On the impact of network data balancing in cybersecurity applications

Marek Pawlicki[1,2], Michał Choraś[1,2], Rafał Kozik[1,2], and Witold Hołubowicz[2]

[1] ITTI Sp. z o.o., Poznań
[2] UTP University of Science and Technology, Bydgoszcz, Poland
chorasm@utp.edu.pl

Abstract. Machine learning methods are now widely used to detect a wide range of cyber attacks. However, they come with challenges of their own - one of those problems lies in network dataset characteristics: it should be well-balanced in terms of percentage of malicious data vs. normal not-attacked / benign traffic to achieve adequate results. Since usually in network traffic data there are significantly fewer malicious samples than benign samples, in this work we address the problem of learning from imbalanced network traffic data in the cybersecurity domain.

Keywords: Data imbalance · Machine Learning · Classifiers · Cybersecurity

1 Introduction

The importance of cybersecurity rises with every passing year, along with the the number of connected individuals and the growing number of devices utilising the Internet for various purposes [1] [2]. The antagonistic forces, be it hackers, crackers, state-sponsored cyberforces or a range of other malicious actors employ a variety of methods to cause harm to common users and critical infrastructure alike [3][4]. The massive loads of data transmitted every single second exceeded the human capacity to deal with them long time ago. Thus, a myriad of machine learning (ML) methods were successfully implemented in the domain [5] [6] [7]. As rewarding as they are, AI-related approaches come with their own set of problems. One of them is the susceptibility to data imbalance.

The data imbalance problem refers to a situation in which one or multiple classes have significantly more learning samples as compared to the remaining classes. This often results in misclassification of the minority samples by a substantial number of classifiers, a predicament especially pronounced if the minority classes are the ones that bear the greatest importance - like malignant cancer samples, fraud events, or, as in the case of this work, network intrusions. Additionally, the deterioration of a given model might go unnoticed if the method is only evaluated on the basis of accuracy.

With the significance of the above-mentioned difficulty in plenty of high-stake practical settings, various methods to counter that issue have been proposed.

These fall roughly into three categories: undersampling, oversampling and cost-sensitive methods. In this work numerous approaches to dataset balancing are examined, the influence each method has on a number of ML classifiers is highlighted and in conclusion the best experimentally found approach in the case of network intrusion detection is chosen.

The major contribution and the unique value presented in this work comes in the form of highlighting the notion that the impact dataset balancing methods have on the behaviour of ML classifiers is not always a straightforward and intuitive one. A number of balancing approaches is thoroughly evaluated and their impact on both the dataset and the behaviour of classifiers is showcased. All of this in the context of a practical, vital domain that is network intrusion detection.

The paper is structured as follows: in Section 2 the pipeline of network intrusion detection is illustrated and described, and the ML algorithms utilised are succinctly introduced, in Section 3 the chosen balancing methods are characterised. Section 4 expresses the experiments undertaken and finally Section 5 showcases the obtained results.

Table 1: Encoded labels and number of instances in Intrusion Detection Evaluation Dataset used in this work (see Section 4)

No of training instances	Class Label	Encoded label
1459377	BENIGN	0
207112	DoS Hulk	4
142924	PortScan	9
115222	DDoS	2
9264	DoS GoldenEye	3
7141	FTP-Patator	7
5307	SSH-Patator	10
5216	DoS slowloris	6
4949	DoS Slowhttptest	5
2713	Web Attack Brute Force	11
1760	Bot	1
1174	Web Attack XSS	13
38	Web Attack SQL Injection	12
10	Heartbleed	8

2 Machine Learning Approach Enhanced with Data Balancer

The focus of this research lies on the impact the balance of the instance numbers among classes in a dataset has on the performance of ML-based classification methods. In general, the step-by-step process of ML-based Intrusion Detection System (IDS) can be succinctly summarised as follows: a batch of annotated data is used to train a classifier. The algorithm 'fits' to the training data, creating a model. This is followed by testing the performance of the acquired model on the testing set - a batch of unforeseen data. In order to alleviate the data balancing problem present in the utilised IDS dataset an additional step is undertaken before the algorithm is trained (as seen in Fig. 1).

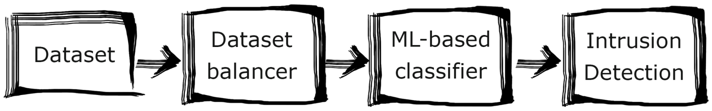


Fig. 1: IDS training pipeline with dataset balancing

The ML-based classifier block of Fig. 1 can be realised by an abundance of different machine learning methods. In fact, recent research showcases numerous novel approaches including deep learning [8][7], ensemble learning [9][10], various augmentations to classical ML algorithms [11] etc. In this work three basic models were chosen to put emphasis on the data balancing part. These are:

- Artificial Neural Network [12][13]
- Random Forest [14]
- Naive Bayes [15]

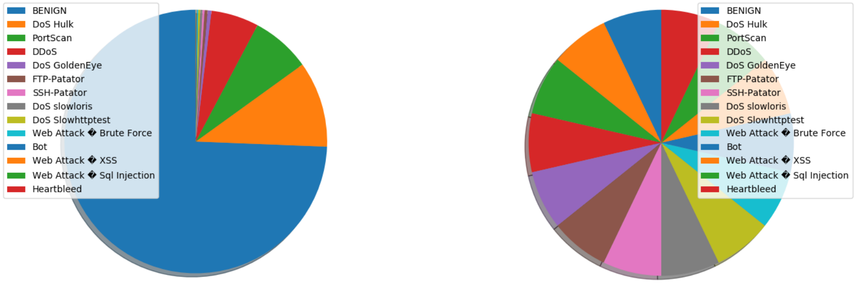
These represent three significantly different approaches to machine learning and were selected to cover possibly the widest range of effects dataset balancing could have on the effectiveness of ML.

3 Balancing Methods

In the cases suffering from the data imbalance problem the number of training samples belonging to some classes is larger in contrast to other classes.

The conundrum of data imbalance has recently been deeply studied in the area of machine learning and data mining. In numerous cases, this predicament impacts the machine learning algorithms and in result deteriorates the effectiveness of the classifier [16]. Typically in such cases, classifiers will achieve higher predictive accuracy over the majority class, but poorer predictive accuracy over the minority class. In general, solutions to this problem can be categorised as (i) data-related, and (ii) algorithm-related.

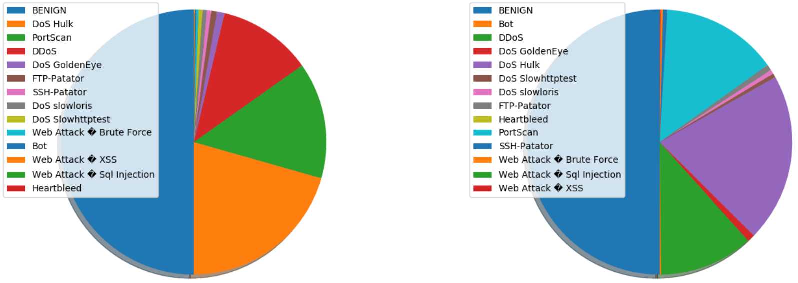
In the following paragraphs, these two categories of balancing methods will be briefly introduced. The focus of the analysis was on the practical cybersecurity-related application that faces the data imbalance problem.



Original Distribution

After SMOTE

Fig. 2: Class distribution in CICIDS 2017 - Original unbalanced distribution and after SMOTE



BENIGN class randomly subsampled

NearMiss

Fig. 3: Class distribution in CICIDS 2017 - After performing random undersampling and NearMiss

3.1 Data-related Balancing Methods

Two techniques, belonging to this category, that are commonly used to cope with imbalanced data use the principle of acquiring a new dataset out of the existing one. This is realised with data sampling approaches. There are two widely recognised approaches called data over-sampling and under-sampling.

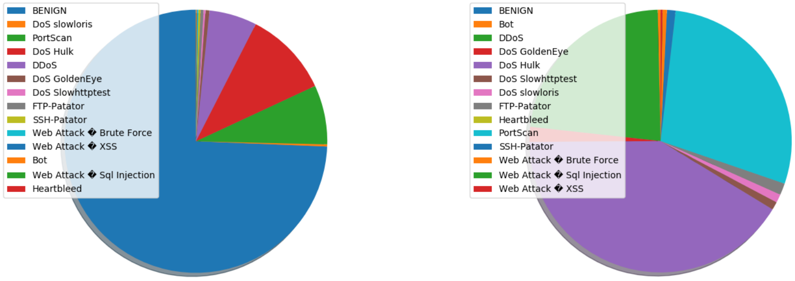
Under-sampling balances the dataset by decreasing the size of the majority class. This method is adopted when the number of elements belonging to the majority class is rather high. In that way, one can keep all the samples belonging to the minority class and randomly (or not) select the same number of elements representing the majority class. In our experiments we tried a number of undersampling approaches, one of those was **Random Sub-sampling**. The effect random subsampling has on the dataset is illustrated in Fig. 3. The results the method has in conjunction with the selected ML algorithms is showcased in Tab. 3

Table 2: CICIDS2017 (full set) / Unbalanced

	ANN			RandomForest			NaiveBayes			
	ACC: 0.9833			ACC: 0.9987			ACC: 0.2905			
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	support
0	0.99	0.99	0.99	1.00	1.00	1.00	1.00	0.10	0.18	162154
1	0.97	0.35	0.52	0.88	0.68	0.77	0.01	0.65	0.01	196
2	1.00	1.00	1.00	1.00	1.00	1.00	0.94	0.95	0.94	12803
3	0.99	0.97	0.98	1.00	0.99	1.00	0.09	0.93	0.16	1029
4	0.95	0.94	0.94	1.00	1.00	1.00	0.74	0.70	0.72	23012
5	0.89	0.98	0.93	0.96	0.98	0.97	0.00	0.67	0.01	550
6	0.99	0.98	0.99	1.00	0.99	0.99	0.05	0.52	0.09	580
7	0.99	0.98	0.99	1.00	1.00	1.00	0.10	0.99	0.18	794
8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1
9	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	15880
10	1.00	0.49	0.66	1.00	1.00	1.00	0.08	0.99	0.15	590
11	0.85	0.10	0.17	0.86	0.99	0.92	0.00	0.07	0.00	301
12	0.00	0.00	0.00	1.00	1.00	1.00	0.01	1.00	0.02	4
13	1.00	0.02	0.05	0.95	0.61	0.74	0.08	0.93	0.14	130
macro avg	0.90	0.70	0.73	0.97	0.95	0.96	0.36	0.75	0.33	218024
weighted avg	0.98	0.98	0.98	1.00	1.00	1.00	0.95	0.29	0.34	218024

Table 3: CICIDS2017 (full set) / Random Subsampling

	ANN			RandomForest			NaiveBayes			
	ACC: 0.9812			ACC: 0.9980			ACC: 0.2911			
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	support
0	1.00	0.98	0.99	1.00	1.00	1.00	1.00	0.10	0.18	162154
1	0.50	0.63	0.56	0.91	0.92	0.91	0.01	0.65	0.01	196
2	1.00	1.00	1.00	1.00	1.00	1.00	0.94	0.95	0.94	12803
3	0.98	0.98	0.98	1.00	1.00	1.00	0.09	0.93	0.16	1029
4	0.90	0.99	0.95	1.00	1.00	1.00	0.74	0.70	0.72	23012
5	0.90	0.99	0.94	0.98	0.99	0.99	0.00	0.67	0.01	550
6	0.97	0.98	0.97	0.99	0.99	0.99	0.05	0.52	0.09	580
7	0.99	0.98	0.98	1.00	1.00	1.00	0.10	0.99	0.19	794
8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1
9	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	15880
10	0.97	0.49	0.65	1.00	0.99	1.00	0.08	0.99	0.15	590
11	0.59	0.23	0.33	0.80	0.97	0.88	0.00	0.07	0.00	301
12	0.00	0.00	0.00	1.00	0.80	0.89	0.01	1.00	0.02	4
13	0.80	0.03	0.06	0.96	0.40	0.57	0.08	0.93	0.15	130
macro avg	0.83	0.73	0.74	0.97	0.93	0.94	0.36	0.75	0.33	218024
weighted avg	0.99	0.98	0.98	1.00	1.00	1.00	0.95	0.29	0.34	218024



After cleaning the Tomek-links

Cluster-Centers undersampling

Fig. 4: Class distribution in CICIDS 2017 - After cleaning the Tomek-Links and performing ClusterCenters undersampling

There are also approaches that introduce some heuristics to the process of sampling selection. The algorithm called **NearMiss** [17] is one of them. This approach engages algorithm for nearest neighbours analysis (e.g. k-nearest neighbour) in order to select the dataset instances to be under-sampled. The NearMiss algorithm chooses these samples for which the average distance to the closest samples of the opposite class is the smallest. The effect the algorithm has on the dataset is illustrated in Fig. 3, the results obtained are found in Tab. 4

Another example of algorithms falling into the undersampling category is called **TomekLinks** [18]. The method performs under-sampling by removing Tomek's links. Tomek's link exists if the two samples are the nearest neighbours of each other. More precisely, A Tomek's link between two samples of different class x and y is defined as $d(x,y) < d(x,z)$ and $d(x,y) < d(y,z)$ for any sample z. The effect removing Tomek-links has on the dataset is illustrated in Fig.4, the effect it has on ML models is found in Tab.5.

A different approach to under-sampling involves centroids obtained from a clustering method. In that type of algorithms the samples belonging to majority class are first clustered (e.g. using k-means algorithm) and replaced with the cluster centorids. In the experiments this approach is indicated as **Cluster Centroids**. The results of the clustering procedure are illustrated in Fig. 4 and in Tab. 6

On the other hand, the oversampling method is to be adopted when the size of the original dataset is relatively small. In that approach, one takes the minority class and increases its cardinality in order to achieve the balance among classes. This can be done by using a technique like bootstrapping. In that case, the minority class is sampled with repetitions. Another solution is to use **SMOTE** (Synthetic Minority Over-Sampling Technique)[19]. There are various modification to the original SMOTE algorithm. The one evaluated in this paper is named **Borderline SMOTE**. In this approach the samples representing the minority class are first categorised into three groups: danger, safe, and noise. The sample x is considered to belong to category *noise* if all nearest-neighbours of x are from

Table 4: CICIDS2017 (full set) / NearMiss

	ANN			RandomForest			NaiveBayes			
	ACC: 0.7725			ACC: 0.7116			ACC: 0.3744			
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	support
0	1.00	0.70	0.82	1.00	0.61	0.76	1.00	0.21	0.35	162154
1	0.02	0.71	0.04	0.03	0.81	0.06	0.01	1.00	0.02	196
2	0.90	1.00	0.95	0.52	1.00	0.68	0.91	0.96	0.93	12803
3	0.99	0.97	0.98	0.97	0.99	0.98	0.22	0.93	0.35	1029
4	0.66	1.00	0.80	0.51	1.00	0.68	0.65	0.70	0.68	23012
5	0.58	0.99	0.73	0.57	0.98	0.72	0.00	0.64	0.01	550
6	0.27	0.98	0.43	0.07	0.99	0.13	0.07	0.82	0.13	580
7	0.19	1.00	0.32	0.25	1.00	0.40	0.10	1.00	0.18	794
8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1
9	0.45	1.00	0.62	0.89	1.00	0.94	1.00	0.99	0.99	15880
10	0.12	0.99	0.21	0.07	1.00	0.13	0.11	0.99	0.20	590
11	0.35	0.56	0.43	0.09	0.99	0.16	0.00	0.08	0.01	301
12	0.00	0.00	0.00	0.05	1.00	0.10	0.01	1.00	0.02	4
13	0.01	0.02	0.02	0.06	0.49	0.11	0.17	0.92	0.29	130
macro avg	0.47	0.78	0.52	0.43	0.92	0.49	0.38	0.80	0.37	218024
weighted avg	0.91	0.77	0.81	0.90	0.71	0.75	0.94	0.37	0.46	218024

Table 5: CICIDS2017 (full set) / Tomek Links

	ANN			RandomForest			NaiveBayes			
	ACC: 0.9836			ACC: 0.9986			ACC: 0.5263			
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	support
0	0.99	0.99	0.99	1.00	1.00	1.00	1.00	0.10	0.18	162154
1	0.92	0.37	0.53	0.81	0.78	0.80	0.01	0.65	0.01	196
2	1.00	1.00	1.00	1.00	1.00	1.00	0.94	0.95	0.94	12803
3	0.99	0.97	0.98	1.00	0.99	0.99	0.09	0.93	0.16	1029
4	0.94	0.95	0.95	1.00	1.00	1.00	0.74	0.70	0.72	23012
5	0.90	0.99	0.94	0.97	0.98	0.98	0.00	0.67	0.01	550
6	0.99	0.98	0.98	0.99	0.99	0.99	0.05	0.52	0.09	580
7	0.99	0.98	0.99	1.00	1.00	1.00	0.10	0.99	0.18	794
8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1
9	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	15880
10	1.00	0.49	0.66	1.00	0.99	1.00	0.08	0.99	0.15	590
11	0.85	0.07	0.13	0.84	0.97	0.90	0.00	0.07	0.00	301
12	0.00	0.00	0.00	1.00	0.75	0.86	0.01	1.00	0.02	4
13	1.00	0.02	0.05	0.91	0.55	0.68	0.08	0.93	0.14	130
macro avg	0.90	0.70	0.73	0.97	0.93	0.94	0.36	0.75	0.33	218024
weighted avg	0.98	0.98	0.98	1.00	1.00	1.00	0.95	0.29	0.34	218024

Table 6: CICIDS2017 (full set) / ClusterCentroids

	ANN			RandomForest			NaiveBayes			
	ACC: 0.4569			ACC: 0.2560			ACC: 0.2832			
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	support
0	1.00	0.47	0.64	1.00	0.00	0.00	1.00	0.09	0.16	162154
1	0.01	0.28	0.02	0.03	1.00	0.07	0.01	0.65	0.01	196
2	0.90	0.63	0.74	0.74	1.00	0.85	0.93	0.95	0.94	12803
3	0.77	0.68	0.72	0.75	1.00	0.85	0.08	0.93	0.16	1029
4	0.86	0.62	0.72	0.81	1.00	0.90	0.67	0.70	0.69	23012
5	0.15	0.69	0.25	0.82	0.99	0.89	0.00	0.67	0.01	550
6	0.35	0.22	0.27	0.25	0.99	0.40	0.05	0.52	0.09	580
7	0.06	0.47	0.11	0.71	1.00	0.83	0.10	0.99	0.18	794
8	0.01	1.00	0.01	0.50	1.00	0.67	1.00	1.00	1.00	1
9	0.57	0.00	0.00	1.00	1.00	1.00	1.00	0.99	0.99	15880
10	0.00	0.00	0.00	0.10	1.00	0.18	0.08	0.99	0.15	590
11	0.00	0.00	0.00	0.17	0.98	0.29	0.00	0.07	0.00	301
12	0.00	0.00	0.00	0.18	1.00	0.31	0.01	1.00	0.02	4
13	0.00	0.03	0.00	0.05	0.65	0.09	0.09	0.93	0.16	130
macro avg	0.34	0.36	0.25	0.51	0.90	0.52	0.36	0.75	0.33	218024
weighted avg	0.93	0.46	0.60	0.95	0.26	0.23	0.94	0.28	0.32	218024

a different class than the analysed sample, *danger* when only a half belongs to different class, and *safe* when all nearest-neighbours are from the same class. In Borderline SMOTE algorithm, only the *safe* data instances are over-sampled [20]. The effect of this procedure on the dataset is expressed in Fig. 2. The results are placed in Tab.7

A final note concluding this section would be the observation that there is no silver bullet putting one sampling method over another. In fact, their application depends on the use case scenarios and the dataset itself. For the sake of clear illustration the original dataset's class distribution is depicted in Fig. 2, the results the ML algorithms have achieved are found in Tab.1

3.2 Algorithm-related Balancing Methods

Utilizing unsuitable evaluation metrics for the classifier trained with the imbalanced data can lead to wrong conclusions about the classifier's effectiveness. As the majority of machine learning algorithms do not operate very well with imbalanced datasets, the commonly observed scenario would be the classifier totally ignoring the minority class. This happens because the classifier is not sufficiently penalized for the misclassification of the data samples belonging to the minority class. This is why the algorithm-related methods have been introduced as a part of the modification to the training procedures. One technique is to use other performance metrics. The alternative evaluation metrics that are suitable for imbalanced data are:

Table 7: CICIDS2017 (full set) / BORDERLINE SMOTE

	ANN			RandomForest			NaiveBayes			
	ACC: 0.9753			ACC: 0.9920			ACC: 0.5263			
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	support
0	1.00	0.97	0.99	1.00	0.99	1.00	1.00	0.52	0.68	162154
1	0.17	0.94	0.29	0.17	0.98	0.30	0.00	0.65	0.01	196
2	0.99	1.00	1.00	1.00	1.00	1.00	0.85	0.95	0.90	12803
3	0.94	0.99	0.96	1.00	1.00	1.00	0.05	0.87	0.10	1029
4	0.93	0.99	0.96	0.99	1.00	1.00	0.68	0.70	0.69	23012
5	0.64	0.96	0.77	0.94	0.98	0.96	0.01	0.20	0.02	550
6	0.78	0.51	0.62	1.00	0.97	0.98	0.01	0.03	0.01	580
7	0.87	0.98	0.92	0.99	1.00	1.00	0.02	0.47	0.05	794
8	0.50	1.00	0.67	1.00	1.00	1.00	1.00	1.00	1.00	1
9	0.99	1.00	0.99	1.00	1.00	1.00	0.01	0.00	0.00	15880
10	0.64	0.53	0.58	1.00	0.89	0.94	0.07	0.50	0.12	590
11	0.20	0.26	0.22	0.85	0.84	0.84	0.02	0.89	0.05	301
12	0.01	0.75	0.01	1.00	1.00	1.00	0.01	1.00	0.03	4
13	0.16	0.77	0.27	0.67	0.90	0.77	0.00	0.00	0.00	130
macro avg	0.63	0.83	0.66	0.90	0.97	0.91	0.27	0.56	0.26	218024
weighted avg	0.98	0.98	0.98	1.00	0.99	0.99	0.87	0.53	0.64	218024

- precission - indicating the percentage of relevant data samples that have been collected by the classifier
- recall (or sensitivity)- indicating the total percentage of all relevant instances that have been detected.
- f1-score - computed as the harmonic mean of precision and recall.

Another technique that is successfully used in the field is a cost-sensitive classification. Recently this learning procedure has been reported to be an effective solution to class-imbalance in the large-scale settings. Without losing the generality, let us define the cost-sensitive training process as the following optimisation formula:

$$\hat{\theta} = \min_{\theta} \left\{ \frac{1}{2}||\theta||^2 + \frac{1}{2}\sum_{i=1}^{N} C_i||e_i||^2 \right\} \tag{1}$$

where θ indicates the classifier parameters, e_i the error in the classifier response for the i-th (out of N) data samples, and C_i the importance of the i-th data sample.

In cost-sensitive learning, the idea is to give a higher importance C_i to the minority class, so that the bias towards the majority class is reduced. In other words, we are producing a cost function that is penalizing the incorrect classification of the minority class more than incorrect classifications of the majority class.

In this paper we have focused on **Cost-Sensitise Random Forest** as an example of cost-sensitive meta-learning. This is mainly due to the fact the Random

Forest classifier in that configuration yields the most promising results. These can be found in Tab. 10

4 Experiments and Results

4.1 Dataset Description - Intrusion Detection Evaluation Dataset - CICIDS2017

CICIDS2017 [21] is an effort to create a dependable and recent cybersec dataset. The Intrusion Detection datasets are notoriously hard to come by, and the ones available display at least one of frustrationg concerns, like the lack of traffic diversity, attack variety, insufficient features etc. The authors of CICIDS2017 offer a dataset with realistic benign traffic, created as an interpolation of the behaviour of 25 users using multiple protocols. The dataset is a labelled capture of 5 days of work, with 4 days putting the framework under siege by a plethora of attacks, including malware, DoS attacks, web attacks and others. This work relies on the captures from Tuesday, Wednesday, Thursday and Friday. CICIDS2017 constitutes one of the newest datasets available to researchers, featuring over 80 network flow characteristics. The Imbalance Ratio of the Majority Class to the sum of all the numbers of samples of the rest of the classes was calculated to be 2.902. The sample counts for particular classes in the training set are showcased in Tab. 1

4.2 Results and Perspectives

CICIDS 2017 dataset consists of 13 classes - 12 attacks and 1 benign class. As depicted in Fig. 2, there is a wide discrepancy among the classes in terms of the number of instances, especially the benign class as compared to the attack classes. The number of instances in the respective classes in the training set is displayed in Tab. 1.

During the tests the initial hypothesis was that balancing the classes would improve the overall results. Random Subsampling (Tab. 3) along a slew of other subsampling methods were used to observe the influence dataset balancing has on the performance of 3 reference ML algorithms - an Artificial Neural Network (ANN), a RandomForest algorithm and a Naive Bayes classifier. Finally, Borderline SMOTE was conducted as a reference oversampling method. The results of those tests are to be witnessed in Tab. 4, 7, 5 and 6. It is immediately apparent from inspecting the recall in the unbalanced dataset (Tab. 1) that some instances of the minority classess are not recognised properly (class 1 and 13). Balancing the benign class to match the number of samples of all the attacks combined changed both the precision and the recall achieved by the algorithm. It also became apparent that none of the subsampling approaches outperformed simple random subsampling in the case of CICIDS2017. The tests revealed an interesting connection among the precision, recall and the imbalance ratio of the dataset. Essentially, there seems to exist a tradeoff between precision and recall that can be controlled by the number of the instances of classes in the training dataset. To evaluate that assertion further tests were conducted. Random Forest algorithm was trained on the Unbalanced dataset and then all the classess were subsampled to match the number of samples in one of the minority classes (Tab. 9 - 1174 instances per class and Tab. 8 - 7141 instances per class).

Table 8: CICIDS2017 / Random Subsampling down to 7141 instances per class / RandomForest

	precision	recall	f1-score	support
0	1.00	0.98	0.99	162154
1	0.13	0.99	0.23	196
2	1.00	1.00	1.00	12803
3	0.92	1.00	0.96	1029
4	0.98	1.00	0.99	23012
5	0.85	0.99	0.92	550
6	0.93	0.99	0.96	580
7	0.93	1.00	0.96	794
8	0.17	1.00	0.29	1
9	1.00	1.00	1.00	15880
10	0.73	1.00	0.85	590
11	0.63	0.98	0.77	301
12	0.07	1.00	0.14	4
13	0.32	0.48	0.39	130
accuracy			0.9872	218024
macro avg	0.69	0.96	0.74	218024
weighted avg	0.99	0.99	0.99	218024

The tests proved that changing the balance ratio undersampling the majority classes improves the recall of the minority classes, but degrades the precission of the classifier on those classes. This basically means that dataset balancing causes the ML algorithms to misclassify the (previously) majority classes as instances of the minority classes, thus boosting the false positives.

Finally, a cost-sensitive random forest algorithm was tested. After trying different weight setups results exceeding any previous undersampling or oversampling methods were attained (Tab. 10). It is noteworthy that the achieved recall for class 13 is higher while still retaining a relatively high precision. A relationship between class 11 and class 13 was also discovered, where setting a higher weight for class 13 would result in misclassification of class 11 samples as class 13 samples and the other way round.

5 Conclusions

In this paper the evaluation of a number of dataset balancing methods for the ML algorithms in the cybersecurity doman was presented. The conducted experiments revealed a number of interesting details about those methods. Firstly,

Table 9: CICIDS2017 / Random Subsampling down to 1174 instances per class / RandomForest

	precision	recall	f1-score	support
0	1.00	0.96	0.98	162154
1	0.07	1.00	0.13	196
2	0.99	1.00	1.00	12803
3	0.69	1.00	0.82	1029
4	0.94	0.99	0.97	23012
5	0.76	0.99	0.86	550
6	0.86	0.99	0.92	580
7	0.81	1.00	0.89	794
8	0.17	1.00	0.29	1
9	1.00	1.00	1.00	15880
10	0.44	1.00	0.61	590
11	0.23	0.65	0.34	301
12	0.07	1.00	0.13	4
13	0.13	0.95	0.23	130
accuracy			0.9657	218024
macro avg	0.58	0.97	0.65	218024
weighted avg	0.99	0.97	0.97	218024

Table 10: CICIDS2017 / Cost-Sensitive RandomForest

	precision	recall	f1-score	support
0	1.00	1.00	1.00	162154
1	0.34	0.91	0.50	196
2	1.00	1.00	1.00	12803
3	1.00	0.99	0.99	1029
4	1.00	1.00	1.00	23012
5	0.97	0.98	0.97	550
6	1.00	0.99	0.99	580
7	1.00	1.00	1.00	794
8	1.00	1.00	1.00	1
9	1.00	1.00	1.00	15880
10	1.00	1.00	1.00	590
11	0.98	0.85	0.91	301
12	1.00	1.00	1.00	4
13	0.72	0.96	0.83	130
accuracy			0.9973	218024
macro avg	0.93	0.98	0.94	218024
weighted avg	1.00	1.00	1.00	218024

in the case of the CICIDS2017 dataset, random subsampling was just as good or better than other undersampling methods and the results were on par with Borderline SMOTE. Secondly, the final proportions of the dataset can bear just as much an impact on the results of ML classification as the choice of the balancing procedure itself. Thirdly, there is a relationship among the size of the majority classes, the precision and the recall achieved, which is simply expressed by the number of majority samples falsely classified as minority samples. Finally, for CICIDS2017 the best results were achieved by using the cost-sensitive approach and a random forest algorithm.

Acknowledgement

This work is funded under the SPARTA project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892.

References

1. G. Parekh, D. DeLatte, G. L. Herman, L. Oliva, D. Phatak, T. Scheponik, and A. T. Sherman. Identifying core concepts of cybersecurity: Results of two delphi processes. *IEEE Transactions on Education*, 61(1):11–20, Feb 2018.
2. A. Tabasum, Z. Safi, W. AlKhater, and A. Shikfa. Cybersecurity issues in implanted medical devices. In *2018 International Conference on Computer and Applications (ICCA)*, pages 1–9, Aug 2018.
3. D. Bastos, M. Shackleton, and F. El-Moussa. Internet of things: A survey of technologies and security risks in smart home and city environments. In *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, pages 1–7, March 2018.
4. Rafał Kozik, Michał Choraś, Massimo Ficco, and Francesco Palmieri. A scalable distributed machine learning approach for attack detection in edge computing environments. *Journal of Parallel and Distributed Computing*, 119:18–26, 2018.
5. M. Sewak, S. K. Sahay, and H. Rathore. Comparison of deep learning and the classical machine learning algorithm for the malware detection. In *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 293–296, June 2018.
6. Michał Choraś and Rafał Kozik. Machine learning techniques applied to detect cyber attacks on web applications. *Logic Journal of the IGPL*, 23(1):45–56, 2015.
7. K. Özkan, Ş. Işık, and Y. Kartal. Evaluation of convolutional neural network features for malware detection. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pages 1–5, March 2018.
8. K. D. T. Nguyen, T. M. Tuan, S. H. Le, A. P. Viet, M. Ogawa, and N. L. Minh. Comparison of three deep learning-based approaches for iot malware detection. In *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, pages 382–388, Nov 2018.
9. Ying Wang, Yongjun Shen, and Guidong Zhang. Research on intrusion detection model using ensemble learning methods. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 422–425, Aug 2016.

10. R. Kumar Singh Gautam and E. A. Doegar. An ensemble approach for intrusion detection system using machine learning algorithms. In *2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 14–15, Jan 2018.
11. Kunal and M. Dua. Machine learning approach to ids: A comprehensive review. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 117–121, June 2019.
12. Sandro Skansi. *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer, 2018.
13. H. A. Sonawane and T. M. Pattewar. A comparative performance evaluation of intrusion detection based on neural network and pca. In *2015 International Conference on Communications and Signal Processing (ICCSP)*, pages 0841–0845, April 2015.
14. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.
15. Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook, 2nd ed.* 01 2010.
16. Rafał Kozik and Michał Choraś. Solution to data imbalance problem in application layer anomaly detection systems. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 441–450. Springer, 2016.
17. J. Zhang and I. Mani. KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets*, 2003.
18. Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, Nov 1976.
19. Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Proceedings of the 2005 International Conference on Advances in Intelligent Computing - Volume Part I*, ICIC'05, pages 878–887, Berlin, Heidelberg, 2005. Springer-Verlag.
20. Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, pages 878–887, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
21. Iman Sharafaldin., Arash Habibi Lashkari., and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*, pages 108–116. INSTICC, SciTePress, 2018.