Partial Undersampling of Imbalanced Data for Cyber Threats Detection

Md Moniruzzaman
Federation University Australia
Ballarat, VIC
m.moniruzzaman@federation.edu.au

A.M. Bagirov
Federation University Australia
Ballarat, VIC
a.bagirov@federation.edu.au

Iqbal Gondal
Internet Commerce Security Laboratory (ICSL)
Ballarat, VIC
iqbal.gondal@federation.edu.au

ABSTRACT

Real-time detection of cyber threats is a challenging task in cyber security. With the advancement of technology and ease of access to the internet, more and more individuals and organizations are becoming the target for various cyber attacks such as malware, ransomware, spyware. The target of these attacks is to steal money or valuable information from the victims. Signature-based detection methods fail to keep up with the constantly evolving new threats. Machine learning based detection has drawn more attention of researchers due to its capability of detecting new and modified attacks based on previous attack's behaviour. The number of malicious activities in a certain domain is significantly low compared to the number of normal activities. Therefore, cyber threats detection data sets are imbalanced. In this paper, we proposed a partial undersampling method to deal with imbalanced data for detecting cyber threats.

CCS CONCEPTS

• **General and reference**; • **Security and privacy**;

KEYWORDS

Cyber threats, Supervised learning, Clustering, Imbalanced data.

ACM Reference Format:
Md Moniruzzaman, A.M. Bagirov, and Iqbal Gondal. 2019. Partial Undersampling of Imbalanced Data for Cyber Threats Detection. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Machine learning based threat detection methods have significant benefits over rule-based or signature based detection. Signature based detection methods needs frequent update of their database to deal with new threats and they are unable to detect unknown attacks. Machine learning based method can detect those new and unknown attacks based on previously trained information. The success of machine learning based methods heavily depends on good trained model. In security domain most of the training dataset is imbalanced as the number of infected samples is significantly lower compared to benign samples.

A dataset is called imbalanced when the number of samples from one or more class is significantly more than the number of samples from other classes. The majority class(es) dominates the training model which leads to a poor classification outcome for the minority class(es). This makes the detection of cyber threats using machine learning approaches a challenge. In cyber security area, detecting those minority samples have higher importance than the majority samples. But it is also important not to misclassify many samples of the majority class either. Finding an optimal balance between cost and performance is an important problem to consider. In this paper, we propose a partial undersampling method to deal with imbalanced data and compare the performance with four mainstream classifiers along with other undersampling and oversampling methods.

The rest of the paper is organized as follows. Section 2 presents an overview of related work. Section 3 discusses our proposed approach. Our experimental setup is discussed in Section 4. Computational results are reported in Section 5. Finally, Section 6 provides concluding remarks and possible direction for future research.

2 LITERATURE REVIEW

Various strategies have been developed to deal with imbalanced datasets. In the broad sense it can be categorized as data level technique and algorithmic level technique. Each of them can be divided in sub-categories. Fig. 1 summarizes the categories of classes imbalanced learning proposed in [14]. Some researchers considered cost-sensitive algorithms as a combination of both data level and algorithmic level technique [7, 16].

The oversampling method duplicates the samples in the minority classes in order to enhance their cardinality [8]. Several techniques are proposed for oversampling. The simplest oversampling method is random oversampling (ROS), which duplicates randomly selected minority objects. The drawback of this approach is that minority objects are grouped together in small areas from where the seed for oversampling is selected. This will cause a problem for the classifiers with the over-fitting problem [11]. The informed oversampling approach like synthetic minority over-sampling techniques (SMOTE) generates synthetic minority class samples to balance the class distribution [3] which eliminates the problem of ROS. It has received a lot of admiration and has an extensive range of practical applications. Many variants of SMOTE have been proposed like adaptive synthetic sampling approach (AdaSyn) [10], Borderline-SMOTE [9], Majority weighted minority oversampling technique and weighted kernel based SMOTE. The drawbacks of oversampling method is that it adds time and memory overhead [12], can cause over-fitting

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

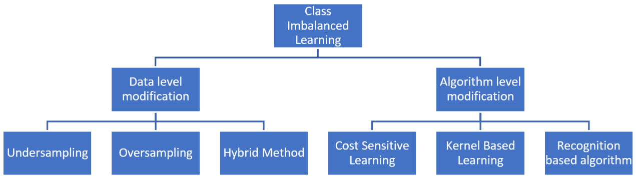

Figure 1: Categories of class imbalanced learning

and some features can not be synthetically generated or lose its property in synthetic data.

Undersampling method reduces the number of points from majority classes to make a balanced training set. But removing points may remove significant information from dataset which will lead to a poor classification. The simplest undersampling method is random undersampling (RUS), which randomly selects and removes majority points until the dataset becomes balanced. If a dataset have very high imbalanced ratio, then undersampling method will remove too many points from the dataset which results in significant loss of information for training the model and will affect the performance.

3 PROPOSED APPROACH

Undersampling method removes a large number of majority points from the dataset to make it balanced. Which results in loss of information and as a result, the accuracy of majority class drops. We propose a partial undersampling on the dataset. Our method works in two phases:

Phase 1. In this phase we apply the incremental clustering algorithm to calculate clusters in the data set.

Phase 2. Using outcomes of Phase 1, we apply undersampling inside some clusters and a supervised training model is created for each cluster.

Phase 1. In this phase we apply the incremental clustering algorithm to find clusters in the dataset. The most popular algorithm for solving the clustering problem is the k-means algorithm. However, this algorithm is very sensitive to the choice of starting cluster centers. Therefore, we apply the incremental clustering algorithm - the modified global k-means algorithm (MGKM) introduced in [2]. This algorithm computes clusters gradually starting from one cluster, which is the whole dataset, and adds one cluster center at each iteration.

The most important component of the MGKM algorithm is the procedure for finding starting cluster centers. These points are found by minimizing the so-called auxiliary clustering function. In its first step, the MGKM algorithm calculates the centre of the whole dataset. Assume that we have already calculated k cluster centres, that is solved the k-clustering problem. In order to solve the $(k+1)$-clustering problem we first formulate the k-th auxiliary clustering problem and solve it to find the set of starting cluster centres for the $(k+1)$-th cluster centre. Then each point from this set is used as a starting cluster centre together with other k cluster centres to solve the clustering problem itself. Both the clustering and the auxiliary clustering problems are solved by applying the k-means algorithm.

We defined majority and minority classes as follows: Let the dataset A containing p classes: $A_1, \dots, A_p$ and n_i be a number of points in the class $A_i,\ i = 1, \dots, p$. Then the total number N of points in the data set A is:

$$N = \sum_{i=1}^{p} n_i.$$

Then the average number $\bar{N}$ of points per class is:

$$\bar{N} = N/p.$$

Define the following threshold:

$$N_T = \alpha \bar{N}$$

where we set $\alpha = 0.5$.

A class $A_j,\ j \in \{1, \dots, p\}$ is called a *majority class* if $n_j \geq N_T$, otherwise it is called a minority class.

Our incremental clustering algorithm continues if none of the clusters is imbalanced or it reaches the maximum number of clusters. After this phase we get the centres and radii of k clusters, a list of points belong to each cluster and the class distribution for each cluster. All of this information are passed to the next phase.

Phase 2: Supervised classification. Based on the class distribution for each cluster, we may have the following cluster types:

Type 1: Cluster containing points only from minority classes.
Type 2: Cluster containing points only from majority classes.
Type 3: Cluster containing points from more than one classes.

For type 1 cluster, no undersampling is done and no classification model is trained, instead the whole cluster region is allocated to the minority class. All the points from the cluster type 2 are accumulated and a combined model is trained if the combined points belong to more than one classes, otherwise the whole region is allocated for the representative majority class. For type 3 clusters, undersampling is applied if minority class has at least 5 points. A classification model is trained for each cluster of type 3.

Once the models are trained they will be saved and be used to predict a new observation based on collected features from them. Figure 2 shows the working principle of our proposed model.

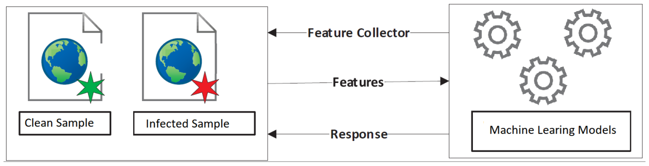

Figure 2: Working principle of proposed model

Classification rules. To classify a new observation the following classification rules are followed:

1. Check whether this observation belongs to the neighbourhood of any cluster of type 1. If yes, the observation is assigned to corresponding class.
2. If an observation belongs to the neighbourhood of any cluster of type 2 and the combined model is not trained, then assign the observation to the representative class of the combined majority class.

3. If an observation belongs to the neighbourhood of any cluster of type 2 and a combined model is trained, then use that model to predict it's class.
4. Check whether this observation belongs to neighbourhood of any cluster of type 3. If yes, then use the trained model for that particular cluster to predict it's class.

4 EXPERIMENTAL SETUP

In our experiment, we used four mainstream classifiers (KNN, Random Forest, SVM and Adaboost). We calculated both classwise and overall accuracy for each of these classifiers. We applied one oversampling technique (SMOTE), one undersampling technique (RUS) and our proposed approach on these classifiers and compared their results.

We used sci-kit library of python for implementing the classifiers for our experiment. Clustering method is implemented by ourself in python programming language. We used three imbalanced datasets (us_crime, ecoli and libras move) from "imblearn" library of python. Library function train_test_split is used to create test data set. Among all the data points, 80% data is used for training and the rest of the data is used for testing. Random state for train-test-split was 42 and random state for undersampling was 7. We set the number of clusters as the number of classes in the dataset. In our experiment all the datasets are the binary class, so we set the number of clusters as 2.

5 NUMERICAL RESULTS

This section presents and discusses the results of our experiment. We assigned variable penalties for false-negative and false-positive results and calculated overall cost for the classification. The failure to detect threats (False-negative) is assigned more penalty than misclassifying a benign sample (False-positive). True-negative and true-positive results do not add any cost. A higher penalty P is assigned for false-negative prediction. This value can be set depending on the impact of misclassifying minority sample. For our experiment we used $P = 10$ and $P = 50$. Table 1 shows the cost matrix used in our experiment.

Table 1: Cost matrix

	prediction $y = 1$	prediction $y = 0$
label $h(x) = 1$	$C_{1,1} = 0$	$C_{0,1} = P$
label $h(x) = 0$	$C_{1,0} = 1$	$C_{0,0} = 0$

*0 = Clean Sample and 1 = Infected Sample

Classification accuracy of conventional classifiers is presented in Table 2. Result shows that shows the four mainstream classifiers have very high accuracy for the majority class and achieves a good overall accuracy. But the performance of classifiers on minority classes is very poor compared to the performance of classifier on majority class. Support Vector Machine (SVM) classifier provides comparatively better results among these four mainstream classifiers.

Table 2: Classification accuracy and cost of conventional classifiers

Dataset	Class	Rnd. Forest	KNN	Adaboost	SVM
Us-Crime	Majority	98.92	98.92	97.29	99.19
	Minority	30.00	26.67	43.33	36.67
	Overall	93.73	93.48	93.23	94.49
Ecoli	Majority	98.36	96.72	96.72	96.72
	Minority	14.29	85.71	28.57	85.71
	Overall	89.71	95.59	89.71	95.59
Libras Move	Majority	100.00	100.00	100.00	100.00
	Minority	20.00	40.00	80.00	80.00
	Overall	94.44	95.83	98.61	98.61

Table 3 shows the accuracy of SMOTE, RUS and our proposed methods combined with the previously mentioned four classifiers. RUS obtains higher accuracy for minority classes in many cases, but it has a lower accuracy rate for majority class. As majority class has large number of points, so small drop in majority class Our proposed method has higher accuracy for majority classes than RUS in most of the cases and it improves the accuracy on minority class compared to conventional classifying methods. SMOTE provides better accuracy for the majority class but the performance on minority class is worse than both of RUS and our proposed method.

Table 3: Classification accuracy with RUS, SMOTE and our proposed method

Dataset	Class	Rnd. Forest	KNN	Adaboost	SVM
Applying Random Undersampling (RUS)					
Us Crime	Majority	81.30	80.22	80.49	82.38
	Minority	86.67	93.33	83.33	93.33
	Overall	81.70	81.20	80.70	83.21
Ecoli	Majority	83.61	78.69	83.61	85.25
	Minority	57.14	100.00	71.43	85.71
	Overall	80.88	80.88	82.35	85.29
Libras Move	Majority	88.06	91.04	71.64	95.52
	Minority	100.00	100.00	80.00	100.00
	Overall	94.44	91.67	72.22	95.83
Applying SMOTE					
Us-Crime	Majority	94.04	82.11	91.06	90.24
	Minority	63.33	80.00	60.00	83.33
	Overall	91.73	81.95	88.72	89.72
Ecoli	Majority	95.08	91.80	93.44	90.16
	Minority	85.71	85.71	71.43	85.71
	Overall	94.12	91.18	91.18	89.71
Libras Move	Majority	100.00	95.52	100.00	97.01
	Minority	60.00	100.00	80.00	80.00
	Overall	97.22	95.83	98.61	95.83
Applying our proposed method					
Us-Crime	Majority	89.70	87.26	83.74	87.80
	Minority	73.33	76.67	83.33	90.00
	Overall	88.47	86.47	83.71	87.97
Ecoli	Majority	95.08	91.80	86.89	93.44
	Minority	71.43	85.71	57.14	85.71
	Overall	92.65	91.18	83.82	92.65
Libras Move	Majority	94.03	86.57	85.07	92.54
	Minority	80.00	100.00	100.00	100.00
	Overall	93.06	87.50	98.61	93.06

We calculated costs for all our experiment for both $P = 10$ and $P = 50$ and the result is presented in Table 4. For all of the cases penalty of misclassifying majority class is set to 1. These values can be set based on the priority of classes and overall cost will change accordingly.

Table 4: Cost for various classifying methods

Dataset	Penalty	Rnd. Forest	KNN	Adaboost	SVM
Cost for conventional methods					
Us-Crime	P = 10	214	224	180	193
	P = 50	1054	1104	860	953
Ecoli	P = 10	61	12	52	12
	P = 50	301	52	252	52
Libras Move	P = 10	40	30	10	10
	P = 50	200	150	50	50
Cost for RUS					
Us-Crime	P = 10	109	93	122	85
	P = 50	269	173	322	165
Ecoli	P = 10	40	13	30	19
	P = 50	160	13	110	59
Libras Move	P = 10	8	6	29	3
	P = 50	8	6	69	3
Cost for SMOTE					
Us-Crime	P = 10	132	126	153	86
	P = 50	572	366	633	286
Ecoli	P = 10	13	15	24	16
	P = 50	53	55	104	56
Libras Move	P = 10	20	3	10	12
	P = 50	100	3	50	52
Cost for our proposed method					
Us-Crime	P = 10	118	117	110	75
	P = 50	438	397	310	195
Ecoli	P = 10	23	15	38	14
	P = 50	103	55	158	54
Libras Move	P = 10	14	9	10	5
	P = 50	54	9	10	5

6 CONCLUSION

From our experiment we can conclude that the mainstream classifiers fail in detecting cyber threats. Undersampling method (RUS) and oversampling method (SMOTE) provide comparatively better solution. But in some cases the solution provided by these methods is achieved in the expense of the majority class. The goal is to obtain higher accuracy in the minority class without sacrificing too much in the majority class. Our proposed method obtains relatively higher accuracy in minority classes without sacrificing too much in majority class. This motivates the development of new sophisticated algorithms or modifying existing algorithms to deal with imbalanced dataset more efficiently. Data level pre-processing and combining unsupervised and supervised learning techniques may provide a better solution as well.

REFERENCES

[1] Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, and Konrad Rieck. 2014. Drebin: Effective and explainable detection of android malware in your pocket. In *28th Annual Network and Distributed System Security Symposium (NDSS)*.

[2] A.M. Bagirov, J. Ugon, and D. Webb. 2011. An efficient algorithm for the incremental construction of a piecewise linear classifier. *Information Systems* 36 (2011), 782 – 790.

[3] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Int. Res.* 16, 1 (June 2002), 321–357. http://dl.acm.org/citation.cfm?id=1622407.1622416

[4] Longting Chen, Guanghua Xu, Qing Zhang, and Xun Zhang. 2019. Learning deep representation of imbalanced SCADA data for fault detection of wind turbines. *Measurement* 139 (2019), 370 – 379. https://doi.org/10.1016/j.measurement.2019.03.029

[5] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[6] Sara Fotouhi, Shahrokh Asadi, and Michael W. Kattan. 2019. A comprehensive data level analysis for cancer diagnosis on imbalanced data. *Journal of Biomedical Informatics* 90 (2019), 103089. https://doi.org/10.1016/j.jbi.2018.12.003

[7] H. Guo, Y Li, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications* 73 (2017), 220 – 239. https://doi.org/10.1016/j.eswa.2016.12.035

[8] He Haibo. 2013. Introduction. In *Imbalanced Learning: foundations, algorithms, and applications*. Wiley-Blackwell, Chapter 1, 1–12. https://doi.org/10.1002/9781118646106.ch1 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118646106.ch1

[9] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Advances in Intelligent Computing*, De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 878–887.

[10] H. He, Y. Bai, E. A. Garcia, and S. Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

[11] Michał Koziarski, Bartosz Krawczyk, and Michał Woźniak. 2019. Radial-Based oversampling for noisy imbalanced data classification. *Neurocomputing* 343 (2019), 19 – 33. https://doi.org/10.1016/j.neucom.2018.04.089 Learning in the Presence of Class Imbalance and Concept Drift.

[12] A.Y. Chung Liu. 2004. The effect of oversampling and undersampling on classifying imbalanced text datasets. *The University of Texas at Austin* (2004).

[13] X. Y. Liu, J. Wu, and Z. H. Zhou. 2009. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 2 (April 2009), 539–550.

[14] A. Sarmanova and S. Albayrak. 2013. Alleviating class imbalance problem in data mining. In *21st Signal Processing and Communications Applications Conference (SIU)*. 1–4.

[15] Michael Spreitzenbarth, Florian Echtler, Thomas Schreck, Felix C. Freiling, and Johannes Hoffmann. 2013. Mobilesandbox: Looking deeper into android applications. In *28th International ACM Symposium on Applied Computing (SAC)*.

[16] S. Vluymans, D.S. Tarraga, Y. Saeys, Ch. Cornelis, and F. Herrera. 2016. Fuzzy rough classifiers for class imbalanced multi-instance data. *Pattern Recognition* 53 (2016), 36 – 45. https://doi.org/10.1016/j.patcog.2015.12.002