Ensemble Learning for Classification - A Survey

Athulya Krishna, M
M.Tech (Computer Science)
Department of Computer Science & Engineering
NSS College of Engineering, Palakkad, Kerala, India

Abstract:
Ensemble learning is a classification process that uses multiple machine learning models that are strategically constructed to solve a particular problem. It classifies new data points by combining the prediction of each base classifier. Initially bayesian averaging is used as an ensemble method but more recent algorithms include Voting, Stacking, Bagging and Boosting. This survey discusses popular ensemble methods for classification and explains why ensembles can often perform better than any single base classifier. The study compares different base classifiers and different ensemble classifiers and revealed that boosting algorithm LightGBM is the most efficient tool for classification problems in machine learning.

Keywords: Bagging, Boosting, Classification, Gradient boosting, Machine Learning, Stacking, Voting.

I. INTRODUCTION

Classification is a supervised learning approach. The computer program learns from the data input given to it and using this learning to classify new observations. It is the process of predicting the class of given data points. Classes are called as targets/ labels or categories. Classification algorithms span a wide area of learning methods such as decision trees, bayesian classification, rule-based methods, memory- based learning and support vector machines to predict future instances with past datasets. Binary classification is the simplest type of machine learning classification. The goal of binary classification is to categorize data points into one of two buckets either 0 or 1, or true or false. For example, email spam detection can be identified as a classification problem. This is binary classification, there are only two classes as spam and not spam. A classifier utilizes training data to understand how given input variables relate to the class. Here known spam and non-spam emails have to be used as the training data. There are many kinds of algorithms used for binary classification like Decision tree, Support Vector Machine (SVM), Naive Bayes (NB), k-Nearest Neighbors (kNN), Multilayer Perceptron (MLP). The word meaning of ensemble is ‘union of parts’ it is a Latin-derived word. Ensemble learning is a way of generating a new classifier from various base classifiers, which performs better than any constituent classifier. It is also called classifier combination methods. Several classifier models called “weak” or “base” classifiers are trained and their results are combined through a voting or averaging process. The ensemble model is shown in figure 1, these base classifiers may differ in the algorithm used, hyper parameters, representation or the training set. The main premise of ensemble learning is that by combining multiple models, the errors of the single base learner will likely be compensated by other base learners and as a result, the overall prediction performance of the ensemble would be better than that of a single base learner [1].

There are several reasons for why ensemble methods often improve predictive performance

• Over-fitting avoidance: When just a small amount of data is available, a learning algorithm is prone to finding many different hypotheses that predict all of the training data perfectly while making poor predictions for unseen instances. Averaging different hypotheses reduces the risk of choosing an incorrect hypothesis and therefore, improves the overall predictive performance.

• Computational advantage: Ensemble methods decrease the risk of obtaining a local minimum by combining several learners.

• Representation: By combining different models, the search space may be extended and hence, a better fit for the data space is achieved.

• Small volumes of data: By re-sampling with replacement, numerous classifiers learn on samples of the same data, yielding a higher performance

There are different types of ensemble learning based on the organization of base learners that are sequential and parallel ensemble methods. In sequential ensemble methods the base learners are arranged sequentially. It exploit the dependence between the base learners. The overall performance of the model can be boosted by weighing previously mislabeled examples with higher weight. In parallel ensemble methods the base learners are arranged in parallel. It exploit independence between the base learners since the error can be reduced by averaging. Based on the type of base learner there are homogeneous ensembles and heterogeneous ensembles. Learners of the same type, leading to homogeneous ensembles and learners of different types, leading to heterogeneous ensembles. The key objective of ensemble learning is to reduce bias and variance. Bias is the difference between the average prediction of a model and the correct value that is trying to predict. A high bias error means to have an under-performing model that keeps on missing important trends. Variance is the measure of how the prediction made on the same observation different from each other. A high variance model will over-fit on the training population and perform badly on any observation beyond training. To improve the model performance, then increase the complexity of the model. As we continue to make a model more complex, then end up over-fitting the model, and hence model will start
suffering from high variance. A better model should maintain a balance between these two types of errors. This is known as the trade-off management of bias-variance errors. Ensemble learning is one way to execute this trade of analysis. The rest of this paper is organized as follows. In Section II, the different ensemble learning methods for classification are introduced. In Section III, different approaches used in classification task are discussed and analysed. Finally, this paper is concluded in Section IV.

Figure.1. Ensemble Model

II. ENSEMBLE METHODS

The following subsections describe popular ensemble learning method and classifiers for binary classification.

A. Output fusion
Output fusion refers to the process of integrating the base model outputs into a single output.

1. Weighting methods: In the weighting approach assigning weight to each base model then outputs are combined to form the final output. Majority voting is the simplest weighting method for classification problems as the selected class is the one with the most votes. For regression problems, this concept is manifested by averaging the outputs across the base learners. In this method, take an average of predictions from all the models and use them to make the final prediction.

2. Meta-learning methods: Meta-learning is a process of learning from learners. Meta-learning models differ from standard machine learning models since they include more than one learning stage. In the meta-learning methods, the individual inducer outputs serve as an input to the meta-learner that generates the final output. Meta-learning methods work well in cases where certain base learners have different performances on different sub-spaces

B. Stacking
Stacking is an ensemble learning method that uses predictions from multiple models to build a new model. That is the process works with a layered architecture [2]. Each layer is composed of one or more classifiers. The prediction from the first layer is used to extend the original feature vector. In stacking an ensemble of classifiers is first trained using samples of the training data, creating Tier 1 classifiers, then their output is used to train a Tier 2 classifier that is called meta-classifier. If a specific classifier incorrectly learned a particular region of the feature space, and hence mis-classifies instances coming from that region, then the Tier 2 classifier learn this behavior, and beside the learned behaviors of other classifiers, it can correct such improper training. Stacking model is shown in figure 2.

Figure.2. Stacking Model

C. Bagging
In the bagging algorithm, each member of the ensemble is made from a special training dataset, and therefore the predictions combined either by uniform averaging or voting over class labels. Each dataset is generated by sampling from the entire N data example, choosing N items uniformly at random with replacement. The name bagging is an acronym derived from Bootstrap AGGregatING and each sample is known as a bootstrap. Since a bootstrap samples N items uniformly at random with replacement, the probability of any individual data item not being selected is

\[ p = (1 - \frac{1}{N})^N \]  

1. Random forest: The decision tree forms the base classifier in a Random forest. As the name suggests randomization is done in two ways in constructing random forests. One is using random sampling for drawing samples and the second is randomly selecting attributes or features for generating decision trees. Decision trees are a good candidate for classification where a large amount of data is to be classified in terms of accuracy and correctness[3]. The random forest techniques are applied to test data set where the trees are constructed while the resultant individuals are combined to predict the class label. To classify a large amount of data with a single classifier is not worthy and may lead to less accuracy in the result. Hence this is often used in many applications where a large amount of data to be classified with a decision tree classification algorithm. Random forests are easy to comprehend both for computer professionals and end-users without a statistical background. The random forest does not require any cross verification and it is not over-fitting.

Algorithm 1
Input: Training set S= (x1,y1),(x2,y2),(xn,yn)
Output: Combined prediction of multiple base classifiers
1) for t=1 to T do
2) Build a dataset St, by sampling N items, randomly with replacement from S.
3) Train a model $h_t$ using $S_t$, and add it to the ensemble
4) end for
5) For a new testing point $(x,y)$
6) If model outputs are continuous, combine them by averaging
7) If model outputs are class labels, combine them by voting

D. Boosting
Boosting is the most famous ensemble model that’s generally less biased than the weak learners that compose it. Unlike bagging method boosting mainly aims at reducing variance, boosting may be a technique that consists in fitting sequentially multiple weak learners in a very adaptive way each model in the sequence is fitted giving more importance to observations within the dataset that were badly handled by the previous models in the sequence. Boosting, like bagging, are often used for regression also as for classification problems. Being mainly focused on reducing bias, the bottom models are often considered for boosting are models with a low variance but high bias. For example, if use trees as base models it will choose most of the time shallow decision trees with only a few depths. Another important reason that motivates the utilization of low variance but high bias models as weak learners for enhancing is that these models are generally less computationally expensive to suit. Indeed, as computations to fit the various models can’t be wiped out parallel (unlike bagging), It could become too expensive to fit sequentially several complex models. Once the weak learners have been chosen, we still need to define how they will be sequentially fitted (what information from previous models do we take into account when fitting the current model) and how they will be aggregated. The main steps involved in the boosting algorithm are given in algorithm 2.

AdaBoost: AdaBoost, short for “Adaptive Boosting”, is that the first practical boosting algorithm proposed by Freund and Schapire in 1996[4]. It focuses on classification problems and aims to convert a group of weak classifiers into a powerful one. The final equation for classification can be represented as

$$F(x) = \text{sign}(Xmfm(x))$$ (2)

Where $f_m$ stands for the $m^{th}$ weak classifier and $\theta_m$ is the corresponding weight. It is exactly the weighted combination of $M$ weak classifiers. AdaBoost is the most well-known dependent algorithm for building an ensemble model. Multiple sequential models are created, each correcting the errors from the last model. AdaBoost assigns weights to the observations which are incorrectly predicted and the subsequent model works to predict these values correctly.

Algorithm 2
Input: training set $(x1,y1), (x2,y2), (xm,ym)$
Output: Combined prediction of multiple base classifiers
1) A subset is created from the original dataset.
2) Initially, all data points are given equal weights.
3) A base model is created on this subset.
4) This model is used to make predictions on the whole dataset.
5) Errors are calculated using the actual values and predicted values.
6) Errors are given higher weights.
7) Another model is created and predictions are made on the dataset.
8) Similarly, multiple models are created, each correcting the errors of previous ones.
9) The final model is the majority voting of all the weak learners.

1. Gradient Boosting Machines: In Gradient boosting machines (GBM) the training of each base learner is dependent on others that have already been trained. The main difference between GBM and other techniques is that in GBM optimization is applied in the function space. It includes a learning procedure in which the goal is to construct the base learners so that they are maximally correlated with the negative gradient of the loss function, related to the whole ensemble[4]. More specifically, in GBM a sequence of regression trees is computed, where each successive tree predicts the pseudo-residuals of the preceding trees given an arbitrary differentiable loss function. An arbitrary loss function requires the specification of the loss function by the user, in addition to the function that calculates the corresponding negative gradient. Predictions are aggregated in an additive manner in which each added model is trained so it will minimize the loss function. It is important to note that a GBM model usually has many shallow trees, as opposed to random forest which has fewer (but deeper) trees. Choosing the right number of trees (i.e. the number of iterations) is very important when training a gradient boosting model. Setting it too high can lead to over-fitting, while setting it too low may result in under-fitting. The selection of the most suitable number of iterations is usually done by using a validation set to evaluate the overall predictive performance. Over-fitting can be reduced by applying a stochastic gradient boosting method in which trees are consecutively trained with small subsets, sampled from the original dataset.

2. Gradient Boosting Decision Tree: Gradient Boosting Decision Tree (GBDT) is an immortal model in machine learning. In fact:

$$\text{GBDT} = \text{Gradient Boosting} + \text{Decision Tree. (3)}$$

That is, if each sub-model in Gradient Boosting is a Decision Tree, such a model is GBDT[5]. It has the advantages of a good training effect and not easily over-fitting. The tools of GBDT includes XGBoost, Pgbtr, Sklearn, Light GBM, and others. GBDT is widely used in industry and is usually used for tasks such as click rate prediction, searching ranking, and other tasks. GBDT is a deadly weapon in various data mining competitions and according to statistics, more than half of the championship programs on the Kaggle contest are based on GBDT. Two popular GBDT algorithms are XGBoost and LightGBM.

3. XGBoost: Based on the GBDT algorithm and the above definition, the XGBoost algorithm introduces the regular function $\Omega(f)$ to control the over-fitting of the model[6]. Therefore, the target function of XGBoost can be expressed in (4) $X$ is information input, all variables used for prediction, Y is the final variables need to be predicted, $Fm(X)$ is m weak classifiers were added to the final predictable model with a certain weight after m times iteration, $(Fm(X),Y)$ is the loss function. The difference between the final predicted variables after m times iteration and the actual ones $\Omega(f)$ is the regularization function in the final prediction model after all iterations. This is a solution commonly used in machine learning at present. A definite iteration target is introduced to the computer. And then constant term C can be selectively omitted according to the specific problems studied. The introduction of regular items can effectively prevent over-fitting phenomenon. The iterative learning process of the algorithm is shown as the iteration continues, the loss function gradually decreases, but the regular terms continue to expand.
A truly good model is a reasonable balance between the accuracy and simplicity of the algorithm. Then expand the objective function into Taylor second order series. The following are some parameter settings for the XGBoost model.

- **objective**: Define the learning tasks and learning
- **log loss**: Negative logarithmic likelihood
- **error**: Binary classification error rate. For the prediction, the estimation will take predicted values greater than 0.5 as positive examples and the rest as negative examples
- **mirror**: Multi-class classification error rate. It is calculated as error cases/all cases.
- **mlogloss**: Multiple recording
- **auc**: The area below the ranking evaluation curve
- **ndcg**: Standardized discount cumulative returns
- **map**: Average accuracy
- **min child weight**: [default=1], The sum weight of smallest sample in a child node.
- **eta**: The contraction step used in the update process to prevent over-fitting. After each calculation, the algorithm will directly gain the weight of the new feature. Eta makes the lifting of the calculation process more conservative by reducing the weight of the features. The range of values is [0, 1].
  - **max depth**: The maximum depth of a tree
- **log loss**: Negative logarithmic likelihood
- **error**: Binary classification error rate. For the prediction, the estimation will take predicted values greater than 0.5 as positive examples and the rest as negative examples
- **mirror**: Multi-class classification error rate. It is calculated as error cases/all cases.
- **mlogloss**: Multiple recording
- **auc**: The area below the ranking evaluation curve
- **ndcg**: Standardized discount cumulative returns
- **map**: Average accuracy
- **min child weight**: [default=1], The sum weight of smallest sample in a child node.
- **eta**: The contraction step used in the update process to prevent over-fitting. After each calculation, the algorithm will directly gain the weight of the new feature. Eta makes the lifting of the calculation process more conservative by reducing the weight of the features. The range of values is [0, 1].
  - **max depth**: The maximum depth of a tree

As an improvement of GBDT, the XGBoost algorithm has many excellent properties. The biggest advantages are as follows:

- The regularization step is introduced to reduce the over-fitting phenomenon.
- Parallel processing improves the speed of operation
- XGBoost allows users to define user-defined optimization goals and evaluation criteria which increase the flexibility
- XGBoost contains rules for handling missing values
- XGBoost has special pruning steps to control the complexity of the decision tree
- XGBoost contains cross-validation to obtain the optimal number of iterations conveniently
- XGBoost can continue training based on existing models.

4. **LightGBM**: LightGBM (Light Gradient Boosting Machine) is a quick and efficient GBDT algorithm in an open-source promotion framework that was designed by the Microsoft MSRA in 2016. This algorithm is used in sorting, classification, regression, and many other machine learning tasks and supports efficient parallel training. LightGBM is a type of GBDT, which being proposed is to solve the problems encountered by GBDT in mass data so that GBDT can be used in practice better and faster. The decision tree sub-model in LightGBM splits the nodes by leaf splitting. Therefore, it computational costs are relatively small, comparing with XGBoost. It is the reason why we have to control the depth of the tree and minimum data of each leaf node to avoid the occurrence of the fitting phenomenon when using this method [5]. LightGBM selects the decision tree algorithm based on a histogram that divides the eigenvalues into many small ‘buckets’ and then searches for divisions on these ‘buckets’ that can reduce storage and computational costs. Also, the processing of category features makes LightGBM a better improvement under certain data. LightGBM can be divided into three categories that are feature parallelism, data parallelism and voting parallelism. Feature parallelism is used concurrently in scenes with many features, data parallelism is applied in scenes with large amounts of data, and voting parallelism is applied in situations where there are many features and votes. LightGBM algorithm has been widely used in the field of big data machine learning since it was released in 2016. Together with XGBoost, it is regarded as a powerful tool in machine learning. LightGBM is more efficient and accurate than other existing boosting tools. LightGBM is faster, requires less memory, and is more accurate than XGBoost. Therefore, the advantages of this algorithm are fast training speed, low memory consumption, good model precision, parallel learning support and fast when dealing with big data. LightGBM achieves algorithm control and optimization through the following main parameters

- **learning rate**: Learning rate
- **max depth**: Maximum learning depth. We limit the maximum depth of the tree model and control for over-fitting. When max depth is < 0, there is no limit on the learning depth.
- **min data**: The minimum number of data in a leaf that can be used to control the fitting phenomenon
- **feature fraction**: The proportion of the selected feature to the total number of features, ranging from 0 to 1. When feature fraction is 0, the Light-GBM randomly selects partial features at each iteration, and have fraction is employed to regulate the ratio of the total number of characteristics. This parameter can be used to accelerate the training speed and to control for over-fitting.
- **bagging fraction**: The ratio of the selected data to the total data, ranging from 0 to 1. It’s almost like the feature fraction, but is randomly and not repeatedly selected and must be greater than 0. This parameter can be used to accelerate the training speed and control for the fitting phenomenon. Control Parameters
  - **min data in leaf**: It is the minimum number of the records a leaf may have. The default value is 20, optimum value.
  - **early stopping round**: This parameter can assist you to speed up your analysis.
  - **lambda**: lambda specifies regularization. The typical value ranges from 0 to 1.
  - **min gain to split**: This parameter will describe the minimum gain to form a split. It can want to control the number of useful splits in trees.
  - **max cat group**: When the amount of category is large, finding the split point thereon is easily over-fitting. So LightGBM merges them into ‘max cat group’ groups and finds the split points on the group boundaries.
  - **Num-boost round**: Number of boosting iterations
  - **Learning rate**: This determines the impact of every tree on
the outcome. GBM works by starting with an initial estimate which is updated using the output of every tree. the training parameter controls the magnitude of this alteration within the estimates. Typical values: 0.1, 0.001, 0.003

III. ANALYSIS AND DISCUSSIONS

This section provides extensive analysis of different types of ensemble learning methods that are used in various application recently. Prediction of P2P network loan default based on Light-GBM and XGBoost algorithms according to different high dimensional data cleaning is proposed in [6]. Peer-to-peer (P2P) is an innovative mode of borrowing that is a strong complement to the normal financial industry. The default risk of loans within the platform is strongly and innovatively predicted. And the results of ensemble LightGBM and XGBoost methods are compared. The data used are loan data from the Lending Club platform. These loan information data are allowed to be published on the platform. All data regarding the borrowing project can be observed, and these projects include both defaults and non defaults. The experiment result shows that for the consolidated data set, the LightGBM method performs better than the XGBoost in the credit default forecasting project. And the classification prediction results of LightGBM algorithm based on multi-observation data sets are the most accurate. It produce error rate of 19.9% and an accuracy of 80.1%. A hardware-assisted Malware Detection (HMD) approach based on ensemble boosting proposed in [7]. In this work, develop a specialized two stage run time HMD referred as 2SMaRT. In order to extract the Hardware information here used Perf tool available under Linux. Perf provides rich generalized abstractions over hardware specific capabilities. 2SMaRT first classifies applications using a multiclass classification technique into either benign or one of the malware classes (Virus, Rootkit, Backdoor, and Trojan). In the second stage, to have a high detection performance, 2SMaRT deploys a machine learning model that works best for every class of malware. To realize an effective run-time solution that relies on only available HPCs, 2SMaRT is further customized using an ensemble learning technique (AdaBoost) to boost the performance of general malware detectors. The experimental results show that 2SMaRT using ensemble technique with just 4HPCs outperforms state-of-the-art classifiers with 8HPCs by up to 31.25% in terms of detection performance, on average across different classes of malware. A static malware detection method by portable executable analysis and GBDT proposed in [8]. Static malware detection is an important layer during a security suite, which attempts to classify samples as malicious or benign before execution. However, most of the related works incur the scalability issues, for examples, methods using neural networks usually take a lot of training time or use imbalanced datasets which makes validation metrics misleading in reality. GBDT implemented by using Light GBM that speeds up the training pro- cess up to over 20 times by appropriately reducing the feature dimension also it efficiently handling the massive data. The performance of Light GBM compared with malconv model and dataset owners baseline model. The experiment results show that proposed method can achieve up to 99.394% detection rate based on more than 600,000 training and 200,000 testing samples from Endgame Malware BEnchmark for Research (EMBER) dataset [18]. Light GBM out perform that of other two methods. Different classifiers are compared for implementing SMOTE on phishing data to enhance cyber security in [9]. SMOTE is Synthetic Minority Over-sampling Technique to deal with the class imbalance. Phishing is a form of cyber security threat where the criminal tries to realize access to users personal information by infecting their system using malware and viruses. Here use dataset from UCI Machine Learning Repository. This multi-variate dataset contains nine attributes which can distinguish phishing websites from legitimate ones. In this work, investigate how prediction accuracy varies in a balanced dataset against an imbalanced one also different classifiers are compared. SMOTE [19] is applied to balance the dataset. XGBoost, Random Forest and SVM have been applied on the phishing dataset and evaluate their performance with SMOTE and without SMOTE. Results show much higher accuracy rates with SMOTE application. The highest jump in accuracy has been recorded in XGBoost from 89.87% to 97.17%. XGBoost is an effective tool for classification. Unlike bagging algorithms, which work by creating decision trees on bootstrap aggregation, boosting method performs multiple iterations to try and maximize accuracy. Here this method can produce better results that random forest and base classifier SVM. Forecasting crypto currency price trend in crypto currency market using Light GBM is proposed in [10]. Forecast the price trend two category falling, or not falling of crypto currency market. In order to utilize market information combine the daily data of 42 kinds of primary crypto currencies with key economic indicators. The dataset is trading data from January 1, 2018, to June 30. For LightGBM, most important hyper-parameters in the whole selection and optimization process are feature fraction and bagging fraction, which largely determines the randomness of the model. Compare Light GBM with Random forest and SVM and results show that LightGBM and Random forest produces almost same accuracy but Light GBM model outperforms SVM and Ran-dom forest in robustness, which would make it an effective forecasting model when managing a large number of data instances and a large number of features simultaneously. Ensemble of multiple classifiers implemented for automatic multimodal brain tumor segmentation in [11]. This work investigate the employment of ensemble methods for improving the accuracy of multimodal brain tumor segmentation. Four different ensemble methods are evaluated, Adaboost, Bagging, Stacking and Voting on MRI dataset. First, an ensemble classifier composed of 20 decision stump classifiers as the base classifier is trained on the train- ing dataset Adaboost method. Second, 20 random forest classifiers are used as base classifiers and trained on the same training set using bagging method. Finally, the 20 classifiers decisions are combined using majority voting. All four ensemble methods are then assessed on the test dataset. Experimental results confirm the performance improvement produced by the ensemble methods over those of 20 different individual classifiers. Majority voting based ensemble method performed the most effective among the four ensemble methods. Ensemble learning on visual and textual data for social image emotion classification proposed in [12]. For classification task here propose to adopt ensemble learning approaches based on the Bayesian model averaging method, that combine five state-of-the-art classifiers. NB, SVM, Bayesian networks (BN), Decision trees, Neural network (NN), KNN are used as base classifiers. Experiments demonstrated that ensemble strategy that combines baseline classifiers considering their marginal prediction capabilities and their reliability is able to further increase the performance of all the classifiers and produces results higher that those achieved in the state of the art of image emotion classification. The performance of machine learning algorithms evaluated for credit card fraud detection
Credit card fraud detection is a particular classification problem due to a very high imbalance in instances of normal and fraudulent transactions as examples. The dataset was collected and analyzed during a research collaboration of Worldline and the Machine Learning Group of ULB (University Libre de Bruxelles) on big data mining and fraud detection. Several popular algorithms in supervised, ensemble and unsupervised categories were evaluated on different metrics. That are NB, Random forest, kNN, Logistic Regression (LR), XGBT, SVM, Neurual network(NN). It is concluded that unsupervised algorithms handle the dataset skewness in better ways and hence perform well overall metrics absolutely and relatively. Out of all chosen methods NN, XGBT, LR, ensemble model gave near-perfect results. Diagnostic Odd Ratio (DOR) is also another balanced metric where higher values are interpreted as better results. XGBT gave the best DOR while all hybrid models gave similar results. A lightweight malware classification system constructed in[14] that combined multiple categories of features and ensemble learning methods to improve the final results. The method doesn’t need any computationally expensive unpacking process and extracts the features from the asm file and bytes file directly. The system employed two efficient ensemble learning models namely XGBoost and Extra tree classifier and also combined stacking method to construct the final classifier. Experimented on the system with the dataset provided by Microsoft hosted in Kaggle for malware classification competition[20], and the final results show that the ensemble method could classify malware to their family effectively and efficiently with the accuracy of 0.997. Improved network traffic classification using ensemble learning is proposed in [15]. Here presents a comparative analysis among meta learning approaches and individual classifiers to classify network traffic. Investigate and evaluate a range of meta-learning techniques including Voting, Stacking, Bagging and Boosting then propose a new experimental analysis of different meta-learning techniques and compare them with their own base classifiers when used individually. The base classifiers adopted in this work were, decision tree and kNN. Finally, considering the emerging popularity of neural networks and analyze this scenario using the MLP classifier. Regarding the three base classifiers, kNN presented the best results, while MLP the worst. Overall, see that Voting, AdaBoost and Bagging were able to improve the performance of its base classifiers. Stacking, on the other hand was not. AdaBoost uses unplaced sampling to create subsets of training data, used to train classifiers sequentially. Then, new classifiers are trained focusing on the error of previous classifiers. Because this base classifiers have different errors, there is a gain of data and therefore, a performance improvement. With the Bagging approach, on the other hand, diversity is obtained from multiple replicas of the training set. The experiments were performed with data provided by the UCI Machine Learning Repository. The best performance was obtained by an ensemble technique (Bagging), which obtained an accuracy of 99.9%. Overall, the ensemble learners were able to reduce the number of false positives (Bagging achieved 0.36 false positives out of 4000 instances), except for Stacking, due to its few gain of information on its first level. Acoustic domain classification and recognition through ensemble-based the multilevel classification proposed in [16]. To make the best use of speech recognition, This work proposes an approach for recognition of the acoustic domain using ensemble-based 3-level architecture rather than one classifier for training and testing. To validate the performance of the proposed multilevel classifier used TIMIT acoustic-phonetic continuous speech corpus, LibriSpeech ASR corpus, and own created dataset. The own dataset is composed of sentences based on various real-life conversations from standard sources of information like newspapers, news channels, social networking sites, journal articles, etc Here used kNN, Support Vector classifier (SVC), Random Forest, AdaBoost, GBM as classifiers. It is estimated the predictions of various classifiers and then selects a set of three classifiers such that, any of the three classifiers must contain the target predictions and finally, these predictions are wont to train another random forest classifier. It yields the ultimate classification results of the test data set. Experimental results indicate that the proposed method has consistent performance though data size is increased with acceptable accuracy i.e. 76.36%. Overcome the class imbalance problem in breast cancer malignancy grading with ensembles of classifiers in[17]. The most common approach to address the class imbalance with ensembles is the introduction of a data preprocessing step which balances the data distribution. These methods are more versatile than those based on cost-sensitive ensembles because the setting of the costs is avoided. The preprocessing based ensemble methods can be further di-vided into Bagging, Boosting, and Hybrid-based ensembles, depending on the ensemble learning algorithm in which they are based. The data were collected at the Department of Pathology and Oncological Cytology, Medical University of Wroclaw, Poland. Currently, the dataset consists of 341 images. Here used Evolutionary Under-sampling Boost (EUSBoost) for improving the accuracy and diversity of the base classifiers. This allowed performing a guided under-sampling of the majority class, selecting the most important objects for the classifier training step. Produce accuracy equal to 96.38%. These excellent results allowed EUSBoost to outperform 13 state-of-the-art ensemble classifiers. Comparison of different ensemble classifiers used in the various application is shown in Table 1, evaluating the performance of classifier by using the type of application, specification of the dataset and understood which classifier is performed better. From all of the above discussions it is analyzed that ensemble learning perform better than that of base classifiers. It produces more accurate prediction in [6], [7], [8], [9], [10], [11], [12], [14], [15], [16]. Efficiently handle the imbalanced dataset in [17], [16],[13]. There are different types of ensemble learning and they are differentiated in the way it learns a problem. Boosting, Bagging, Voting, Stacking is the widely used ensemble technique. Here compare the performance of these ensemble methods. While comparing different types of ensemble methods in [7],[9],[10],[13] boosting produces more accurate results to perform better than that of bagging, stacking, voting algorithms. In [6], [8], [13], [14] boosting efficiently handle the massive data, reduces the training time, scalability, and produce good predictive performance. It is analyzed that boosting is efficient for classification problems. Different boosting techniques are compared here [6],[9],[10],[13] that are LightGBM, XGBoost, AdaBoost, GBM from which it is analyzed that Light GBM, XGBoost widely used boosting tools in many applications and produces accurate results. While comparing with Light GBM and XGBoost Light GBM is Perform better than XGBoost[6]. It is analyzed that among ensemble learning, the boosting algorithm performs better for classification problems. Experimental results show that the LightGBM is more efficient and accurate than other existing boosting tools. LightGBM is faster, requires less memory, and is more accurate than XGBoost. LightGBM is efficient for binary classification.
IV. CONCLUSION

This survey provides a comprehensive review of ensemble methods in machine learning classification. There are different types of base classifiers like SVM, kNN, NB, MLP, Decision tree. The output of these base classifiers combined to form ensemble classifiers. The study shows that ensemble classifiers produce more accurate performance than base classifiers output. It decreases the diversity among base classifiers, produces consistent performance even if the data size is increased and decrease classification error that is it over-coming to the limitations of base classifiers. There are different types of ensemble learning like bagging, boosting, stacking, and voting. Among different ensemble classifiers in the majority of cases boosting technique produce better result compared to bagging and stacking. Adaboost, Light-GBM, XGBoost are different types of boosting algorithms. It is observed that LightGBM and XGBoost are the most advanced and widely used boosting tools. LightGBM has several advantages of fast training speed, low memory consumption, good model precision parallel learning support, fast when dealing with big data. XGBoost has advantages of reducing the over- fitting improves the speed of operation, increase the flexibility, contains rules for handling missing values, control the complexity of the decision tree. While comparing LightGBM with XGBoost, LightGBM have leaf wise learning better than level-wise learning in XGBoost then LightGBM is faster, computational costs are relatively small, requires less memory and is more accurate than XGBoost. Therefore from the analysis, it is inferred that LightGBM is an efficient tool for classification.

<table>
<thead>
<tr>
<th>Table.1. Comparison of ensemble classifiers in different applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFERENCES</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Hardware-assisted Malware Detection[7]</td>
</tr>
<tr>
<td>SMOTE implementation on phishing data[9]</td>
</tr>
<tr>
<td>Multimodal BrainTumor Segmentation [11]</td>
</tr>
<tr>
<td>Social image emotion classification [12]</td>
</tr>
<tr>
<td>Credit card fraud detection [13]</td>
</tr>
<tr>
<td>Speech recognition[16]</td>
</tr>
<tr>
<td>Classification of Breast Cancer Malignancy [17]</td>
</tr>
</tbody>
</table>

ACKNOWLEDGEMENT

Hereby, I would like to kindly acknowledge sincere cooperation and valuable instruction of all the staffs of computer science and engineering department. Thanks to all those who helped me in the completion of this work knowingly or unknowingly.

V. REFERENCES


[15]. Possebon, Isadora, et al. ”Improved Network Tra

