DOI: http://dx.doi.org/10.26483/ijarcs.v8i9.4913
ISSN No. 0976-5697
Volume 8, No. 9,November-December 2017
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

ALEAN STACKED ENSEMBLE MODEL(LSEM) TO ENHANCE THE EFFECTIVENESS OF CLASSIFYING DATA WITH HUGE IMBALANCE

Dr. K.Mani
Associate Professor in Computer Science,
Nehru Memorial College, Puthanampatti, Trichy.

N. Elavarasan
Research Scholar in Computer Science,
Nehru Memorial College, Puthanampatti, Trichy.

Abstract: Knowledge discovery and analysis has become one of the major needs of the current information rich world. Effective information identification and prediction requires effective models. Several machine learning models are available for prediction. This paper concentrates on classification, a supervised machine learning model. An effective classifier can enable effective predictions. However, not all input data are perfect to enable highly accurate classification. Several factors such as data imbalance, noise and borderline entries affect the classifiers. This paper proposes a Lean SVM based Ensemble Model (LSEM) that enables effective classification of data without the need for pre-processing. A heterogeneous ensemble is created using Random Forest and One-Class SVM. The requirement of partial training data for SVM makes the model lean, enabling faster training. Experiment is conducted on data with varied imbalance levels and it is identified that the proposed LSEM operates better than state-of-the-art models and ensembles and hence enabling better predictions.

Keywords: Classification; Data Imbalance; Ensembles; Heterogeneous base learners; Random Forest; One-Class SVM.

1. INTRODUCTION

Information is considered to be the most valuable resource in the current data-rich world. The rich data has information hidden within it, which needs to be extracted. Machine learning is the process of analysing, pre-processing and extracting such valuable insights from the available data. Since data as-such is of no use, it is machine learning that makes the data useful and provides comprehensible business insights. Machine learning tasks are usually categorized into supervised and unsupervised learning models. Supervised learning models are trained on data with prediction results whereas unsupervised learning models are trained on raw data without the prediction results. Classification is a supervised machine learning model that analyses input datatocategorize the input records into defined classes. Classifiers have vast usage scope and can be used in several domains such as fraud detection, intrusion detection, sentiment analysis, seizure prediction etc.

Several factors are observed to affect the process of classification. The major factors are identified to be data imbalance, noisy samples and borderline samples. Data imbalance refers to the domination or ample-availability of instances of certain class instances and low availability of other class instances. The dominating class is usually referred to as the majority class, while the class with low representations is referred to as the minority class. The major issue due to data imbalance is that it affects the classifier's performance to a large extent [1]. The mode of operation of a classifier follows the training-test model. The classifier is initially trained on the training instances and its performance is measured using the test set. The classifier gets most of its training from the majority class instances and very less training from the minority class instances. Imbalanced data does not have sufficient minority class instances to train the classifier, hence the classifier's training is biased leading to unreliable predictions. Noise in data refers to anomalous entries of one class occurring in the safe zones of another class, while borderline entries refers to multiple class entries occurring together leading to definite distinction of class boundaries. Though occurrence of noise and borderline entries also hinders the process of classification, data imbalance is the major issue that sometimes causes noisy and borderline entries. Thus, this paper concentrates on data imbalance and techniques to overcome the shortcoming occurring due to imbalance.

Ensemble learning is the process of utilizing multiple learning models to obtain a predictive performance that is better than any of the single constituent algorithms [14-16]. Ensembles can be categorized into four major categories viz., bagging, boosting, stacking and best-of-n models. Several other variations of these ensembles have also been proposed in recent works, however, the above four categories are the major and mostly used ensemble models.

Bagging model creates multiple bags, with each bag being a single base learner. The base data is sampled with replacement and passed to the ensemble model for training. Each bag learns with only the data assigned to it. Final predictions are made by aggregating the predictions made from all the bags. Boosting uses a single base learner and iteratively improves its solution by incorporating the error factor during the subsequent predictions. Best-of-n model trains several learners and finally chooses the model that provides the best prediction. Stacking involves training base learners and combining their results with a custom combiner. Custom combiner is developed based on the problem requirements. Bagging, stacking and best-of-n ensembles can be both homogeneous and heterogeneous, while boosting uses a single homogeneous base learner. The proposed lean ensemble model uses stacking ensemble as the ensemble architecture model.

Several ensemble models developed specifically for the process of classification has been observed in literature. A bagging based ensemble for network intrusion detection was proposed by Perez *et al.*, in [2]. This model has been specifically designed to identify masquerade detection. The TPMiner has been designed as a one-class classifier, hence specific attack based training is made possible. A similar

botnet analysis ensemble model was proposed in [7]. However, on data with imbalance, thesemodel suffer from data insufficiency. A model that detects the indecision region for effective dynamic ensemble selection was proposed by Oliveira et.al.[3]. This proposes a dynamic selection framework for binary classification problems. This model aids in the detection of noisy samples that form a part of imbalanced data. A semi-supervised heterogeneous ensemble classifier, Multi-train was proposed by Gu et.al. in [4]. This model generates several heterogeneous classifiers that use varied classifier models and data features to perform predictions. A web service classifier ensemble that utilizes majority votes for prediction was proposed by Qamar*et al.*,in [5]. This model utilizes heterogeneous ensembles for the prediction process. A voting based ensemble that utilizes voting accuracy for reducing the errors was proposed by Bharadwaj*et al.*, in [6].

2. ISSUES DUE TO DATA IMBALANCE

Data is said to be imbalanced if one of its classes have a huge dominance over the other classes in a dataset. Imbalance ratio is defined as the ratio between the minority classes and the majority classes.

$$ImbalanceRatio = \frac{\#\, majinstances}{\#\min instances}$$

Level of overlook experienced by the minor class is proportional to the imbalance level contained in the dataset. In some multi-class datasets, minor classes containing very few entries tend to get totally ignored. Class imbalance provides overtraining for the major class, while the minor class gets undertrained. Since most classifiers tend to implicitly consider the dataset to be balanced, the accuracy and reliability of predictions is highly affected when it comes to imbalanced data.

Most of the classifiers tend to assume that the data is balanced during the classification process. Hence they tend to assign distinct cost to the training samples [8]. While it becomes beneficial for the majority classes, the minority classes suffer from insufficient training and sometimes even get ignored or masked by the majority classes. In order to neutralize this effect, data balancing techniques [9] have been proposed. These techniques are classified into two major categories. The first category deals with modifying the existing algorithms to handle imbalanced data [8], while the next category deals with incorporating data pre-processing techniques (oversampling or under-sampling) to equalize the imbalance [10-14].

Metrics being used for analysis of classifiers are presented in Table 1. The table states the metric, provides the formula for calculating it, behaviour of the metric towards a classifier that makes a classifier significant and a brief description of the metric.

Table 1: Metric Analysis

Metric	**Formula**
True Positive Rate (TPR)/ Sensitivity/ Recall	$\frac{TP}{TP + FN}$
True Negative Rate/ Specificity (TNR)	$\frac{TN}{FP + TN}$
False Positive Rate (FPR)	$\frac{FP}{FP + TN}$
False Negative Rate (FNR)	$\frac{FN}{FN + TP}$
Precision/ Positive Prediction Rate (PPR)	$\frac{TP}{TP + FP}$
Negative Prediction Rate (NPR)	$\frac{TN}{FN + TN}$
F-Measure	$\frac{2 * Precision * Recall}{Precision + Recall}$
Accuracy	$\frac{(TP + TN) * 100}{TP + FP + TN + FN}$
Area Under Curve (AUC)	$\frac{1 + TPR - FPR}{2}$

Mathew's Correlation Coefficient (MCC)	$\frac{TP * TN - FP * F}{(TP + FP) * (TP + FN) * (FP +}$

3. STACKING BASED ENSEMBLES

Stacking also referred to as a stacked generalization involves training several models on the data and the final result is obtained by using a custom combiner. Logistic regression is the simplest and the mostly used combiner model, however, any arbitrary combiner can be used for this process. Stacking has been observed to provide improved performance levels, better than any of the single base learners [17]. The major advantage of creating stacking ensembles is that they effectively support heterogeneity in the base learners. Since the combiner can be customized, it is possible to incorporate heterogeneity in the base learners and provide appropriate combiner rules to handle the varied results. Although several algorithms exist in literature, not all algorithms exhibit high performance levels in all the datasets. Each algorithm has its own shortcomings. Heterogeneity in base learners can help overcome these shortcomings. The drawbacks of one model can be compensated by another model. Combining the results can provide effective predictions. This is not possible in homogeneous models, as the error is amplified on each application of the rather than getting reduced.

Stacking has been a recently explored domain. The model has been explored for both classification and regression. The initial work of Breiman [18] was based on creating a regression model. A stacked generalization model by Ozay*et al.,* [19] proposes a fuzzy generalization model. A density estimator combination model using stacking was proposed by Smyth *et al.,* in [20]. Stacking has also been used as an error estimator in several models [21, 22]. A feature weighing stacking based model proposed by Sill *et al.,* [23] also utilizes a variant of the stacking approach. Although several models exist in literature, stacking has been used as a homogeneous model. Heterogeneity, although can be imposed, has not been modelled. The proposed Lean Ensemble (LSEM) creates a heterogeneous ensemble model with a heuristic combiner to operate on imbalanced data.

4. PROPOSED LEAN ENSEMBLE MODEL (LSEM)

Handling imbalance in data has become one of the major requirements of the current classification models. However, imbalance handling is usually performed using external pre-processing modules such as feature selection [24], oversampling and under-sampling. The major downside of using such modules is that, it either duplicates data leading to improper distribution of significant levels, leading to overtraining on the same data or eliminates data leading to loss of valuable information. The proposed Lean SVM based Ensemble Model (LSEM) has been developed such that the base data set remains undisturbed. Significance is imposed in the construction of the model itself. Hence the requirement of pre-processing just for the sake of balancing the data has been eliminated.

The Lean SVM based Ensemble Model (LSEM) is based on two base learners; Random Forest and One-Class SVM. The architecture is created in two levels, with the first level pre-processing is performed by the Random Forest ensemble and the second level processing is performed by one-class SVM. A formal description of the problem are as follows.

Let $D=\{(x_1,C_1),(x_2,C_2),\ldots,(x_n,C_n)\}$, where $X=\{x_1,x_2\ldots x_n\}$ are the data points and $C=\{C_1, C_2,\ldots,C_n\}$ are set of classes. The proposed model operates on binary classifiers, hence $C \in \{0,1\}$.Consider the data for prediction be $TD=\{tx_1,tx_2,\ldots,tx_m\}$. The problem is to predict appropriate classes for $tx_i, \forall i = 1, \ldots, m$

Let *RF(D) and OCSVM(D) b*e the functions which represent of Random Forest and one-class SVM respectively. Then

$$T_{RF} = RF(D)$$
$$Res = T_{RF}(TD)$$

Where T_{RF} is the trained model of Random Forest and *Res* is the prediction result obtained from Random Forest on applying the test data *TD.*

$$Res = \{C_1, C_2, \ldots, C_m\}$$
$$C_i \in \{0,1\} \forall tx_i$$

The first level result R_{L1} represents positively predicted results from *Res*, represented by

$$R_{L1} = \{C_i | Res(tx_i) = 1\} \forall i, 1 \leq i \leq m$$

This forms a part of the final results, while the second part of the result is obtained from one-class SVM. One-Class SVM, being a part of the LSEM, incorporates the *Lean* component into the structure. The major advantage of using one-class SVM is that it gets trained on only one class, hence considerably reducing the training time. The test data for one-class SVM is obtained by extracting the data points that are predicted as negative. This is given by

$$TD_{L2} = \{tx_i | Res(tx_i) = 0\} \forall i, 1 \leq i \leq m$$

Similarly, the training data for one-class SVM is obtained by extracting the negative predictions from the global training data (*D*). This is given by

$$D_{SVM} = \{x_i | C_i = 0 \ \wedge x_i \in D\} \forall i, 1 \leq i \leq n$$

The trained one-class SVM model is given by

$$T_{OCSVM} = OCSVM(D_{SVM})$$

The partial test data TD_{SVM} is passed to the trained model and the second level result R_{L2} is obtained.

$$R_{L2} = T_{OCSVM}(TD_{L2})$$

The final predictions $Pred_{LESM}$is obtained from aggregation of the level 1 and the level 2 results [25]. This is given by

$$Pred_{LESM} = R_{L1} \cup R_{L2}$$

5. RESULTS AND DISCUSSION

Datasets have been selected based on varying imbalance levels from low to moderate to high and shown in table 2. Datasets have been obtained from the KEEL repository [26]. The proposed LSEM model is built using Python and results were obtained by applying the datasets on the proposed model. Comparisons are conducted on state-of-the-art existing models (Naïve Bayes and Decision Tree) and ensembles (Random Forest and Gradient Boosted Trees).

Table 2: Dataset Properties

Dataset	Imbalance Ratio
Vehicle0	3.35
E-Coli3	8.6
Cleveland-0_vs_4	12.62
Poker-8_vs_6	85.88

ROC(Receiver Operating Characteristics) and PR(Precision Recall)plots of the proposed model, the state-of-the-art models and ensembles are shown in figures 1-4. Although each data exhibits varied performance levels due to the imbalance levels, several performance based patterns could be observed from the plots. Considering the ROC plots, Naïve Bayes and Decision Trees, being simple machine learning models exhibits reduced performances as the imbalance levels increases and sometimes exhibits fluctuating performances. The ensemble model GBT exhibits moderate performance, while Random Forest and LSEM exhibits high and stable performances shown in table 3.

Table 3: LSEM PERFORMANCE TABLE

Method	CLEVELAND 0_vs_4				E COLI 3				POKER-8_vs_6				VEHICLE 0			
	ROC PLOT		PR PLOT		ROC PLOT		PR PLOT		ROC PLOT		PR PLOT		ROC PLOT		PR PLOT	
	TPR	FPR	TPR	FPR	TPR	FPR	Precision	Recall	TPR	FPR	Precision	Recall	TPR	FPR	Precision	Recall
GBT	0.55	0.2	0.4	0.12	0.4	0.12	0.45	0.34	1	0	1	0.1	0.94	0.2	0.92	0.9
DT	0.8	0.1	0.5	0.12	0.5	0.12	0.5	0.53	0.8	0	0.9	0.7	0.9	0.2	0.9	0.8
NB	0.9	0.1	0.6	0.2	0.6	0.2	0.4	0.6	0	0	0	0	0.98	0.4	0.5	0.9
RF	0.8	0.1	0.45	0.2	0.45	0.2	0.5	0.5	0.4	0	1	0.5	0.9	0.2	0.9	0.86
LSEM	0.8	0.02	0.45	0.1	0.45	0.1	0.69	0.5	0.4	0	1	0.5	0.9	0.1	0.91	0.8

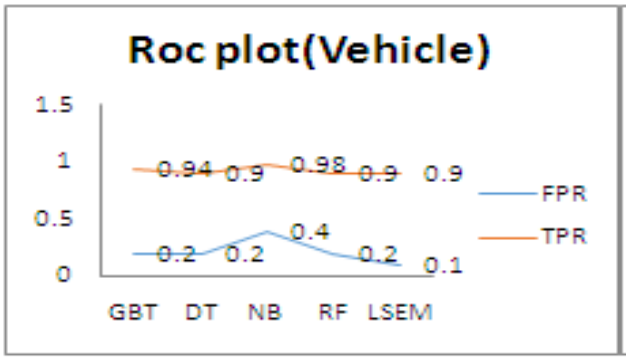


Figure 1(a). ROC Plot (Vehicle)

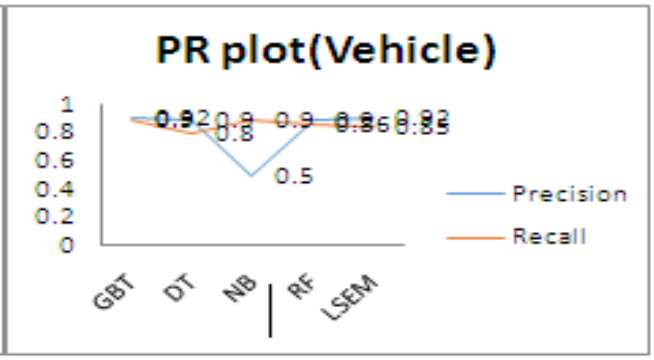


Figure1 (b). PR Plot (Vehicle)

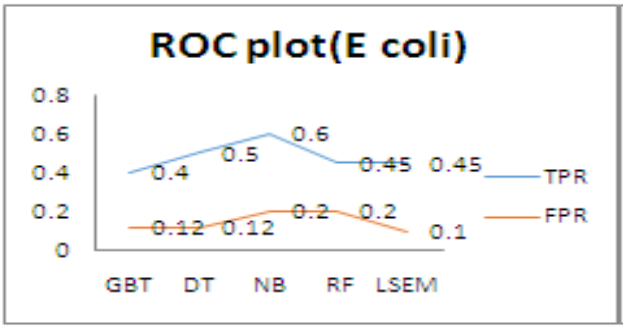


Figure 2 (a). ROC Plot (E-Coli)

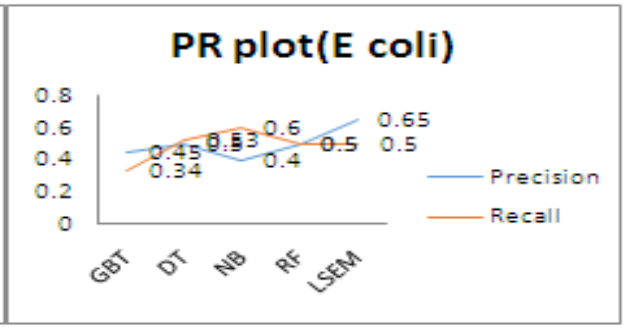
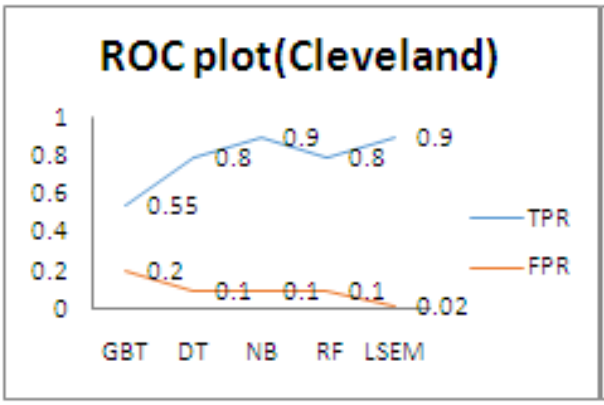
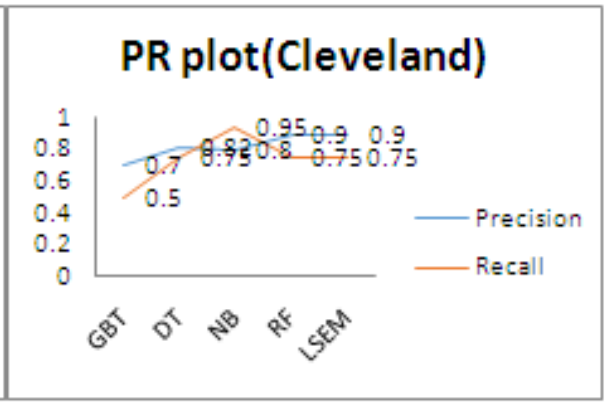


Figure 2 (b). PR Plot (E-Coli)





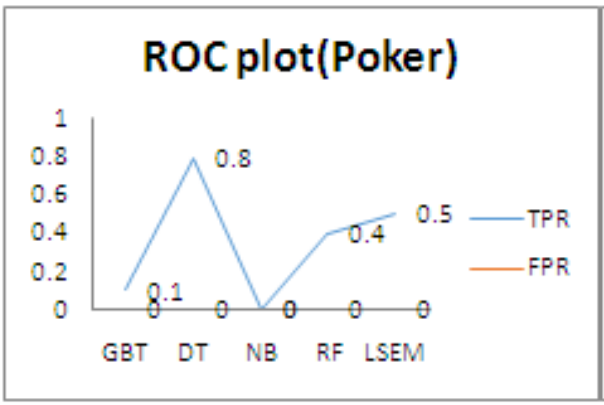


Figure 4 (a). ROC Plot (Poker)

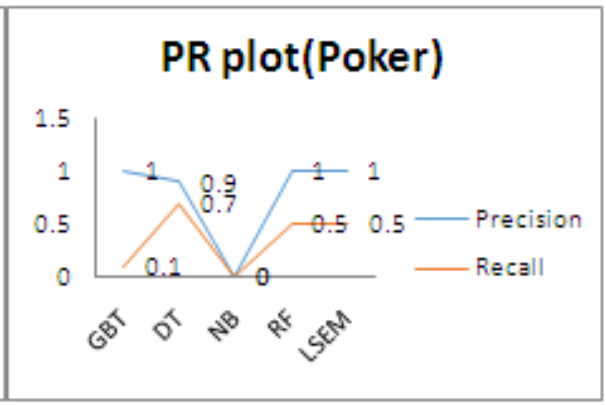


Figure 4 (b). PR Plot (Poker)

Considering the PR plots, Naïve Bayes exhibits very low to moderate precision, exhibiting its inability towards effectively selecting the positive data. A slightly better performance was observed by Decision Trees, while ensembles GBT and Random Forest (RF) exhibits moderate to high performances.

Table 4: Performance Comparison

Cleveland-0_vs_4(Dataset)						
Technique	**Accuracy**	**F-Measure**	**TNR**	**FNR**	**MCC**	**Prediction Levels**
Decision Tree	0.907	0.8	0.951	0.230	0.740	0.860
GradientBoosting	0.851	0.636	0.951	0.461	0.561	0.744
Random Forest	**0.925**	0.833	**0.975**	0.230	0.790	0.872
Naïve Bayes	**0.925**	**0.857**	0.926	**0.076**	**0.811**	**0.924**
LSEM	**0.925**	0.833	**0.975**	0.230	**0.790**	0.872

Ecoli3(Data set)						
Technique	**Accuracy**	**F-Measure**	**TNR**	**FNR**	**MCC**	**Prediction Levels**
DT	0.819	0.514	0.894	0.5	0.403	**0.697**
GBT	0.797	0.424	0.894	0.611	0.304	0.641
RF	0.819	0.484	0.907	0.555	0.378	0.676
NB	0.744	0.478	0.776	**0.388**	0.333	0.693
LSEM	**0.851**	**0.533**	**0.947**	0.555	**0.461**	**0.695**

Poker-8_vs_6(Data set)						
Technique	**Accuracy**	**F-Measure**	**TNR**	**FNR**	**MCC**	**Prediction Levels**
DT	**0.989**	**0.846**	0.997	**0.214**	**0.843**	**0.891**
GBT	0.965	0.133	**1**	0.928	0.262	0.535
RF	**0.978**	0.6	**1**	0.571	0.647	0.714
NB	0.963	0	**1**	1	0	0.5
LSEM	**0.978**	0.6	**1**	0.571	0.647	0.714

Vehicle0(Data set)						
Technique	**Accuracy**	**F-Measure**	**TNR**	**FNR**	**MCC**	**Prediction Levels**
DT	0.945	0.904	0.962	0.095	0.867	0.933
GBT	**0.963**	**0.936**	**0.974**	0.063	**0.911**	**0.955**
RF	0.941	0.896	0.962	0.111	0.855	0.925
NB	0.698	0.645	0.591	**0.0317**	0.509	0.779
LSEM	**0.954**	0.919	**0.974**	0.095	0.888	0.939

Comparison of other performance metrics such as accuracy, F-Measure, TNR, FNR and the balanced measures MCC and Prediction levels are shown in table 3. The best prediction values for each metric and the near best prediction values for the metric (with variance of <0.1) are marked in bold.

Prediction levels is the aggregated metric, which is the average of the positive prediction rates TPR and TNR, and is given by

$$PredictionLevels = \frac{TPR + TNR}{2}$$

It could be observed that in most of the datasets, the proposed LSEM exhibits high prediction levels indicating the efficiency of the proposed model.

6. CONCLUSION

Effective predictions irrespective of the implicit issues in data are the need for the current information age. This paper proposes aheterogeneous stacking ensemble classifier LSEM for effective classification of imbalanced data. Data imbalance is one of the major issues affecting the reliability of the prediction processes. Algorithms tend to consider balanced representations of data in the data set. This leads to biased training when operated upon imbalanced data. The proposed LSEM ensemble aims to provide a solution for this issue by utilizing multiple base learners for processing. Random Forest and One-Class SVM are used as the base learners. The architecture is considered to be lean, as only a part of the training data is provided to the one-class SVM for training and only a part of the testing data (data that is considered to have uncertain predictions) is provided to the

one-class SVM. This reduces the complexity of the ensemble, enabling faster and better predictions. Comparisons are performed with state-of-the-art existing classifiers Decision Trees and Naïve Bayes, and state-of-the-art ensembles Random Forest and GBT. Performance comparison indicates stable and effective performances from LSEM, while highly fluctuating performances from the other models. Limitations of the proposed model is that it exhibits slightly reduced performances on data with high imbalance levels. Future works will deal with incorporating techniques to improve the prediction scalability of the model in terms of varied imbalance levels.

REFERENCES

[1] AkilaSomasundaram and Srinivasulu Reddy U,"Modelling a Stable Classifier for Handling Large Scale Data with Noise and Imbalance", IEEE International Conference on Computational Intelligence in Data Science, 2017.

[2] Medina-Pérez, Miguel Angel*et al.*,"Bagging-TPMiner: a classifier ensemble for masquerader detection based on typical objects", Soft Computing ,557-569,2017.

[3] Oliveira, Dayvid VR, George DC Cavalcanti and Robert Sabourin, "Online pruning of base classifiers for Dynamic Ensemble Selection ", Pattern Recognition 72 ,44-58, 2017.

[4] Gu, Shenkai and Yaochu Jin,"Multi-train: A semi-supervised heterogeneous ensemble classifier ", Neurocomputing202-211, 2017.

[5] QamarandUsman*et al.*,"A Majority Vote Based Classifier Ensemble for Web Service Classification ", Business & Information Systems Engineering249-259,2016.

[6] Bhardwaj, Manju, and VasudhaBhatnagar, "Towards an optimally pruned classifier ensemble", International Journal of Machine Learning and Cybernetics , 699-718, 2015.

[7] Bijalwan, Anchit*et al.*,"Botnet analysis using ensemble classifier ",Perspectives in Science 502-504,2016.

[8] Pazzani and Michael*et al.*, "Reducing misclassification costs.", Proceedings of the Eleventh International Conference on Machine Learning, 1994.

[9] Van Hulse, Jason, TaghiKhoshgoftaar and Amri Napolitano, "Experimental perspectives on learning from imbalanced data ", Proceedings of the 24th international conference on Machine learning ACM, 2007.

[10] Lewis, David D, and Jason Catlett, "Heterogeneous uncertainty sampling for supervised learning ", Proceedings of the eleventh international conference on machine learning, 1994.

[11] Kubat, Miroslav and Stan Matwin, "Addressing the curse of imbalanced training sets: one-sided selection ", ICML Vol. 97, 1997.

[12] Japkowicz and Nathalie, "The class imbalance problem: Significance and strategies ", Proc. of the Int'l Conf. on Artificial Intelligence, 2000.

[13] Ling, Charles X and Chenghui Li, "Data mining for direct marketing: Problems and solutions", KDD. Vol. 98, 1998.

[14] Opitz, David W. and Richard Maclin., "Popular ensemble methods: An empirical study ",Artif. Intell J. Res.(JAIR)11 169-198,1999.

[15] PolikarandRobi, "Ensemble based systems in decision making", IEEE Circuits and systems magazine 21-45,2006.

[16] RokachandLior, "Ensemble-based classifiers", Artificial Intelligence Review ,2010.

[17] Wolpertand David H,"Stacked generalization." Neural networks 41-259,1992.

[18] Schlundand Michael*et al.*,"TanDEM-X elevation model data for canopy height and aboveground biomass retrieval in a tropical peat swamp forest", International Journal of Remote Sensing 5021-5044,2016.

[19] Ozay, Mete, Fatos T and YarmanVural, "A new fuzzy stacked generalization technique and analysis of its performance.",arXiv preprint arXiv:1204.0171 ,2012.

[20] Smyth, Padhraic and David Wolpert, "Linearly combining density estimators via stacking.", Machine Learning 59-83,1999.

[21] Wolpert, David H and William G. Macready, "An efficient method to estimate bagging's generalization error",Machine Learning 41-55,1999.

[22] Clarke and Bertrand, "Comparing Bayes model averaging and stacking when model approximation error cannot be ignored", Journal of Machine Learning Research 683-712,2003.

[23] SillJoseph, et. al., "Feature-weighted linear stacking.",arXiv preprint arXiv:0911.0460, 2009.

[24] Yao and Gangetal.,"Integration of classifier diversity measures for feature selection-based classifier ensemble reduction" Soft Computing 2995-3005,2016.

[25] Omari, Adil, Aníbal R andFigueiras-Vidal,"Post-aggregation of classifier ensembles" ,Information Fusion 26, 96-102,2015.

[26] http://Sci2s.ugr.es/keel/imbalanced.php?order=ir#sub10.