International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 9 (2017), pp. 2267-2281

http://www.ripublication.com

Data Imbalance and Classifiers: Impact and Solutions from a Big Data Perspective

K.Madasamy [1] and M.Ramaswami [2]

[1]Research Scholar (Part-Time), Department of Computer Applications, Madurai Kamaraj University, Madurai-625 021, India.
[2]Associate Professor, Department of Computer Applications, Madurai Kamaraj University, Madurai-625 021, India.

Abstract

Data being generated in real-time environment is prone to inconsistencies like data imbalance and messy data with noise. Hugeness of the data also poses an additional complexity to the prediction mechanisms. Data imbalance is an intrinsic part of the real-time data and hence cannot be overlooked. Experiments reveals that data imbalance and data hugeness do affect the predictors. However, each predictor model exhibits its own advantages, which can be observed in terms of their performance metrics. However, a single predictor with overall high performance is absent. Hence it was concluded that a combination of prediction algorithms would be best suitable for such data. In order to identify the appropriate set of algorithms, a performance analysis was carried out on several standard algorithms. An analysis on ensemble techniques was also carried out to identify the best combiner technique, and boosting and stacking ensembles were shortlisted as they exhibited promising models.

Keywords: Classification, Multi class classification , Data Imbalance, Big data, Ensemble.

1. INTRODUCTION

Massive increase in online activities results in accumulation of huge amount of data. Need to perform analysis on such data can provide deep insights into the nature of

data also become essential for intelligent decision making purpose. Domains such as Customer based Product Prediction, Churn Prediction, Fraud Detection, Network Intrusion Detection, Particle analysis and Prediction such as Higgs Boson, Prediction of eruptions in Volcanoes, Weather Prediction etc. are some of the areas where such analysis can be performed. Prediction is the major form of analysis that operates on the available data to forecast the future occurrences and / or to provide suggestions depending on the past data and it is part of supervised learning apptoach..

Supervised learning refers to training a model based on the available data with defined class distinctions. Such models can then be used to predict future data. Classifier is a category of supervised predictors operating on nominal class labels. The major requirement of any supervised learning technique is that it requires appropriate data for learning. Classifiers work on a learn and predict model. Learning can be hindered by unavailability of sufficient data for training. Data insufficiency occurs due to imbalance. Huge data is another issue as it aggravates this problem manifold [1]. This paper discusses two major issues related to training classifiers; data hugeness and data imbalance. It also discusses the algorithms and performance metrics that best suits to deal imbalanced huge data.

2. DATA IMBALANCE: AN ANALYSIS

Data is set to be imbalanced if one of its classes exhibits a huge dominance over the other classes. In other words one class has very large number of representations, while the other classes exhibit low representation levels. The class with large representatives is called the major class, while the ones with low levels of representation are called minority classes [2]. Imbalance is usually referred in terms of ratio between the number of instances in the majority class to the number of instances in the minority class.

$$Imbalance\ Level = \frac{\#\ Inst_{maj}}{\#\ Inst_{min}} \qquad (1)$$

In a multi-class dataset, majority class represents the class with highest number of instances and minority class refers to the class with the lowest instance levels [9].

The issue of data imbalance appears to be a very rare occurrence in most datasets, however it is a very common occurrence in most of the real-time datasets. As an example, consider the domain of churn prediction. It could be observed that the level of churners would be very low compared to the level of non-churners. Similarly, in the case of network intrusion detection, the rate of occurrence of anomalous packets would be very low compared to the occurrence of normal packets. On further

analysis, it was observed that the minority classes are of significance compared to the major classes. For example, identifying churners have higher significance compared to identifying non-churners, similarly identifying anomalous packets in a network has higher significance compared to identifying normal packets. Further, noisy instances and borderline entries, the intrinsic characteristics of imbalanced data aggravates this situation [6]. Hence effective and appropriate identification of minority classes is of major focus in most of the predictive algorithms [15].

3. IMPACT OF DATA IMBALANCE ON CLASSIFIERS: A PERSPECTIVE FROM BIG DATA

Classifiers are based on supervised learning, i.e., it requires appropriate training prior to the prediction process. Classifiers are generally constructed with the prospect of balanced data. Usually classifiers assume that the data is balanced during the training phase. They require balanced representations of all the classes contained in a dataset to perform effectively [3]. Though a slight skewness in the balance levels is acceptable, an increase in this gap leads to improper classifier training, in turn leading to inappropriate predictions. In the current Big Data scenario, though the ratio between classes (imbalance level) remains the same as regular data, due to the hugeness of the data, actual instance levels tend to increase manifold. This leads to huge representation levels for major classes and low representation levels for minority classes [4]. This leads to a major issue in terms of classifier overtraining for major classes and undertraining in terms of the minority classes [10]. A classifier when presented with Big Data laden with such a property, tends to predict majority classes effectively, however, the minority class prediction levels are lowered to a large extent, hence reducing the reliability levels of the model.

The imbalance acceptance levels of classifiers vary between models [11]. Hence an analysis of algorithms in terms of imbalance levels, data size and their impact on classifier metrics is presented below.

3.1 Algorithms and Metric Analysis

Algorithms are chosen for comparison are based on the category of the algorithm and has shown in table 1. Naïve Bayes is the representative for probability based technique, Decision Tree is a tree based technique, Decision Table is a rule based technique and Logistic Regression is for function based techniques.

Table 1. Categorization of Algorithms

Algorithm	Category
Probability Based Algorithm	
Naïve Bayes	A conditional probability based technique based on Bayes Theorem
Function Based Algorithm	
Logistic Regression	Regression model with a dependent categorical variable
Rule Based Algorithm	
Decision Table	A rule based prediction technique
Tree Based Algorithms	
Decision Tree	Decision making approach using tree like structures

Selected classifiers are applied on the datasets and the results are recorded in the confusion matrix. A sample confusion matrix is shown in table 2.

Table 2: Two Class Confusion Matrix

Actual / *Predicted*	***True***	***False***
True	**TP**	**FP**
False	**FN**	**TN**

Standard classifier metrics used for analysis are shown in table 3. All the metrics are derivable using data from the confusion matrix. The current analysis examines all the metrics in terms of imbalance and data hugeness.

Table 3: Performance Metrics

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	$\frac{2 * Precision * Recall}{Precision + Recall}$
Correct Classification % / Accuracy	$\frac{(TP + TN) * 100}{TP + FP + TN + FN}$
Incorrect Classification %	$\frac{(FP + FN) * 100}{TP + FP + TN + FN}$
Area Under Curve (AUC)	$\frac{1 + TPR - FPR}{2}$
Mathews Correlation Coefficient (MCC)	$\frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + TN) * (TP + FN) * (FP + TN) * (TN + FN)}}$

3.2 Dataset Analysis

Datasets have been selected based on varying characteristics with representations in all categories (table 4), namely, large to moderate size, binary/ multi-classes and with varying imbalance levels ranging from low (0) to very high (164091). KDD CUP 99 dataset was obtained from UCI repository [31], E-coli, Iris and Bank were obtained from KEEL repository [32] and Bank data was obtained from [33].

Table 4: Dataset Properties

Dataset	Class	Size	Imbalance Ratio
KDD CUP 99	Multi-Class (23)	Large (4000000)	**164091**
E-Coli	Multi-Class (8)	Moderate (336)	**71.5**
Iris	Multi-Class (3)	Moderate (150)	**0**
Bank	Binary (2)	Large (345719)	**25.7**
Abalone	Binary (2)	Moderate(4177)	**129.4**

3.3 Results and Discussion

Selected algorithms were applied on the datasets and the results are examined in terms of ROC plot [13, 14], PR plot and the performance metrics shown in table. ROC and PR plots are used to identify the performance of classifiers [12]. WEKA 3.8.1 was used to apply the algorithms and the confusion matrices obtained from WEKA are used for creating the plots.

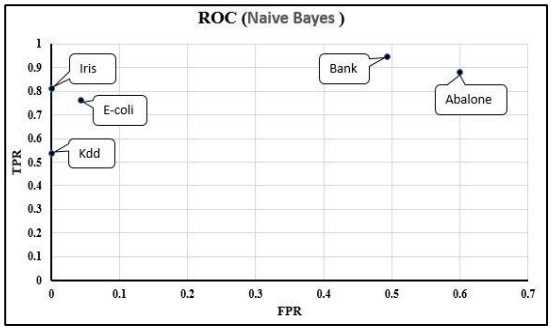

Figure 1: ROC (Naïve Bayes)

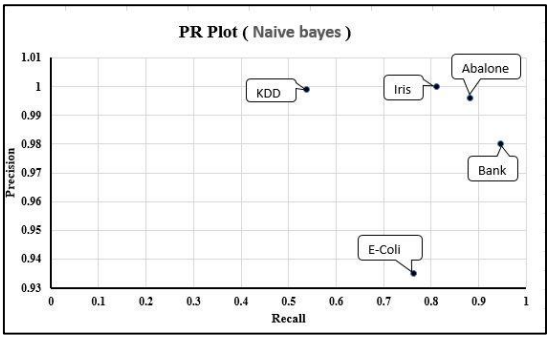

Figure 2: PR Plot (Naïve Bayes)

ROC and PR plots generated by Naïve Bayes algorithm is shown in figures 1 and 2. It could be observed that Iris, being a moderate sized balanced data exhibits the highest performance levels in terms of ROC and PR. Bank data, corresponding to the low imbalance level exhibits moderate performance with high TPR levels but moderate FPR levels. E-coli with moderate data imbalance moderate performance in ROC (TPR ~0.7) and high performance in PR plot. Abalone with high imbalance level exhibits moderate performance levels in ROC (FPR ~0.6) and high performance levels in PR. KDD with very high imbalance levels exhibits low performance levels in ROC (TPR ~0.5) and in PR plots

(Recall ~0.5). Hence it could be generalized that Naïve Bayes handles low to moderate imbalance levels effectively, while high imbalance levels tend to affect the performance.

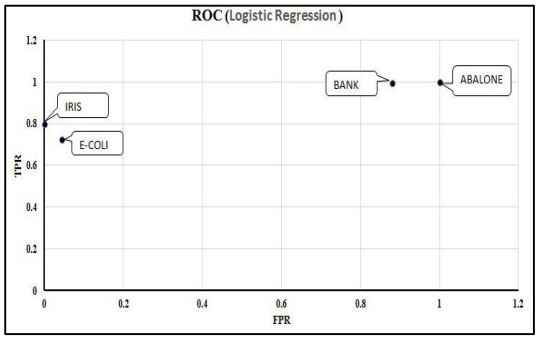

Figure 3: ROC (Logistic Regression)

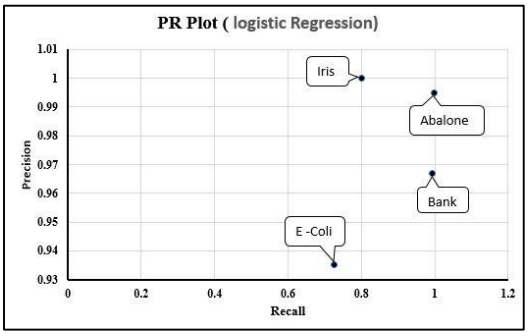

Figure 4: PR Plot (logistic Regression)

ROC and PR plots generated by Multinomial Logistic Regression algorithm is shown in figures 3 and 4. It could be observed that Iris and E-Coli exhibit acceptable performance levels, like their metrics in Naïve Bayes. Though Bank and Abalone datasets (Moderate and High Imbalance) exhibits good performances in terms of precision and recall, however their FPR levels are very high reaching unacceptable levels. Hence it could be generalized that Logistic Regression has very low tolerance towards imbalance levels. Logistic Regression was unable to handle KDD, hence making it unsuitable for predicting huge data.

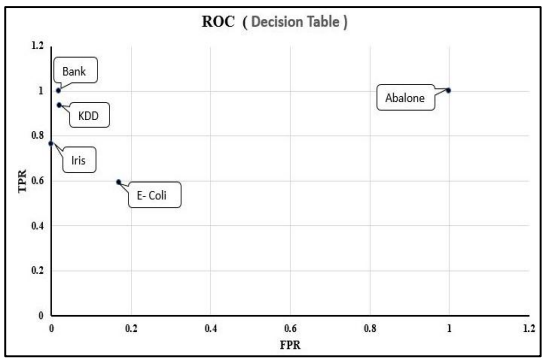

Figure 5: ROC (Decision Table)

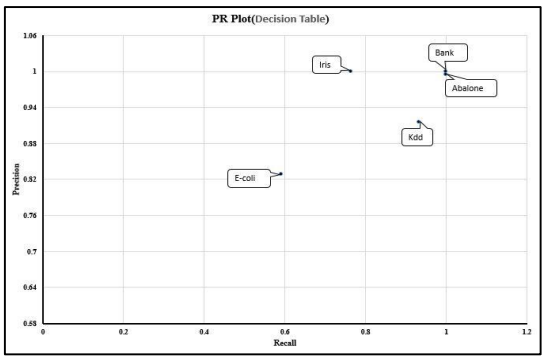

Figure 6: PR Plot (Decision Table)

ROC and PR plots generated by Decision Table algorithm is shown in figures 5 and 6. Iris exhibits good performance levels, while bank with low imbalance level exhibits better performance. However, bank data with a low imbalance level performs much better in terms of both ROC and PR. However, E-coli with moderate imbalance level shows reduced performance, and Abalone with high imbalance levels also exhibit

reduced performance levels. However, KDD with very high imbalance levels exhibits good performance levels. Hence it could be generalized that Decision Table has the efficiency to exhibit high performance levels if sufficient training data is provided for rule building. However, with low data, even low imbalance levels exert negative effects in their performance levels.

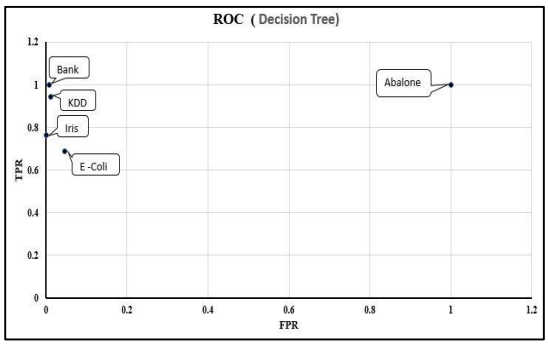

Figure 7: ROC (Decision Tree)

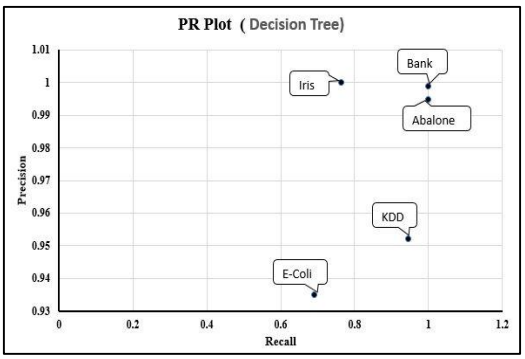

Figure 8: PR (Decision Tree)

ROC and PR plots generated by Decision Tree algorithm is shown in figures 7 and 8. Performance levels of all the datasets can be observed to be similar to the Decision Table. However, E-Coli shows a slight increase in the performance levels (reduced FPR). Hence it could be generalized that Decision Tree has the efficiency to exhibit high performance levels if sufficient training data is provided for tree building. However, with low data levels, even low imbalance levels exert negative effects in their performance levels.

Analysis of other performance metrics are shown from table 5 to table 9. The best entries are marked in bold.

Table 5. Performance Metrics - Iris

	TNR	**FNR**	**F-Measure**	**MCC**	**AUC**	**Accuracy**
Iris						
Naïve Bayes	**1**	**0.1875**	**0.896552**	**0.521372**	**0.90625**	**0.918919**
Logistic Regression	**1**	0.2	0.888889	0.498468	0.9	**0.918919**
Decision Table	**1**	0.235294	0.866667	0.501039	0.882353	0.891892
Decision Tree	**1**	0.235294	0.866667	0.501039	0.882353	0.891892

Table 6. Performance Metrics - Bank

	TNR	FNR	F-Measure	MCC	AUC	Accuracy
Bank						
Naïve Bayes	0.5064	0.052747	0.963512	0.336773	0.726826	0.930915
Logistic Regression	0.118014	0.005107	0.980753	0.225563	0.556454	0.962397
Decision Table	0.980331	0.000397	0.999423	0.966678	0.989967	0.998889
Decision Tree	**0.993131**	**0.000132**	**0.999802**	**0.97629**	**0.9965**	**0.999618**

Table 7. Performance Metrics – E-Coli

	TNR	FNR	F-Measure	MCC	AUC	Accuracy
E-Coli						
Naïve Bayes	**0.956522**	**0.236842**	**0.84058**	**0.483738**	**0.85984**	**0.869048**
Logistic Regression	0.954545	0.275	0.816901	0.464738	0.839773	0.845238
Decision Table	0.828571	0.408163	0.690476	0.326581	0.710204	0.690476
Decision Tree	0.952381	0.309524	0.794521	0.446479	0.821429	0.821429

Table 8. Performance Metrics – Abalone

	TNR	FNR	F-Measure	MCC	AUC	Accuracy
Abalone						
Naïve Bayes	**0.4**	0.118497	0.935583	**0.0599**	**0.640751**	87.9
Logistic Regression	0	0.001927	0.996633	-0.00305	0.499037	0.993289
Decision Table	0	**0**	**0.997597**	0	0.5	**0.995206**
Decision Tree	0	**0**	**0.997597**	0	0.5	**0.995206**

Table 9. Performance Metrics – KDD99

	TNR	FNR	F-Measure	MCC	AUC	Accuracy
KDD						
Naïve Bayes	**0.999951**	0.462008	0.699521	0.247519	0.768971	0.900729
Decision Table	0.979052	0.065038	0.925386	0.412193	0.957007	0.970395
Decision Tree	0.988718	**0.054267**	**0.949255**	**0.415409**	**0.967225**	**0.980375**

Iris with balanced training data exhibits similar performance levels on all categories of algorithms. In terms of other datasets, it was observed that imbalance affects Naïve Bayes and Logistic Regression, while has low or tolerable impact on Decision Table and Decision Tree. However, increase in the data size contributes to the positive side in Decision Tree and Decision Table. Number of prediction classes do not impact the algorithms, until sufficient data are available for training. This proves that each algorithm has its own pros and cons, hence utilizing a single algorithm on huge and highly imbalanced data is not an effective approach. However, combining the algorithms could improve the efficiency levels comparatively to using single techniques.

4. ENSEMBLES: A BRIEF OVERVIEW

Ensemble is a supervised learning technique, which is a combination of learning algorithms. Ensemble is the process of utilizing multiple algorithms to obtain better predictive performance compared to the usage of single learning techniques [18,19]. Hence they are not bound by the number or type of the individual components being used. Machine learning ensembles, in contrast to statistical ensembles utilizes finite models for building classifiers, however, they allow flexible structures to exist in the mechanism [16].

Analysis on standard models exhibits good performance levels on certain parameter metrics and low performance levels on others. Further, imbalance and enormity of data also effects classifier performance [4,5]. These issues were usually counteracted by modifying or fine tuning existing algorithms according to the data distributions [8]. However, such cases become applicable only when a single specific dataset is used. In this paper imbalance in several levels are examined, making data-specific algorithm-fine-tuning impossible. The current work is focused on identifying a generic model that can effectively handle imbalance at all levels without the need for data-specific fine-tuning. As single algorithm approaches are not feasible, this section examines the feasibility of ensemble models and their variants.

4.1 Ensemble Variants and Applicability Levels: A Discussion

Some commonly used ensembles include Bagging, Boosting, Bucket of Models and Stacking. An analysis of ensembles and their varied flavors in terms of performance measures are available in literature. However, ensembles, being a fairly new modelling technique, has not been examined in terms of the nature of data that they are being applied on.

4.1.1 Bagging

Bootstrap Aggregating [20] or Bagging is a machine learning ensemble model with its major focus on increasing the stability and accuracy of the machine learning models. It is referred to as the model averaging approach. Though it is usually applied on tree based models, it can support heterogeneity. Heterogeneous multi model based bootstrap techniques have not yet been proposed. Random Forest [21] is one of the most well-known bagging techniques.

Bagging operates by effectively sampling data and training multiple classifiers on the subsets. The training models are usually multiple instances of the same classifier. Consider m classifier instances and a training set of size n. Bagging generates m new training sets each of size a, where a < n. However, it is maintained that the size of a is usually (1-1/e) or ~63.2% of the unique examples in the training data. Sampling is performed with replacement; hence duplicates can be expected in the training data. Voting is used as the final combination technique.

The major advantage of bagging is that it provides improvements for unstable procedures such as ANN, classification and regression trees. Since only a part of the data is used for training individual models, imbalance can be counteracted, as some trees might receive balanced data with equal minority and majority class levels, others might receive minority class data alone and most others a part of majority and minority classes in several ratios. Hence this leads to a mixed training, combination of which can provide an enhanced training model. However, scalability of such a system is in question. Increase in data size leads to increased training data on the ensemble components. Since multiple such components are created, computational complexity increases exponentially, leading to scalability issues when used on huge datasets.

Real-time applications of bagging includes a general supervised learning enhancer [22], bank profitability analysis [23] and population dynamics estimator [24].

4.1.2 Boosting

Boosting is an ensemble learning technique primarily focused on reducing bias and variance in supervised learning techniques. It operates on the basis that several weak learners can be effectively combined to generate a strong learner. A weak learner has slight correlation with the true classification, better than random guessing, while a strong learner has high correlation with the true classifications [29]. Boosting operates by iteratively training weak classifiers on a single data distribution and hence building

the strong classifier based on the combination of rules generated by the weak classifier.

Boosting operates by initially fitting a model *f(x)* to the data. Being a supervised approach, the model is then reiterated and backtracked to identify the errors.

Fit the model to data *f(x)=y*

$$e(x) = y - f\ (x) \tag{2}$$

Where *e(x)* is the error.

A new model *f1* is created by incorporating errors into the model f(x)

$$f_1 = f(x) + e(x) \tag{3}$$

Unlike bagging, boosting reiterates through a single model, hence scalability issues are reduced extensively. Further, due to reiterated training and error handling mechanism, it is believed that boosting can handle very high imbalance levels, provided sufficient data is given for training.

Prediction based problems that include boosting are bankruptcy prediction on imbalanced data [7], drug combination prediction [25] and waste prediction [26]. ADA-BOOST [27] proposed by Freund et al. has attracted attention due to its generalized nature. However, it is specific problem based and does not have special concentrations on data imbalance.

4.1.3 Bucket of Models

Bucket of models is an ensemble modelling mechanism that operates on a variety of algorithms to provide the best algorithm based on the training data. Hence the bucket of models can produce results that is the best among available algorithms. When operated upon with a single algorithm, this technique provides the best among available results. However, while operated using several algorithms, due to the diverse nature, results obtained would be much better than using single techniques [30]. Creating an ensemble provides the flexibility to use any type of data on the model, rather than the training data that was used to create the trained model. The issue of imbalance and data hugeness will be handled by the best algorithm that can most effectively handle such issues. One major advantage of this technique is that it supports heterogeneity in selecting the individual classifiers, hence providing flexibility and performance.

4.1.4 Stacking

Stacking is an enhanced extension of bucket of models, in the sense that it supports heterogeneous models in the formation of ensemble [28]. However, unlike its counterpart, stacking requires a combiner algorithm that combines the results of individual models to provide a model that performs better than any of the individual models. The combiner algorithm is a heuristic that effectively operates on the results from individual models. A single layer logistic regression is usually used as a combiner; however, combiner is problem specific and can be effectively used to fine-tune the result sets to obtain results suiting to the problem domain [17]. The major advantage of this approach is that it utilizes several models, hence can provide the best component of all the available models. Although scalability might be an issue, the improved performance levels and heterogeneity incorporation would provide a huge tradeoff in terms of accuracy.

5 CONCLUSION

Real time data is usually prone to imbalance. Imbalance and hugeness of data complicates the prediction process. Predictions performed on such types of data are generally not effective when observed from the point of reliability. This paper analyzes ensembles and their effectiveness in operating on imbalanced data. An analysis of classifiers was performed, which indicates inefficiency in incorporating hugeness and imbalance levels. Hence ensembles are proposed as probable solutions to the issue. In-order to prove their validity towards handling imbalance and data hugeness, a theoretical analysis was performed based on their operational nature. An analysis of the ensembles indicate prominence of boosting and stacking techniques on huge and imbalanced data due to their scalability levels and effectiveness in handling imbalance. Future directions will be based on identifying the best ensemble and the best algorithms to be incorporated as ensemble components to create a generic ensemble model that can provide effective predictions irrespective of the imbalance levels in data.

REFERENCES

[1]. Mao, W., Wang, J., He, L. and Tian, Y., 2017. Online sequential prediction of imbalance data with two-stage hybrid strategy by extreme learning machine. Neurocomputing.

[2]. V., Fernández, A., García, S., Palade, V. and Herrera, F.,2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Information Sciences,250, 113-41.

[3]. Gong, J. and Kim, H., 2017. RHSBoost: Improving classification performance in imbalance data. Computational Statistics & Data Analysis, 111, pp.1-13.

[4]. Brodley, C.E. and Friedl, M.A., 1999. Identifying mislabeled training data. Journal of Artificial Intelligence Research, 11, pp.131-167.

[5]. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J. and Folleco, A., 2014. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. Information Sciences, 259, pp.571-595.

[6]. Napierała, K., Stefanowski, J. and Wilk, S., 2010. Learning from imbalanced data in presence of noisy and borderline examples. In International Conference on Rough Sets and Current Trends in Computing, Springer Berlin Heidelberg,(pp. 158-167).

[7]. Kim, M.J., Kang, D.K. and Kim, H.B., 2015. Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. Expert Systems with Applications, 42(3), pp.1074-1082.

[8]. Zong, W., Huang, G.B. and Chen, Y., 2013. Weighted extreme learning machine for imbalance learning. Neurocomputing, 101, pp.229-242.

[9]. Chaudhari, P., and Sane, S.S.,2016. Article: Multilevel Classification Exploiting Coupled Label Similarity with Feature Selection. IJCA Proceedings on Emerging Trends in Computing ETC 2016(4):1-4.

[10]. López, V., Fernández, A. and Herrera, F., 2014. On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. Information Sciences, 257, pp.1-13.

[11]. Maurya, C.K., Toshniwal, D. and Venkoparao, G.V., 2016. Online sparse class imbalance learning on big data. Neurocomputing, 216, pp.250-260.

[12]. Davis, J., and Mark, G., 2006. The relationship between Precision-Recall and ROC curves. Proceedings of the 23rd international conference on Machine learning. ACM, 2006.

[13]. Fawcett, T.,2006. An introduction to ROC analysis. Pattern recognition letters, 27.8, 861-874.

[14]. Hand, D.J. and Till, R.J., 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine learning, 45(2), pp.171-186.

[15]. Tomašev, N., and Dunja, M.,2013. "Class imbalance and the curse of minority hubs." Knowledge-Based Systems 53 (2013): 157-172.

[16]. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. and Herrera, F., 2012. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4), pp.463-484.

[17]. Ozay, M. and YarmanVural, F. T., 2013. A New Fuzzy Stacked Generalization Technique and Analysis of its Performance. arXiv:1204.0171

[18]. Polikar, R., 2006. Ensemble based systems in decision making. IEEE Circuits and systems magazine, 6(3), pp.21-45.

[19]. Rokach, L., 2010. Ensemble-based classifiers. Artificial Intelligence Review, 33(1-2), pp.1-39.

[20]. Breiman, L., 1996. Bagging predictors. Machine learning, 24(2), pp.123-140.

[21]. Ho, T.K., 1998. The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20(8), pp.832-844.

[22]. Aledo, J.A., Gámez, J.A. and Molina, D., 2017. Tackling the supervised label ranking problem by bagging weak learners. Information Fusion, 35, pp.38-50.

[23]. Erdal, H. and Karahanoğlu, İ., 2016. Bagging ensemble models for bank profitability: An emprical research on Turkish development and investment banks. Applied Soft Computing, 49, pp.861-867.

[24]. Simidjievski, N., Todorovski, L. and Džeroski, S., 2015. Predicting long-term population dynamics with bagging and boosting of process-based models. Expert Systems with Applications, 42(22), pp.8484-8496.

[25]. Xu, Q., Xiong, Y., Dai, H., Kumari, K.M., Xu, Q., Ou, H.Y. and Wei, D.Q., 2017. PDC-SGB: Prediction of effective drug combinations using a stochastic gradient boosting algorithm. Journal of theoretical biology, 417, pp.1-7.

[26]. Johnson, N.E., Ianiuk, O., Cazap, D., Liu, L., Starobin, D., Dobler, G. and Ghandehari, M., 2017. Patterns of waste generation: A gradient boosting model for short-term waste prediction in New York City. Waste Management.

[27]. Freund, Y. and Schapire, R.E., 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. In European conference on computational learning theory. Springer Berlin Heidelberg, pp. 23-37.

[28]. Wolpert, D.H., 1992. Stacked generalization. Neural networks, 5(2), pp.241-259.

[29]. Zhi-Hua, Z., 2012, Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC. p. 23. ISBN 978-1439830031.

[30]. Dzeroski, S.andZenko, B.,2004. Is Combining Classifiers Better than Selecting the Best One, Machine Learning, pp. 255—273.

[31]. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[32]. http://sci2s.ugr.es/keel/datasets.php

[33]. Halvaiee, Neda Soltani, and Mohammad Kazem Akbari. A novel model for credit card fraud detection using Artificial Immune Systems. Applied Soft Computing 24 (2014): 40-49.