The Effect of Sampling Methods on the CICIDS2017 Network Intrusion Data Set

Yan-Bing Ho, Wun-She Yap, and Kok-Chin Khor

Abstract Handling unbalanced intrusion detection data sets are difficult as minority intrusion classes may not be easy to detect. One of the possible causes of the problem is the characteristic of learning algorithms that usually favour majority classes in data sets. The contribution of this study is to improve the detection rate for intrusions in the unbalanced CICIDS2017 data set by using sampling techniques. We evaluated Random Under-Sampling (RUS), Synthetic Minority Over-sampling Technique (SMOTE) and the combination of RUS and SMOTE. After applying the sampling techniques, we performed intrusion detection and used the accuracy plus True Positive Rate (TPR) as the evaluation metrics for the detection results. The results showed that RUS gave the best detection performance overall. Besides, 12 out of the 15 classes, including some hard-to-detect minority classes, were detected with result improvement.

Keywords Samplings · Intrusion detection · CICIDS2017

1 Introduction

Intrusion Detection Systems (IDS) are used to detect malicious activities in the networks and information systems. Due to the increasing network scale and traffic, large network data are generated almost every seconds. However, intrusion activities are relatively rare compared to the overall traffic amount causing the network data to be unbalanced.

Y.-B. Ho (✉) · W.-S. Yap · K.-C. Khor
Lee Kong Chian Faculty of Engineering Science, Universiti Tunku Abdul Rahman, Kampar, Malaysia
e-mail: yanbing114@1utar.my

W.-S. Yap
e-mail: yapws@utar.edu.my

K.-C. Khor
e-mail: kckhor@utar.edu.my

H. Kim and K. J. Kim (eds.), *IT Convergence and Security*, Lecture Notes in Electrical Engineering 782, https://doi.org/10.1007/978-981-16-4118-3_4

Using such data to train detection models for IDS is difficult because the learning algorithms usually favour large classes for maximising accuracy and may have difficulties detecting the minority intrusions. Further, minority intrusions may not be able to form actual decision boundaries for the learning algorithms. Decision boundaries are important as they are the regions in a feature space that separates classes of a data set so that the learning algorithms can learn the classes effectively.

In this study, we attempted to improve the detection rates for minority intrusions by balancing the data set involved. The data set in this study used was CICIDS 2017 [1]. We firstly attempted under-sampling for the large class. Secondly, we attempted over-sampling, Synthetic Minority Over-sampling Technique (SMOTE) [2] for the seven weak intrusion classes that usually give weak detection rates. Finally, we combined both sampling techniques to seek better improvement in intrusion detection.

2 Literature Review

2.1 CICIDS2017 Data Set Overview

The CICIDS2017 data set [1] contains eight different files, and each of them contains network activities collected over five days. Table 1 shows the class distribution of the CICIDS2017 data set after combining the eight files. It comprises 2,830,743 instances, 78 features and 15 classes with no duplicated data. The data set is highly unbalanced as the BENIGN class takes 80.3% of the data set.

2.2 Sampling Techniques

Unbalanced class distribution is a common problem for real-world data sets such as network intrusions detection [1] and credit card fraud detection [3]. The rare classes are often the primary interests of classification [4]. Researchers have proposed several sampling techniques to tackle the unbalanced class distribution and improve classification performance, i.e., over-sampling, under-sampling, and combining sampling [5].

2.2.1 Over-Sampling

Over-sampling duplicates instances of minority classes or generates the duplicates based on the characteristic of the minority classes. This shall decrease the rareness of minority classes, thereby decreasing the overall level of class imbalance [4]. A basic over-sampling method is Random Over-Sampling (ROS) that duplicates the minority instances randomly [6]. Increasing the size of a minority class using ROS

Table 1 The class distribution of the CICIDS2017 data set

No	Normal/attack label	Number of instances	% of the total instances
1	BENIGN	2,273,097	80.3004
2	DoS hulk	231,073	8.1630
3	PortScan	158,930	5.6144
4	DDoS	128,027	4.5227
5	DoS goldeneye	10,293	0.3636
6	FTP-patator	7938	0.2804
7	SSH-patator	5897	0.2083
8	DoS slowloris	5796	0.2048
9	DoS slowhttptest	5499	0.1943
10	Bot	1966	0.0695
11	Web attack–Brute force	1507	0.0532
12	Web attack–XSS	652	0.0230
13	Infiltration	36	0.0013
14	Web attack–Sql injection	21	0.0007
15	Heartbleed	11	0.0004
	Total	2,830,743	100.0000

can increase the time taken to build a model and may lead to an overfitting problem [7]. Further, the lack of minority information may persist even after duplicating existing instances using ROS. Studies [7] show that ROS is less effective at improving the detection of minority classes. Therefore, Chawla et al. [2] proposed an advanced over-sampling method, Synthetic Minority Over-sampling Technique (SMOTE), to create new minority instances rather than duplicating existing instances. This technique creates synthetic instances using the nearest neighbour rule in the feature space. However, SMOTE considers only minority classes without taking care of majority classes. Therefore, increasing the size of minority classes may increase the chances of overlapping among classes [8].

2.2.2 Under-Sampling

Under-sampling removes the existing majority instances to balance a data set. A basic under-sampling technique is Random Under-Sampling (RUS) that removes the majority instances randomly. However, this may cause the removal of potentially useful information from a data set and the performance degradation in classification [4, 9].

2.2.3 Combining Sampling

Combining sampling is to apply a combination of sampling techniques on an unbalanced data set to improve the classification performance [10]. One example of combining sampling is to combine under-sampling and over-sampling. Das et al. [6] stated that under-sampling should be applied before over-sampling as a data cleaning method because it helps reduce the overlapping classes' effect.

3 Methodology

We transformed the CICIDS2017 data set into a format understandable by data mining algorithms used with data pre-processing. We replaced the missing values in the CICIDS2017 data set with the mean values of the features. Infinity values were then replaced by values that were ten times the maximum feature value. We also used Z-score normalisation to standardise all the features because the original range of their values is varied widely.

The unbalanced class distribution has caused the learning algorithms to bias majority classes and may produce low detection rates for minority classes. We used three sampling methods to address the problem, namely, over-sampling, under-sampling and hybrid sampling.

Four learning algorithms were used for intrusion detection, i.e., Gaussian Naïve Bayes (GNB) [11], C4.5 [11], Neural Network (NN-MLP) [12], K-Nearest Neighbour (KNN) [13], and Logistic Regression (LR) [14]. We used tenfold cross-validation to evaluate the performance of the learning algorithms. The data set was split into ten groups for both training and testing purposes.

The CICIDS2017 data set is unbalanced. Therefore, accuracy is a less suitable metric to evaluate learning algorithms. If the majority class is correctly classified, then the accuracy shall be high even though the rare classes are wrongly classified. Complementing accuracy with True Positive Rate (TPR) to examine learning algorithms' performance is a better option. This is because TPR can examine the detection performance for each of the classes in the data set.

4 Results and Discussion

Table 2 shows the detection result using the learning algorithms, i.e., Gaussian Naïve Bayes (GNB), C4.5, Neural Network (NN-MLP), K-Nearest Neighbour (KNN), and Logistics Regression (LR). By comparing the average TPR, C4.5 was the best performer among the single classifiers, with an accuracy (average TPR) of 0.9927. We noticed seven weak intrusion classes (bold classes in Table 2) that were hard to detect; below-average TPRs (less than 0.8.) were obtained using some of the learning algorithms. They were Bot, DoS Slowloris, Heartbleed, Infiltration, and

Table 2 The intrusion detection result for the full CICIDS 2017 data set using the learning algorithms. The classes in bold are the weak intrusion classes that give below average TPR

	GNB	C4.5	NN-MLP	KNN	LR
BENIGN	0.6500	0.9939	0.9955	0.9930	0.9709
Bot	0.9980	0.7872	0.3481	0.5607	0.0092
DDoS	0.9573	0.9996	0.9984	0.9977	0.9648
DoS goldeneye	0.9320	0.9173	0.9484	0.9610	0.8060
DoS hulk	0.7123	0.9898	0.9874	0.9874	0.9210
DoS slowhttptest	0.6767	0.9073	0.8440	0.8658	0.8056
DoS slowloris	0.6290	0.9154	0.8848	0.8681	0.4756
FTP-patator	0.9956	0.9961	0.9880	0.9948	0.5491
Heartbleed	0.8000	0.9000	0.0000	1.0000	0.0000
Infiltration	0.8417	0.7583	0.0000	0.2000	0.0750
PortScan	0.9885	0.9913	0.9809	0.9845	0.9939
SSH-patator	0.9944	0.9969	0.9646	0.9866	0.0270
WA–Brute force	0.0916	0.7306	0.1128	0.7658	0.0000
WA–Sql injection	1.0000	0.5667	0.0000	0.1500	0.0000
WA–XSS	0.9232	0.4125	0.0169	0.2990	0.0000
Average TPR	0.6907	**0.9927**	0.9922	0.9911	0.9614

three Web Attacks (WAs)—Brute Force, Sql Injection and XSS. Such performance could be caused by the unbalanced class distribution of the data set as the BENIGN is the immense majority in the data set. The learning algorithms' characteristic that favours the majority class (BENIGN) also contribute to such performance.

To improve the detection rate overall and for these weak intrusion classes, we attempted under-sampling, over-sampling and a combination of them to balance the data set.

Firstly, we attempted random under-sampling (RUS) on the majority class, BENIGN to balance the data set and reduce the effect of the majority BENIGN. Table 3 shows the RUS results using C4.5. C4.5 was used since it gave the best performance among the single classifiers, as shown in Table 1. The best overall accuracy (average TPR of 0.9985) was achieved by reducing BENIGN between 30 and 90% of its original size. The result was not much different from the full data set. However, the TPR for 12 of the classes were improved, including four of the weak intrusion classes.

We then attempted the over-sampling technique, Synthetic Minority Over-sampling (SMOTE), to increase the size of the seven weak intrusion classes. Table 4 shows the results of the over-sampling. The best average TPR (0.9900) was achieved by increasing the size of these minority classes to 250% of the full data set, and the result was slightly weak compared with the full data set. Improvements were noticed for some of the classes, including only three of the weak intrusion classes. Overall, the detection performance was slightly weak as compared to RUS.

Table 3 The intrusion detection result for the CICIDS 2017 data set resampled using RUS on BENIGN. The numbers in bold shows better TPRs than the results obtained using the full data set

Label	30%	40%	50%	60%	70%	80%	90%	Full data set
BENIGN	**0.9989**	**0.9990**	**0.9991**	**0.9991**	**0.9992**	**0.9992**	**0.9992**	0.9939
Bot	**0.8861**	**0.8861**	**0.8698**	**0.8596**	**0.8474**	**0.8210**	**0.8128**	0.7872
DDoS	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**	0.9996
DoS goldeneye	**0.9971**	**0.9965**	**0.9969**	**0.9963**	**0.9965**	**0.9963**	**0.9959**	0.9173
DoS hulk	**0.9995**	**0.9994**	**0.9992**	**0.9993**	**0.9992**	**0.9992**	**0.9992**	0.9898
DoS slowhttptest	**0.9891**	**0.9891**	**0.9862**	**0.9869**	**0.9898**	**0.9840**	**0.9876**	0.9073
DoS slowloris	**0.9959**	**0.9959**	**0.9962**	**0.9959**	**0.9959**	**0.9955**	**0.9962**	0.9154
FTP-Patator	**0.9992**	**0.9995**	**0.9992**	**0.9992**	**0.9992**	**0.9992**	**0.9992**	0.9961
Heartbleed	0.8000	0.8000	0.6000	0.6000	0.8000	0.8000	0.8000	0.9000
infiltration	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.7583
PortScan	**0.9978**	**0.9968**	**0.9961**	**0.9954**	**0.9948**	**0.9942**	**0.9931**	0.9913
SSH-Patator	**0.9983**	**0.9993**	**0.9986**	**0.9983**	**0.9980**	**0.9980**	**0.9980**	0.9969
WA–Brute force	0.7145	0.7092	0.7118	0.7145	0.7131	0.7118	0.7158	0.7306
WA–Sql Injection	**0.6000**	**0.6000**	0.1000	0.3000	0.2000	0.2000	0.5000	0.5667
WA–XSS	**0.4141**	0.4049	0.3988	0.4049	0.4049	**0.4141**	**0.4202**	0.4125
average TPR	**0.9985**	**0.9985**	**0.9985**	**0.9985**	**0.9985**	**0.9985**	**0.9984**	0.9926

Finally, we combined RUS and SMOTE to seek improvement in detection. Figure 1 shows the TPRs achieved using the combination of these two sampling techniques. The x-axis represents the percentage of the remaining majority class samples, BENIGN, after under-sampling. On the other hand, the y-axis represents the TPRs. There are four line-plots that represent the percentage of over-sampling on the seven minority classes. The best result achieved was 30% under-sampling on BENIGN and 300% over-sampling on the seven weak intrusion classes. The average TPR obtained was 0.9934.

Table 5 shows the result comparison of the full data set and the resampled data sets. The RUS (30%) achieved the best average TPR (0.9985) among the sampling techniques. Using RUS, we achieved the best TP rates for 11 out of 15 classes as compared with the full data set, and the data sets resulted using SMOTE and RUS (30%) + SMOTE (300%). To conclude, the sampling technique RUS gave a slight improvement in detection overall and most of the CICIDS 2017 data set classes.

Table 4 The results for the data set yielded using SMOTE on the five minority classes (bolded font). The numbers in bold shows better TPRs than the results obtained using the full data set

Label	100% (Full data set)	150%	200%	250%	300%
BENIGN	0.9939	0.9929	0.9929	0.9929	0.9926
Bot	0.7872	**0.7968**	**0.7968**	**0.8075**	**0.8121**
DDoS	0.9996	0.9994	0.9996	0.9994	0.9995
DoS goldeneye	0.9173	0.9110	0.9109	**0.9178**	0.9177
DoS hulk	0.9898	**0.9899**	**0.9901**	**0.9907**	**0.9907**
DoS slohttptest	0.9073	0.8564	0.8762	**0.9269**	**0.9254**
DoS slowloris	0.9154	0.8826	0.8824	0.8817	0.8948
FTP-patator	0.9961	0.9958	**0.9966**	**0.9972**	**0.9962**
Heartbleed	0.9000	0.8000	0.9000	0.8000	0.9000
Infiltration	0.7583	**0.8417**	**0.7833**	**0.7917**	**0.8500**
PortScan	0.9913	0.9912	0.9912	0.9912	0.9911
SSH-patator	0.9969	0.9969	0.9969	**0.9973**	**0.9971**
WA–Brute Force	0.7306	**0.7399**	**0.7472**	**0.7446**	**0.7339**
WA–Sql Injection	0.5667	0.5667	0.4167	**0.6333**	0.4167
WA–XSS	0.4125	0.3678	0.3649	0.3894	0.3990
Average TPR	0.9927	0.9899	0.9899	0.9900	0.9898

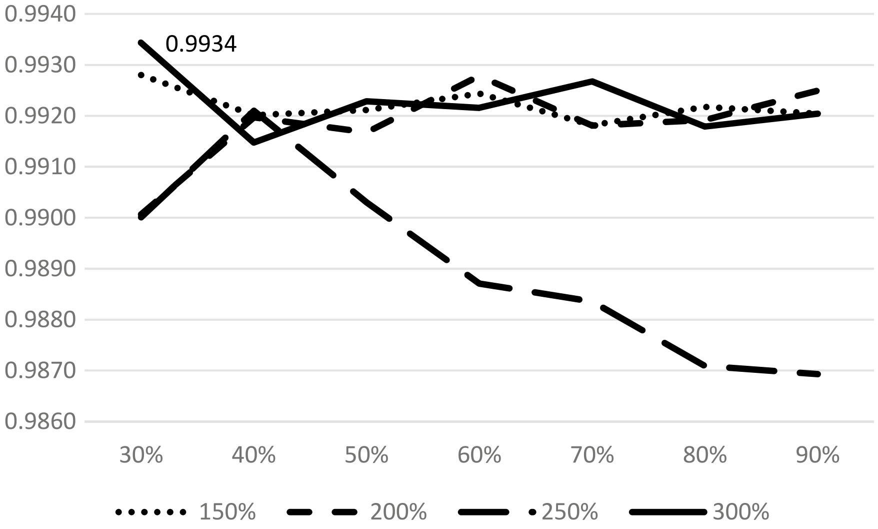

Fig. 1 The detection results achieved using the combination of under-sampling and oversampling. RUS (30%) + SMOTE (300%) gives the best result—a TPR of 0.9934

Table 5 The result comparison using the full and resampled data sets. The numbers in bold shows the best TPRs obtained by using RUS

	Full dataset	SMOTE	RUS (30%)	RUS (30%) + SMOTE (300%)
BENIGN	0.9939	0.9929	**0.9989**	0.9945
Bot	0.7872	0.7842	0.8861	0.8918
DDoS	0.9996	0.9995	**0.9998**	0.9994
DoS goldeneye	0.9173	0.9172	**0.9971**	0.9743
DoS hulk	0.9898	0.9896	**0.9995**	0.9880
DoS slowhttptest	0.9073	0.8520	**0.9891**	0.8735
DoS slowloris	0.9154	0.9173	**0.9959**	0.8577
FTP-patator	0.9961	0.9961	**0.9992**	0.9976
Heartbleed	0.9000	0.9000	0.8000	1.0000
Infiltration	0.7583	0.8667	0.6667	0.7500
PortScan	0.9913	0.9912	**0.9978**	0.9971
SSH-patator	0.9969	0.9969	**0.9983**	0.9968
WA–Brute Force	0.7306	0.7438	0.7145	0.7439
WA–Sql Injection	0.5667	0.5833	**0.6000**	0.4500
WA–XSS	0.4125	0.4141	**0.4141**	0.3834
Average TPR	0.9927	0.9918	**0.9985**	0.9934

5 Conclusion and Future Work

This paper aims to use sampling techniques to improve the intrusion detection rate using the CICIDS 2017 data set. We attempted under-sampling, over-sampling, and a hybrid of them to balance the data set, as the learning algorithms work by assuming data sets involved are balanced in class distribution. The RUS gave the best overall accuracy, measured using average TPR. The average TPR obtained was 0.9985, a slight improvement compared to the full and resampled data sets using SMOTE and RUS + SMOTE. We also noticed an improvement in detecting most of the classes, including some weak intrusion classes.

Acknowledgements This work was supported by the UTAR Research Grant (IPSR/RMC/UTARRF/2019-C2/K02) provided by the Universiti Tunku Abdul Rahman, Malaysia.

References

1. Sharafaldin I, Lashkari AH, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterisation. In: ICISSP 2018–Proc. 4th Int. Conf. Inf. Syst. Secur. Priv. 2018-Janua. pp 108–116

2. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority oversampling technique. J Artif Intell Res 16:321–357
3. Kalid SN, Ng K, Tong G, Khor K (2020) A multiple classifiers system for anomaly detection in credit card data with unbalanced and overlapped classes. IEEE Access 8:28210–28221
4. Weiss G (2004) Mining with rarity: a unifying framework. SIGKDD Explor 6:7–19
5. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. Prog Artif Intell 5:221–232
6. Das B, Krishnan NC, Cook DJ (2014) Handling imbalanced and overlapping classes in smart environments prompting dataset
7. Drummond C, Holte RC (2003) C4.5, class imbalance, and cost sensitivity : why under-sampling beats over-sampling
8. Sáez JA, Luengo J, Stefanowski J, Herrera F (2014) Managing borderline and noisy examples in imbalanced classification by combining smote with ensemble filtering. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 8669 LNCS. pp 61–68
9. Dal Pozzolo A, Caelen O, Bontempi G (2010) Comparison of balancing techniques for unbalanced datasets. Mach Learn Gr Univ Libr Bruxelles Belgium 16:732–735
10. Seiffert C, Khoshgoftaar TM, Van Hulse J (2009) Hybrid sampling for imbalanced data. Integr Comput Aided Eng 16:193–210
11. Abdulrahman AA, Ibrahem MK (2019) Evaluation of DDoS attacks detection in a new intrusion dataset based on classification algorithms. Iraqi J Inform Commun Technol 1:49–55
12. Toupas P, Chamou D, Giannoutakis KM, Drosou A, Tzovaras D (2019) An intrusion detection system for multi-class classification based on deep neural networks. In: Proc.–18th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2019. pp 1253–1258
13. Yong Y (2012) The research of imbalanced data set of sample sampling method based on K-Means cluster and genetic algorithm. Energy Procedia 17:164–170
14. Zhang Y, Chen XU, Jin LEI, Wang X, Guo DA (2019) Network intrusion detection: based on deep hierarchical network and original flow data. IEEE Access 7:37004–37016