	0.91 (+/-0.02)
ChronicKidneyDescies	0.99 (+/- 0.01)	0.97 (+/- 0.02)	0.96(+/-0.03)	0.97 (+/- 0.02)
WA_Fn-UseC_-Telco-Customer-Churn	0.8 (+/-0.007)	0.79 (+/- 0.01)	0.78 (+/-0.01)	0.78 (+/- 0.01)
Liver-Disease-Lab-Data	0.73 (+/-0.06)	0.67 (+/- 0.07)	0.72 (+/-0.08)	0.66 (+/- 0.07)
DiabetesData	0.74 (+/-0.37)	0.75 (+/- 0.04)	0.68 (+/-0.06)	0.75 (+/- 0.05)
HeartData	0.8 (+/- 0.09)	0.79 (+/- 0.07)	0.81 (+/-0.05)	0.78 (+/- 0.08)
Android_Traffic	0.76 (+/-0.012)	0.77 (+/-0.01)	0.76 (+/-0.01)	0.9 (+/- 0.01)
Fakedata	0.92 (+/-0.03)	0.91 (+/- 0.03)	0.88(+/-0.35)	0.92 (+/- 0.03)
Brain_TumorData	0.93 (+/-0.017)	0.93 (+/- 0.01)	0.93(+/-0.01)	0.94 (+/- 0.02)
HepatitisData	0.57 (+/-0.11)	0.58 (+/- 0.11)	0.55(+/-0.17)	0.6 (+/- 0.11)

50000, or more than that number of instances, even though there might not get a higher prediction accuracy, the prediction result can be stronger than a dataset of a lesser number of instances. There are some studies in which the researchers have obtained a better prediction result from a smaller dataset than even with large data instances (Noem- DeCastro-Garc et. al, 2019).. From the accuracy measures derived by the proposed ensemble have shown diverse values for various datasets. As per that result, there was no relationship between the size of the dataset and their accuracies. However, if the analysis performs with a dataset with images, having a huge set of images in the dataset will be an advantage for a better result.

5. CONTRIBUTION AND IMPLICATIONS

This study contributes to data mining in two aspects. Firstly, proposed a novel architecture for the existing stacked generalization. Secondly, the architecture is optimized using SAA hyperparameter optimization approach to obtain the best prediction result for any classification problem.

The experiments were executed on an Intel Core (R) i5 in 8GB memory 72000 CPU @ 2.7GHz machine with NVDIA TITAN GPU processor of which the prediction result was independent from the execution environment. However, when the size of the dataset is getting increased, there was a delay in producing the prediction result of the proposed ensemble than a small dataset. On average it required 12.87 milliseconds to complete the execution of datasets below 800 instances and 17.76 milliseconds to complete the execution of datasets above 800 instances. This implies a considerable computation time is taken by the proposed ensemble to derive the optimal prediction results. Hyperparameter optimization of the classifiers may also be a reason for the increase of computational time of the classifier. As previous studies have shown, the spatial and time complexity of a model are affected by the size of the dataset (Noem- DeCastro-Garc et. al, 2019). Therefore, while increasing the size of the dataset, the computational complexity will increase. Since diverse classifiers are involving and as they behave in diverse methods to produce the prediction task in the proposed ensemble, the complexity of this approach will be getting increased. However, while increasing the accuracy of the ensemble, it increases the time and the special complexity. Therefore, this can be considered as another limitation of this study. Nevertheless, the user could execute the proposed ensemble in a computer which is suitable for data analysis tasks with a high performing processor and with more memory capacity to overcome this limitation.

The proposed stacking ensemble has been evaluated with many classifiers in the layer 2 and layer 3 meta classification to ascertain the best classifiers to obtain the optimal output. This was a challenging task, as the evaluations had carried out with multiple benchmark datasets. In some situations, the ensemble was slow to produce the output due to the computational complexity of the ensemble environment. Each experiment was repeated with 3-fold, 5-fold and 10-fold cross validation in order to select the best cross validation approach as well. However, the classifiers that generated the most accurate results was utilized in each meta classification layer to finalize the ensemble.

6. CONCLUSIONS AND FUTURE WORK

In addressing several deficiencies identified in past researches, an innovative machine learning solution has been proposed via a stacking ensemble by combining various individual classifiers. Rather than selecting the algorithm which generates the prediction result with most outperforming accuracy, it combines the results of every classifier to generate the best result through a weighed ranking method and a hyperparameter optimization procedure. Past studies have shown that, though the stacking leads to a better prediction accuracy than the individual classifiers, there exists some limitations. This research aimed to address those shortcomings. First, the commitment of individual classifiers to the prediction is unclear in stacking. This study has proposed a weighted scoring approach to select the classifiers with the best prediction accuracies into the next level of the stack. Secondly, the parameters

of the base learners may affect the final prediction accuracy. Therefore, the selection of the parameters must be done very cautiously. This issue has been addressed by optimizing the hyperparameters of base classifiers as well as meta-classifiers. Finally, the enhancement of the prediction accuracy has been achieved by introducing a three layered stacked generalization framework. After a comparison of the performance of novel stacking ensemble, it has shown that a noticeably a better result could generate by the proposed method. The statistical tests are proven that the prediction models generates by the novel ensemble does the prediction more significantly better than the individuals. Thereby, the researchers were able to propose an optimal stacking ensemble learner with improved accuracy and robustness. As future work, the authors' intention is to research on optimizing the proposed novel stacking ensemble to increase the prediction accuracy while minimizing the time and special complexity. In order to do that the authors are expecting to apply big data technology such as Spark on the proposed ensemble to manage the big collection of data and derive the optimal solution with less complexity (Cai Z et. al., 2014, Ni Z., 2013 and Zaharia M, 2012) . Finally, the authors supposed to propose an ensemble for regression tasks as well.

UCI machine learning repository. (n.d.). http://archive.ics.uci.edu/ml/datasets.html

ACKNOWLEDGMENT

The authors would like to express their gratitude to University of Kelaniya and Sri Lanka Institute of Information Technology for the fullest support provided in gathering data for this study.

REFERENCES

Babalyan, K., Sultanov, R., Generozov, E., Sharova, E., Kostryukova, E., Larin, A., Kanygina, A., Govorun, V., & Arapidi, G. (2018). LogLoss-BERAF: An ensemble-based machine learning model for constructing highly accurate diagnostic sets of methylation sites accounting for heterogeneity in prostate cancer. *PLoS One*, *13*(11), e0204371. doi:10.1371/journal.pone.0204371 PMID:30388122

Bardenet, M. B., Kégl, B., & Sebag, M. (2013). Collaborative hyperparameter tuning. *Proceedings of the 30th International Conference on International Conference on Machine Learning*, *28*, 199-207.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. doi:10.1007/BF00058655

Brown, G. (2017). Ensemble Learning. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning and Data Mining* (pp. 393–402). Springer US. doi:10.1007/978-1-4899-7687-1_252

Cai, Z., Gao, J., Luo, S., Perez, L. L., Vagena, Z., & Jermaine, C. (2014). A comparison of platforms for implementing and running very large scale machine learning algorithms. *Proceedings of the 2014 ACM SIGMOD international conference on Management of data (SIGMOD'14)*, 1371–82. doi:10.1145/2588555.2593680

Clarke, B. (2003). *Comparing Bayes model averaging and stacking when model approximation error cannot be ignored* (Vol. 4). JMLR.org.

DeCastro-Garcia, Muñoz Castañeda, Escudero Garcia, & Carriegos. (2019). Effect of the Sampling of a Dataset in the Hyperparameter Optimization Phase over the Efficiency of a Machine Learning Algorithm. *Complexity*. doi:10.1155/2019/6278908

Delahaye, D., Chaimatanan, S., & Mongeau, M. (2019). Simulated annealing: From basics to applications. In Handbook of Metaheuristics. Springer.

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, *7*, 1–30.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In. Lecture Notes in Computer Science: Vol. 1857. *Multiple Classifier Systems. MCS 2000*. Springer. doi:10.1007/3-540-45014-9_1

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, *40*(2), 139–158. doi:10.1023/A:1007607513941

Douglas, P. K., Harris, S., Yuille, A., & Cohen, M. S. (2011). Performance comparison of machine learning algorithms and number of independent components used in fMRI decoding of belief vs. disbelief. *NeuroImage*, *56*(2), 544–553. doi:10.1016/j.neuroimage.2010.11.002 PMID:21073969

Džeroski, S., & Ženko, B. (2004). Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning*, *54*(3), 255–273. doi:10.1023/B:MACH.0000015881.36452.6e

Figshare data. (n.d.). https://figshare.com/

James Bergstra, Y. (2012). Bengio. Random Search for Hyper-Prarmeter Optimization. *Journal of Machine Learning Research*, *13*(1), 281–305.

Jones, E., Oliphant, T., & Peterson, P. (2001). *SciPy: Open source scientific tools for Python*. Academic Press.

Kaggle datasets. (n.d.). https://www.kaggle.com/datasets

Kim, J., Lee, B., Shaw, M., Chang, H., & Nelson, W. (2001). Application of Decision Tree Induction Techniques to Personalized Advertisements on Internet Storefronts. *International Journal of Electronic Commerce*, *5*(3), 45–62. doi:10.1080/10864415.2001.11044215

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science Journal, 220(4598), 671-680. doi:10.1126/science.220.4598.671

Kuncheva, L. I. (2014). *Combining Pattern Classifiers: Methods and Algorithms* (2nd ed.). Retrieved from https://www.wiley.com/en-us/Combining+Pattern+Classifiers%3A+Methods+and+Algorithms%2C+2nd+Edition-p-9781118914540

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning*, *51*(2), 181–207. doi:10.1023/A:1022859003006

Ladds, M., Thompson, A., Kadar, J., Slip, D., Hocking, D., & Harcourt, R. (2017). Super machine learning: Improving accuracy and reducing variance of behaviour classification from accelerometry. *Animal Biotelemetry*, *5*(1), 8. Advance online publication. doi:10.1186/s40317-017-0123-1

Large, J., Lines, J., & Bagnall, A. (2019). A probabilistic classifier ensemble weighting scheme based on cross-validated accuracy estimates. *Data Mining and Knowledge Discovery*, *33*(6), 1–36. doi:10.1007/s10618-019-00638-y PMID:31632184

Lévesque, Gagné, & Sabourin. (2016). Bayesian Hyperparameter Optimization for Ensemble Learning. Academic Press.

Li, C., Wang, J., Hu, L., & Gong, P. (2013). Comparison of Classification Algorithms and Training Sample Sizes in Urban Land Classification with Landsat Thematic Mapper Imagery. *Remote Sensing*, *6*(2), 964–983. Advance online publication. doi:10.3390/rs6020964

Löw, Michel, Dech, & Conrad. (2013). Impact of Feature Selection on the Accuracy and Spatial Uncertainty of per-Field Crop Classification Using Support Vector Machines. *ISPRS Journal of Photogrammetry and Remote Sensing, 85*, 102–119. doi:.isprsjprs.2013.08.00710.1016/j

Markovic, R., Wolf, S., Cao, J., Spinnräker, E., Wölki, D., Frisch, J., & van Treeck, C. (2017). Comparison of Different Classification Algorithms for the Detection of User's Interaction with Windows in Office Buildings. *Energy Procedia*, *122*, 337–342. doi:10.1016/j.egypro.2017.07.333

Naimi, A. I., & Balzer, L. B. (2018). Stacked generalization: An introduction to super learning. *European Journal of Epidemiology*, *33*(5), 459–464. doi:10.1007/s10654-018-0390-z PMID:29637384

Ni, Z. (2013). *Comparative Evaluation of Spark and Stratosphere* (Thesis). KTH Royal Institute of Technology.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., & Grisel, O. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (1992). Numerical Recipes in C (2nd ed.). Academic Press.

Purushotham, S., Meng, C., Che, Z., & Liu, Y. (2017). Benchmark of Deep Learning Models on Large Healthcare MIMIC Datasets. *Journal of Biomedical Informatics*, *83*, 112–134. Advance online publication. doi:10.1016/j.jbi.2018.04.007 PMID:29879470

Qureshi, A. S., Khan, A., Zameer, A., & Usman, A. (2017). Wind power prediction using deep neural network based meta regression and transfer learning. *Applied Soft Computing*, *58*, 742–755. doi:10.1016/j.asoc.2017.05.031

Qureshi, A.S., Khan, A., Zameer, A., & Usman, A. (n.d.). *Wind power prediction using deep neural network based meta regression and transfer learning*. Academic Press.

Re, M., & Valentini, G. (2012). Ensemble methods. *RE:view*.

Romesburg, H. C. (2014). *Cluster Analysis for Researchers. Melbourne*. Krieger.

Rose, S. (2013, March 1). Mortality Risk Score Prediction in an Elderly Population Using Machine Learning. *American Journal of Epidemiology*, *177*(5), 443–452. doi:10.1093/aje/kws241 PMID:23364879

Sanvitha Kasthuriarachchi, K. T., Liyanage, S. R., & Bhatt, C. M. (2018). A Data Mining Approach to Identify the Factors Affecting the Academic Success of Tertiary Students in Sri Lanka. In S. Caballé & J. Conesa (Eds.), *Software Data Engineering for Network eLearning Environments. Lecture Notes on Data Engineering and Communications Technologies* (Vol. 11). Springer. doi:10.1007/978-3-319-68318-8_9

Shahhosseini, M., Hu, G., & Pham, H. (2020). *Optimizing Ensemble Weights for Machine Learning Models: A Case Study for Housing Price Prediction*. 10.1007/978-3-030-30967-1_9

Syarif, I., Zaluska, E., Prugel-Bennett, A. & Wills, G. (2012). *Application of bagging, boosting and stacking to intrusion detection*. Academic Press.

Ting, K. M., & Witten, I. H. (1999). Issues in stacked generalization. *Journal of Artificial Intelligence Research*, *10*, 271–289. doi:10.1613/jair.594

Li & Shami. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing, 415*, 295-316. 10.1016/j.neucom.2020.07.061

Ren, Y., Zhang, L., & Suganthan, P. N. (2016, February). Ensemble Classification and Regression-Recent Developments, Applications and Future Directions [Review Article]. *IEEE Computational Intelligence Magazine*, *11*(1), 41–53. doi:10.1109/MCI.2015.2471235

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., & Zhao, S. (2019). Applications of machine learning in drug discovery and development. *Nature Reviews. Drug Discovery*, *18*(6), 463–477. doi:10.1038/s41573-019-0024-5 PMID:30976107

Varian, H. R. (2014). Big Data: New Tricks for Econometrics. *The Journal of Economic Perspectives*, *28*(2), 3–28. doi:10.1257/jep.28.2.3

Wang, G., Jia, Q. S., Qiao, J., Bi, J., & Liu, C. (2020). A sparse deep belief network with efficient fuzzy learning framework. *Neural Networks*, *121*, 430–440. doi:10.1016/j.neunet.2019.09.035 PMID:31610414

Wang, G., Qiao, J., Bi, J., Jia, Q., & Zhou, M. (2019). An Adaptive Deep Belief Network With Sparse Restricted Boltzmann Machines. *IEEE Transactions on Neural Networks and Learning Systems*. Advance online publication. doi:10.1109/TNNLS.2019.2952864 PMID:31880561

Wolpert, D., & Macready, W. (1996). *No Free Lunch Theorems for Search*. Academic Press.

Wolpert, D. H. (1992). Wolpert., David H. Stacked generalization. *Neural Networks*, *5*(2), 241–259. doi:10.1016/S0893-6080(05)80023-1

Wong, J., Manderson, T., Abrahamowicz, M., Buckeridge, D., & Tamblyn, R. (2019). Can Hyperparameter Tuning Improve the Performance of a Super Learner?: A Case Study. *Epidemiology (Cambridge, Mass.)*, *30*(4), 1. doi:10.1097/EDE.0000000000001027 PMID:30985529

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *J Electron Sci Technol*, *17*, 26–40. doi:10.11989/JEST.1674-862X.80904120

Yang, P., Yang, J., Zhou, B., & Zomaya, A. (2010). A Review of Ensemble Methods in Bioinformatics. *Current Bioinformatics*, *5*. Advance online publication. doi:10.2174/157489310794072508

Yogatama, D., & Mann, G. (2014). *Efficient Transfer Learning Method for Automatic Hyperparameter Tuning*. AISTATS.

Zaharia, M., Chowdhury, M., Das, T., & Dave, A. (2012). Fast and interactive analytics over Hadoop data with Spark. *USENIX Login.*, *37*(4), 45–51.

Zhou, G., Sohn, K., & Lee, H. (2014). Online incremental feature learning with denoising autoencoders. *International Conference on Artificial Intelligence and Statistics*, 1453–1461.

APPENDIX

This section provides a detailed description about the benchmark datasets of this study.
KNN, RF, NB, SVM, DT, ANN and Logistic Regression were executed utilizing Python *Scikitlearn* library. Every one of these algorithms were applied to fifteen diverse datasets. The details of the datasets are as follows.

(1) *LMSData*, which was gathered by getting to the MOODLE information of a course module offered for thirteen weeks of an Information Technology degree program in a Sri Lankan college. It contained 799 occasions and 11 highlights.
(2) *ClassroomData*, was accumulated by conveying an organized poll among the undergraduate students who were taken a crack at an Information Technology degree program. There are 170 occasions and 20 factors in this dataset.
(3) *InClassSurveyData*, which was gathered through another open-ended questionnaire with university students. There are 171 examples and 20 highlights in this dataset.
(4) *InClassDBData* was another dataset used to make the expectation and assess the precision of the forecast. This has gathered through a blend of a study and getting to the undergraduates' records from the college database. It contained 3795 occurrences and 18 highlights.

Publicly available datasets were taken from various vaults. The insights concerning on the web datasets utilized are as per the following.

(5) *xAPI-Edu-Data* contains 481 examples and 14 highlights.
(6) *BreastCancerData* dataset has taken from UCI machine learning repository which contains 9 attributes and 268 instances (UCI machine learning repository, 2020).
(7) *ChronicKidneyDescies* dataset has taken from UCI machine learning repository and it contains 25 attributes and 400 instances.
(8) *WA_Fn-UseC_-Telco-Customer-Churn* is one of the Keggle dataset with 21 attributes and 7043 instances (Kaggle datasets, 2020).
(9) *liver-infection lab-information* has gotten from UCI machine learning repository and it has 11 attributes and 483 instances.
(10) *DiabetesData* is again taken from UCI AI vault. It has 20 attributes and 769 instances.
(11) *HeartData* is taken from UCI machine learning repository which contains 303 records with 14 attributes.
(12) *Android_traffic* dataset is acquired from Keggle repository which has 17 attributes and 7846 instances.
(13) *FakeData* dataset was obtained from Keggle repository and it contains 12 attributes and 697 instances (Kaggle datasets, 2020).
(14) *brain_tumorData* is gotten from Figshare repository with 1449 instances and 19 attributes (Figshare, 2020).
(15) *hepatitisData* is a dataset hosted at UCI machine learning repository and it has 19 attributes and 155 instances.

Figure 5A. AUC- ROC Curve Analysis of Individual Classifiers and Proposed Ensemble – Part 1

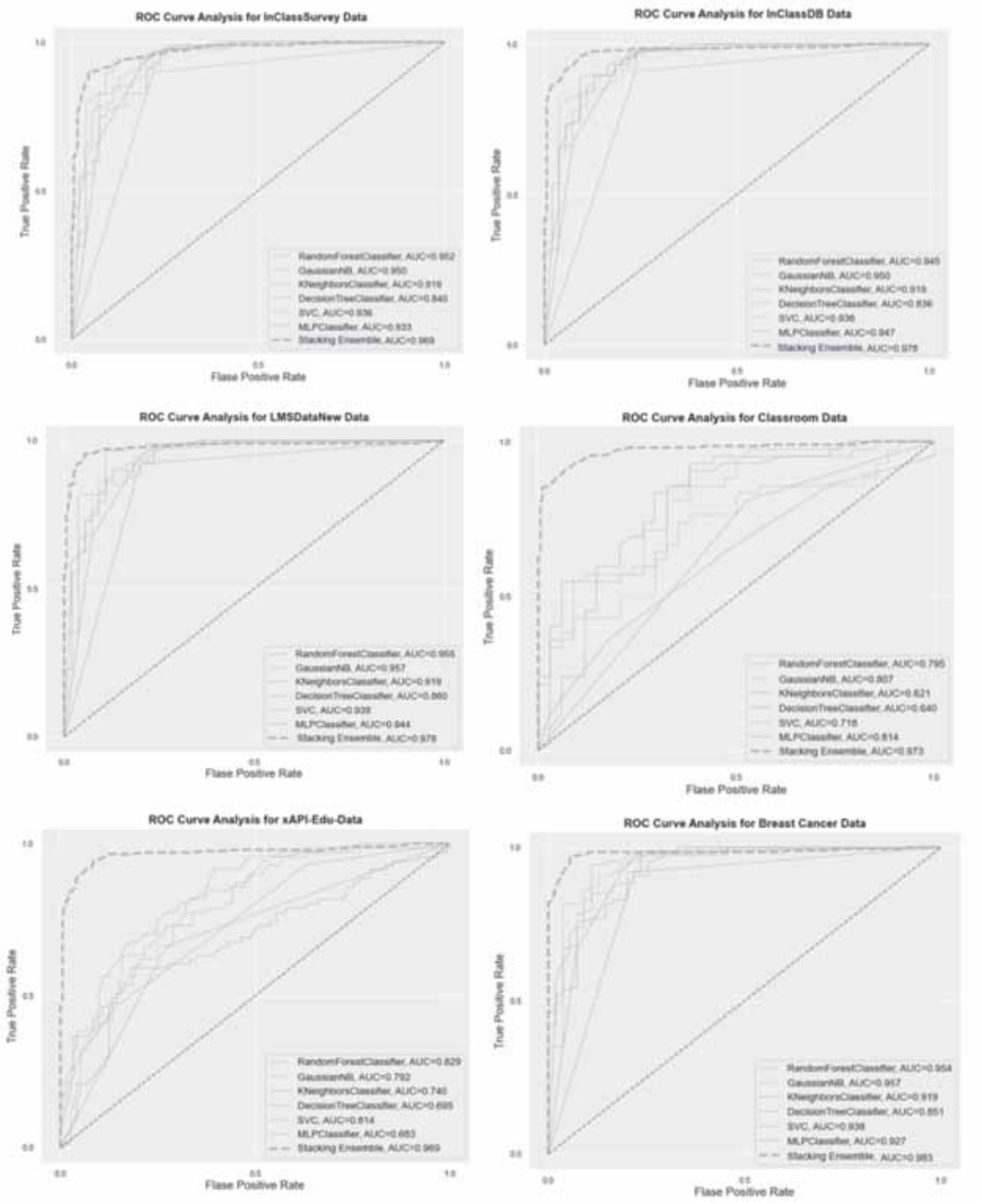


Figure 5B. AUC- ROC Curve Analysis of Individual Classifiers and Proposed Ensemble – Part 2

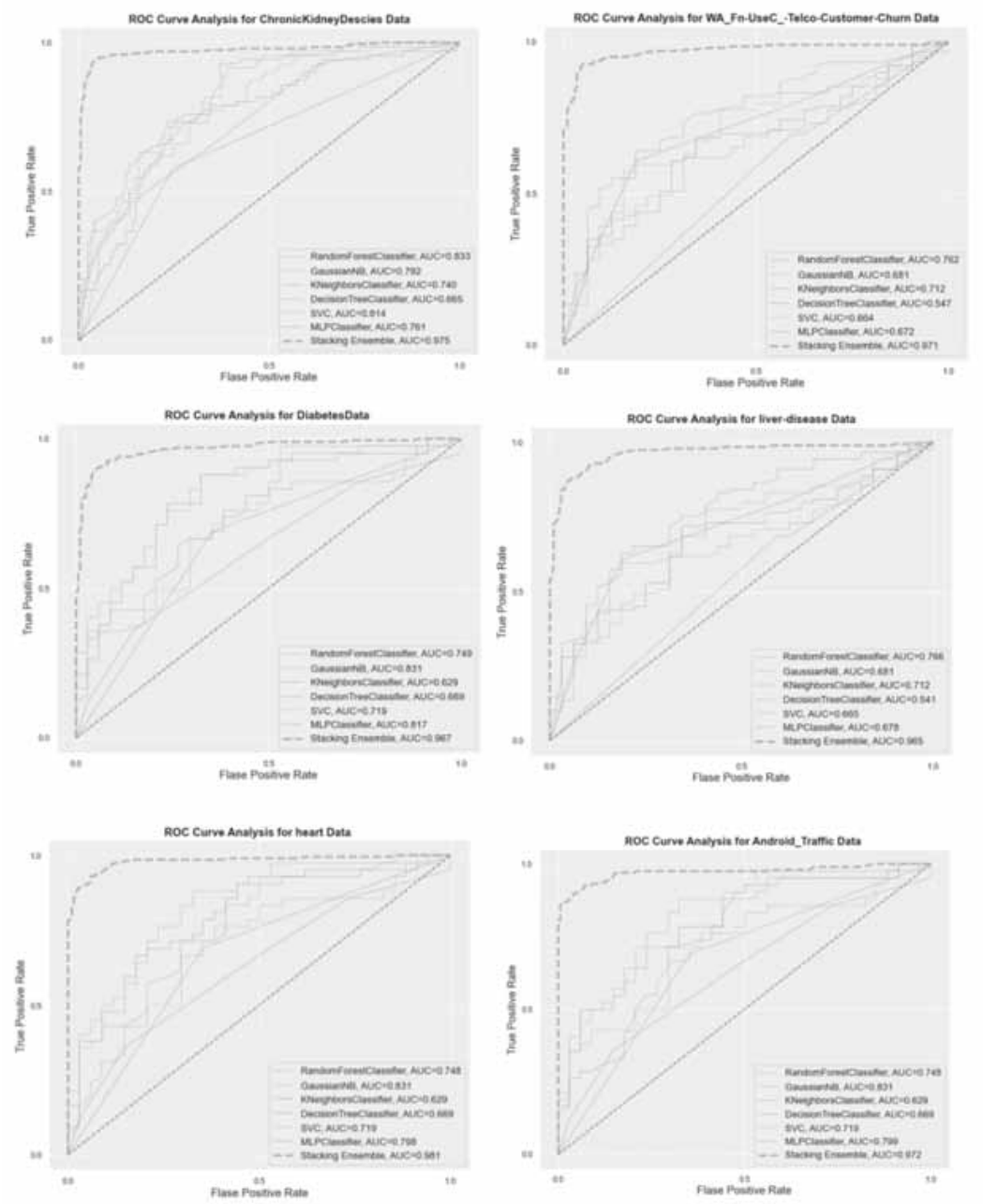
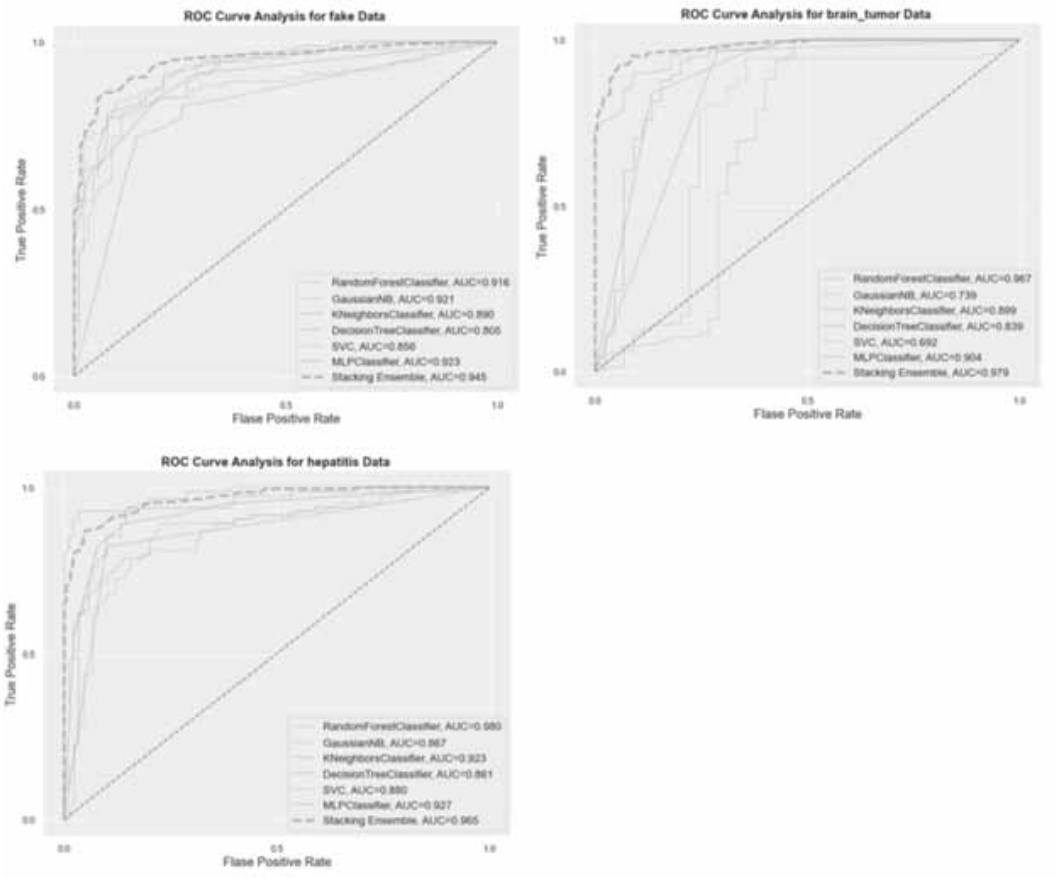


Figure 5C. AUC- ROC Curve Analysis of Individual Classifiers and Proposed Ensemble – Part 3



K. T. S. Kasthuriarachchi has a B.Sc. in Information Technology and M.Sc. in Information Technology. Currently, she is a Ph.D. degree candidate in the Department of Software Engineering, Faculty of Information and Communication Technology at the University of Kelaniya, Sri Lanka. Her main topics of interest are data mining, data analysis, and education. She has published her findings in many conferences and journals. She has contributed to book chapters as well. Kasthuriarachchi has won the best poster award in the International Poster Presentation Competition held by the young scientists' association.

Sidath R. Liyanage graduated from University of Kelaniya with BSc Honours in Statistics and Computer Science in 2005, he completed Master of Philosophy in Computer Engineering from University of Peradeniya in 2009 and received PhD from National University of Singapore in 2013. His research interests are in Brain Computer Interfaces, Data Science and applications of Machine Learning and pattern recognition. He is the Head of Department and a Senior Lecturer attached to Department of Software Engineering, Faculty of Computing and Technology, University of Kelaniya. He is a member of IEEE and a Council member of Sri Lanka Association for Artificial Intelligence.