Addressing the Big Data Multi-class Imbalance Problem with Oversampling and Deep Learning Neural Networks

V. M. González-Barcenas[1], E. Rendón[1], R. Alejo[1](✉),
E. E. Granda-Gutiérrez[2], and R. M. Valdovinos[3]

[1] Division of Postgraduate Studies and Research,
National Institute of Technology of Mexico (TecNM) Campus Toluca,
Av. Tecnológico s/n, Agrícola Bellavista, 52149 Metepec, México
ralejoe@toluca.tecnm.mx

[2] UAEM University Center at Atlacomulco, Universidad Autónoma del Estado de México, Carretera Toluca-Atlacomulco Km. 60, 50450 Atlacomulco, México

[3] Faculty of Engineering, Universidad Autónoma del Estado de México,
Cerro de Coatepec s/n, Ciudad Universitaria, 50100 Toluca, México

Abstract. The class imbalance problem is a challenging situation in machine learning but also it appears frequently in recent Big Data applications. The most studied techniques to deal with the class imbalance problem have been Random Over Sampling (ROS), Random Under Sampling (RUS) and Synthetic Minority Over-sampling Technique (SMOTE), especially in two-class scenarios. However, in the Big Data scale, multi-class imbalance scenarios have not extensively studied yet, and only a few investigations have been performed. In this work, the effectiveness of ROS and SMOTE techniques is analyzed in the Big data multi-class imbalance context. The KDD99 dataset, which is a popular multi-class imbalanced big data set, was used to probe these oversampling techniques, prior to the application of a Deep Learning Multi-Layer Perceptron. Results show that ROS and SMOTE are not always enough to improve the classifier performance in the minority classes. However, they slightly increase the overall performance of the classifier in comparison to the unsampled data.

Keywords: Multi-class imbalance · Deep learning neural networks · Big data

1 Introduction

Big Data applications have increased importantly in recent years [23]. The volume of information, the speed of data transference, and the variety of data are the main characteristics of Big Data paradigm [8,10]. Concerning to the volume

This work has been partially supported under grants of project 5106.19-P from TecNM.

A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11867, pp. 216–224, 2019.
https://doi.org/10.1007/978-3-030-31332-6_19

of information, a data set belongs to Big Data scale when it is difficult to process with traditional analytical systems [18].

In order to efficiently seize the large amount of information from Big Data applications, Deep Learning techniques have become an attractive alternative, because these algorithms generally allow to obtain better results than traditional machine learning methods [12,20]. Multi-Layer Perceptron (MLP), the most common neural network topology, has been also translated to the Deep Learning context [14].

Deep Learning MLP (DL-MLP) incorporates two or more hidden layers in its architecture [11], which increases the computational cost of processing large size and high dimension data sets. However, this disadvantage can be overtaken by using modern efficient frameworks, such as Apache-Spark [24] or Tensor-Flow [1]. Thus, the high performance, robustness to overfitting, and high processing capability of this deep neural networks can be exploited.

Nevertheless, deep learning algorithms are strongly affected by the class imbalance problem [6]. The class imbalance problem refers to situations where the number of samples in one or more classes of the data set is fewer than in another class (or classes), producing an important deterioration of the classifier performance [5]. In literature, many investigations dealing with this problem have been documented, being Random Over Sampling (ROS), Random Under Sampling (RUS) and Synthetic Minority Over-sampling Techniques (SMOTE) the most popular methods [15]. Although the results are not conclusive for the specific application in the Big Data scale, they have motivated the development of other over-sampling methods [13,17].

The KDD CUP 1999 intrusion detection data set (KDD99) was introduced at The Third International Knowledge Discovery and Data Mining Tools Competition [4]. It consists of more than 4 million instances (with 41 attributes); it is divided into twenty-three types of attacks clustered in four categories, therefore it is formally considered as Big Data [22]. Some attacks in KDD99 have less of ten instances; i.e., it is highly imbalanced and few represented, which implies a Big Data challenge with class imbalance problem [15,18].

Previous works have been focused in the study of the KDD99 dataset to probe different machine learning techniques. Nevertheless, most of them have used only a subset of it [22]. For example, in [23] KDD99 was divided into four two-class data sets and the class imbalance problem has been addressed with parallel models of evolutionary under-sampling methods based in the Map Reduce paradigm. Seo et al. [22] used a KDD99 subset of five classes: four of them were the attack categories and the fifth class was the normal connections; then, a wrapper method was proposed to find the best SMOTE ratio by identifying the best level of sampling for the minority classes.

In this paper, the whole KDD99 data set was analyzed, by using all the twenty three attacks as classes with the aim of study the performance of the classical oversampling approaches, like ROS and SMOTE, in the Big Data class imbalance context, while the Deep Learning MLP was used as base classifier.

2 Theoretical Framework

2.1 Deep Learning Multilayer Perceptron

MLP constitutes the most conventional neural network architecture. It is commonly based on three layers: input, output, and one hidden layer [14]. Thus, the MLP can be translated into a deep neural network by incorporating two or more hidden layers within its architecture, becoming a Deep Learning MLP. This allows to reduce the number of nodes per layer and uses fewer parameters, but it leads to a more complex optimization problem [11]. However, due to the availability of more efficient frameworks, such as Apache-Spark or Tensorflow, this disadvantage is less restrictive than before.

Traditionally, MLP has been trained with the back-propagation algorithm (which is based in the stochastic gradient descent) and its weights randomly initialized. However, in the late versions of DL-MLPs, the hidden layers are pre-trained by an unsupervised algorithm and the weights are optimized by the back-propagation algorithm [14].

MLP uses sigmoid activation functions, such as the hyperbolic tangent or logistic function. In contrast, DL-MLP includes (commonly) the Rectified Linear Unit (ReLU) $f(z) = \max(0, z)$ because typically learns much faster in networks with many layers, allowing training of a DL-MLP without unsupervised pre-training.

There are three variants of the descending gradient that differ in how many data are used to process the gradient of the objective function [21]: (a) Batch Gradient Descendent calculates the gradient of the cost function to the parameters for the entire training data set, (b) Stochastic Gradient Descendent performs an update of parameters for each training example, and (c) Mini-batch Gradient Descendent takes the best of the previous two and performs the update for each mini-batch of a given number of training examples.

The most common algorithms of descending gradient optimization are: (a) Adagrad, which adapts the learning reason of the parameters, making bigger updates for less frequent parameters and smaller for the most frequent ones, (b) Adadelta is an extension of Adagrad that seeks to reduce aggressiveness, monotonously decreasing the learning rate instead of accumulating all the previous descending gradients, restricting accumulation to a fixed size, and (c) Adam, that calculates adaptations of the learning rate for each parameter and stores an exponentially decreasing average of past gradients. Other important algorithms are AdaMax, Nadam and RMSprop [21].

2.2 Classical Sampling Methods Used to Deal with the Class Imbalance Problem

The class imbalance problem has been a hot topic in machine learning and data mining, and more recently in deep learning and Big Data [7,15]. Oversampling (mainly ROS and SMOTE) are the most common techniques used to face with the class imbalance problem, mainly due to their independence of the underlying classifier [17]. ROS replicates samples in the minority class biasing the discrimination process to compensate the class imbalance, while SMOTE generates artificial samples from the minority class by interpolating existing instances that lie close together [9].

Table 1. A brief summary of main characteristics of the KDD99 data set.

Attack name	Attacks in data set	Category	Class imbalance ratio
normal	972781	NORMAL	0.19859032
warezclient	1020	R2L	0.00020823
multihop	7	R2L	0.00000143
ftp_write	8	R2L	0.00000163
imap	12	R2L	0.00000245
guess_passwd	53	R2L	0.00001082
warezmaster	20	R2L	0.00000408
spy	2	R2L	0.00000041
phf	4	R2L	0.00000082
neptune	1072017	DOS	0.21884906
back	2203	DOS	0.00044974
teardrop	979	DOS	0.00019986
smurf	2807886	DOS	0.57322151
pod	264	DOS	0.00005389
land	21	DOS	0.00000429
buffer_overflow	30	U2R	0.00000612
loadmodule	9	U2R	0.00000184
rootkit	10	U2R	0.00000204
perl	3	U2R	0.00000061
portsweep	10413	PROBE	0.00212578
satan	15892	PROBE	0.00324430
ipsweep	12481	PROBE	0.00254796
nmap	2316	PROBE	0.00047280

The under-sampling methods also have shown effectiveness to deal with the class imbalance problem [13]: the RUS technique is one of the most successful under-sampling methods, which eliminates random samples from the original data set (usually from the majority class) to decrease the class imbalance. However, this method loses effectiveness when it removes significant samples [17]. To compensate this disadvantage, other important under-sampling methods include a heuristic mechanism [13].

Lately, Dynamic Sampling Methods have become an interesting alternative to sampling class imbalanced data sets because they automatically set the class imbalance sampling rate [2], and select the best samples to train the classifier [16]. The key of these methods is that they use the neural network output to identify those samples that are either close or in the decision regions of other classes; i.e., in the frontier decision or class overlap region.

3 Experimental Set-Up

KDD99 data set was used in the experimental stage, which is available from the University of California at Irvine (UCI) machine learning repository [4]. It contains about 4 million instances with 41 attributes each.

In order to deal with the Big Data multi-class imbalance problem, all the twenty-three attacks of KDD99 data set were defined as classes for this investigation. The hold–out method was used to randomly split the KDD99 data set in training (70%) and test (30%). Table 1 shows a brief summary of main characteristics of the KDD99 data set.

The main goal of this paper is to show the performance of classical over-sampling approaches (ROS and SMOTE) to deal with the Big Data class imbalance problem. SMOTE and ROS were selected because they have shown their success to deal with the multi-class imbalance problem and even SMOTE is considered the "de facto" standard in the framework of learning from imbalanced data [9]. Thus, the scikit-learn library was used to perform SMOTE and ROS algorithms. Scikit-learn is a free library software for machine learning for the Python programming language [19].

Two hidden layer were used in the DL-MLP with ReLU activation functions in its nodes, and softmax function on its output layer. The configuration of each hidden layer was 30 nodes. The number of hidden layers and nodes were obtained by a trial-error strategy. DL-MLP was performed in TensorFlow framework [1], and Adam algorithm [21] was used as the training method.

The most widely used metrics on investigations to face the multi-class imbalanced problems has been the Multi-class Area Under the receiver operating characteristic Curve (MAUC) [2] and the Geometric Mean of Sensitivity and Precision (g-mean) [25]. However, these are global metrics and the evidence of the individual performance of ROS and SMOTE over the minority classes is more interesting for this paper; thus, the accuracy by-class was used instead.

Finally, in order to compute the general classification performance, the Ranks method was used. This assigns the rank 1 to the best algorithm, 2 to the second best, 3 to the third best, and so on up to the umpteenth best rank; if ties exist, then the average rank is calculated. The lesser the rank number, the better the algorithm performance.

4 Results and Discussion

Table 2 shows the accuracies by-class obtained by SMOTE and ROS in each individual class. It is organized in three parts: the first column represents the evaluated class, the second column are the number of samples classified correctly

Table 2. Back-propagation classification performance. The results represent the averaged values between ten folds and the initialization of ten different weights of the neural network. The bold numbers represent the best average MAUC values.

Class	Standard		ROS		SMOTE	
	Correct/Total	Average(%)	Correct/Total	Average(%)	Correct/Total	Average(%)
normal	**291199/291835**	**99.7**	276466/291835	94.7	290712/291835	99.6
warezclient	244/306	79.7	237/306	77.4	**263/306**	**85.9**
multihop	0/3	0	0/3	0	**1/3**	**33.3**
ftp_write	0/3	0	**1/3**	**33.3**	**1/3**	**33.3**
imap	**3/4**	**75**	**3/4**	**75**	**3/4**	**75**
guess_passwd	0/16	0	**14/16**	**87.5**	11/16	68.7
warezmaster	0/6	0	2/6	33.3	**3/6**	**50**
spy	0/1	0	0/1	0	0/1	0
phf	0/1	0	**1/1**	**100**	**1/1**	**100**
neptune	**121542/321606**	**37.79**	121419/321606	37.75	114277/321606	35.5
back	642/661	97.1	657/661	99.3	**654/661**	**98.9**
teardrop	**294/294**	**100**	**294/294**	**100**	293/294	99.6
smurf	**842317/842366**	**99.9**	312365/842366	37	312365/842366	37
pod	77/80	96.2	**80/80**	**100**	79/80	98.7
land	**3/7**	**42.8**	**3/7**	**42.8**	**3/7**	**42.8**
buffer_overflow	0/9	0	**8/9**	**88.8**	7/9	77.7
loadmodule	0/3	0	0/3	0	0/3	0
rootkit	0/3	0	0/3	0	0/3	0
perl	0/1	0	**1/1**	**100**	**1/1**	**100**
portsweep	3009/3124	96.3	2974/3124	95.2	**3070/3124**	**98.2**
satan	4683/4768	98.2	46484683/4768	97.4	**47114683/4768**	**98.8**
ipsweep	**2960/3745**	**79**	2585/3745	69	2935/3745	78.3
nmap	590/695	84.8	**681/695**	**97.9**	562/695	80.8
Average Rank		2.18		1.95		1.87

and the total of samples belonging to these class, and the third column is the average accuracy by-class. This is repeated for each sampling method: Standard (unsampled), ROS and SMOTE.

It is noticeable in Table 2 that some minority classes like *back*, *teardrop* and *pod* seem unaffected by the class imbalance problem. Another example is class *imap*, which is very poorly represented but the DL-MLP classifies correctly three of four of its samples. This confirmed the findings of others works, which affirmed that the class imbalance problem only increases the major disadvantage of the algorithms based in the back-propagation; i.e., the slow rate of convergence of the neural network and often it is the cause of the poor classification performance of the classifier, but not always [3].

It is observed also that the classifier accuracy by-class, in a few minority classes is not improved by the application of ROS or SMOTE methods. For example, the accuracy of the class *multihop* is not increased using ROS. The accuracies of the classes *spy*, *loadmodule* and *rootkit* are neither improved by ROS and SMOTE. Moreover, the classifier performance on the minority class *ipsweep* was reduced when ROS or SMOTE were applied. This could be originated by the increase of the noise or overlap in these minority classes when they are sampled.

Within the machine learning community, it is known that the class imbalance problem is severely stressed by other factors, such as class overlapping, small disjuncts, the lack of density and information, noisy data, the significance of the borderline samples and its relationship with noisy samples, and the data set shift problem [17].

All of these classes have a common feature: they are severely imbalanced, and the origin of this imbalance comes from different sources. Thus, an important question is how to deal with this problem. Maybe, the solution to this problem is not only the over-sampling of the minority classes, but heuristically sub-sampling the majority classes close to severely imbalanced minority classes, in a similar way to [3]. Then, an effective over-sampling method should be applied. However, another problem appears in the scene: how to identify the decision frontier of those minority classes. The use of the neuronal network output could be an interesting alternative [2].

Table 2 also exhibits that, in overall, the sampling methods improve the classifier performance in comparison to the unsampled data set. The average rank for both, SMOTE (1.87) and ROS (1.95), represent better results than standard rank (2.18).

In Big Data context, results from Table 2 confirm the conclusions of other investigations, which affirm that the class imbalance problem adversely affect the classifier performance, but in other situations it is not the main cause of effectiveness loss of classifier. In other words, the class imbalance problem in Big Data follows a similar behavior that the studied so far in machine learning community.

5 Conclusion

In this paper, the performance of two successful methods to deal with the multi-class imbalance problem, ROS and SMOTE, was analyzed. Results show that ROS and SMOTE are not always enough to improve the classifier performance in the minority classes, in the Big Data multi-class imbalance context. However, these oversampling methods increase the DL-MLP accuracy on most of the cases. It is considered necessary a cleaning stage before applying either SMOTE or ROS, and the neural network output could be a good alternative for this stage. Thus, further research is required to investigate the potential of recent dynamic sampling methods [2,16], which use the neural network output to identify and delete samples from majority classes that are close or in the minority classes decision regions. Subsequently, the use of SMOTE or ROS would improve the classification performance on these minority classes.

References

1. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. OSDI 2016, pp. 265–283, USENIX Association, Berkeley (2016). http://download.tensorflow.org/paper/whitepaper2015.pdf
2. Alejo, R., Monroy-de Jesús, J., Ambriz-Polo, J.C., Pacheco-Sánchez, J.H.: An improved dynamic sampling back-propagation algorithm based on mean square error to face the multi-class imbalance problem. Neural Comput. Appl. **28**(10), 2843–2857 (2017). https://doi.org/10.1007/s00521-017-2938-3
3. Alejo, R., Valdovinos, R., García, V., Pacheco-Sanchez, J.: A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. Pattern Recogn. Lett. **34**(4), 380–388 (2013)
4. Asuncion, A., Newman, D.: UCI machine learning repository (2007). www.ics.uci.edu/~mlearn/
5. Błaszczyński, J., Stefanowski, J.: Local data characteristics in learning classifiers from imbalanced data. In: Gawęda, A.E., Kacprzyk, J., Rutkowski, L., Yen, G.G. (eds.) Advances in Data Analysis with Computational Intelligence Methods. SCI, vol. 738, pp. 51–85. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67946-4_2
6. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw. **106**, 249–259 (2018). https://doi.org/10.1016/j.neunet.2018.07.011
7. Dong, Q., Gong, S., Zhu, X.: Imbalanced deep learning by minority class incremental rectification. CoRR abs/1804.10851 (2018)
8. Elshawi, R., Sakr, S., Talia, D., Trunfio, P.: Big data systems meet machine learning challenges: towards big data science as a service. Big Data Res. **14**, 1–11 (2018). https://doi.org/10.1016/j.bdr.2018.04.004
9. Fernandez, A., Garcia, S., Herrera, F., Chawla, N.V.: SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J. Artif. Intell. Res. **61**, 863–905 (2018)
10. Fernández, A., del Río, S., Chawla, N.V., Herrera, F.: An insight into imbalanced big data classification: outcomes and challenges. Complex Intell. Syst. **3**(2), 105–120 (2017). https://doi.org/10.1007/s40747-017-0037-9
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
12. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: a review. Neurocomputing **187**, 27–48 (2016)
13. He, H., Garcia, E.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009). https://doi.org/10.1109/TKDE.2008.239
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
15. Leevy, J.L., Khoshgoftaar, T.M., Bauder, R.A., Seliya, N.: A survey on addressing high-class imbalance in big data. J. Big Data **5**(1), 42 (2018). https://doi.org/10.1186/s40537-018-0151-6
16. Lin, M., Tang, k., Yao, X.: Dynamic sampling approach to training neural networks for multiclass imbalance classification. IEEE Trans. Neural Netw. Learn. Syst. **24**(4), 647–660 (2013). https://doi.org/10.1109/TNNLS.2012.2228231
17. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf. Sci. **250**, 113–141 (2013). https://doi.org/10.1016/j.ins.2013.07.007

18. Oussous, A., Benjelloun, F.Z., Lahcen, A.A., Belfkih, S.: Big data technologies: a survey. J. King Saud Univ. - Comput. Inf. Sci. **30**(4), 431–448 (2018). https://doi.org/10.1016/j.jksuci.2017.06.001
19. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
20. Reyes-Nava, A., Sánchez, J.S., Alejo, R., Flores-Fuentes, A.A., Rendón-Lara, E.: Performance analysis of deep neural networks for classification of gene-expression microarrays. In: Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Olvera-López, J.A., Sarkar, S. (eds.) MCPR 2018. LNCS, vol. 10880, pp. 105–115. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92198-3_11
21. Ruder, S.: An overview of gradient descent optimization algorithms. CoRR abs/1609.04747 (2016)
22. Seo, J.H., Kim, Y.H.: Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection. Comput. Intell. Neurosci. **2018**, 1–11 (2018). https://doi.org/10.1155/2018/9704672
23. Triguero, I., et al.: Evolutionary undersampling for imbalanced big data classification. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 715–722, May 2015. https://doi.org/10.1109/CEC.2015.7256961
24. Zaharia, M., et al.: Apache Spark: a unified engine for big data processing. Commun. ACM **59**(11), 56–65 (2016). https://doi.org/10.1145/2934664
25. Zarinabad, N., Wilson, M., Gill, S., Manias, K., Davies, N., Peet, A.: Multiclass imbalance learning: improving classification of pediatric brain tumors from magnetic resonance spectroscopy. Magn. Reson. Med. **77**(6), 2114–2124 (2017). https://doi.org/10.1002/mrm.26318