ZKG INTERNATIONAL

ISSN: 2366-1313

Benchmark Of Data Preprocessing Methods for Imbalanced Classification

[1]MR. V.Pranay, [2] Kankala Aravind,[3]Ragamshetty Ravi Teja,[4] Pasula Saikumar Reddy, [5]Sai Vishal Paka

[1] Assistant Professor, Dept. Of CSE, Samskruti College of Engineering & Technology, TS.

[2,3,4,5]B. Tech Student, Dept. Of CSE, Samskruti College of Engineering & Technology, TS.

Abstract: *The large elegance imbalance is one of the most important things that make it difficult to learn cybersecurity gadgets. Some of the rerecorded documents were introduced over time. This technique modifies the training data using oversampling, under sampling, or a combination of both to improve the overall performance of the classes encountered on that data. . Although these methods are used occasionally in cybersecurity, there is a lack of comprehensive, independent standards evaluating their effectiveness across a wide range of cybersecurity issues. This paper provides benchmarks of sixteen preliminary methods on six cybersecurity datasets as well as 17 public datasets from different sources. We test the following strategy in various hyper parameter settings and use the Auto ML gadget to train the class on a priori data, reducing the resources due to the specific choice of hyper parameters or classifiers. Special attention is also given to comparing strategies using comprehensive performance measures that are a true indicator of performance in today's global cybersecurity systems. The main conclusions of our study are: 1) Most of the time, there is a pre-teach that improves the course. 2) The simple do-nothing approach has achieved many of the strategic goals. Third) Oversampling techniques are often more effective than standard sampling. 4) The greatest overall performance has been added using the same old SMOTE algorithm and many complex strategies provide exceptional improvements at the cost of lower performance.*

Keywords: machine learning, cybersecurity, classification, imbalanced classification

I INTRODUCTION

A class of hassle is said to be unbalanced when the beauty before the appearance of at least one elegance, generally the elegance of pleasure, is less than the time

ISSN: 2366-1313

which passes by a certain beauty. Classroom distraction issues appear across the wide range of gadget mastery applications, including medicine [48], finance [47], [58], astronomy [32], and many others.

Particularly, in cybersecurity, to be honest, the most common classroom problems are unsustainable (e.g., cybersecurity [13], malware detection [18], phishing detection [21]). Furthermore, the greatest uncertainty is often too much, with the earliest of expressions of interest being 10-5 and decreasing [13] due to the fact that brutality and criminality are (fortunately) rare. For example, in network telemetry, most people recorded are related to regular (benign) visitors to the site, and only a small element is associated with bad activity. Interestingly, a class error in even a small portion of telemetry is associated with poor performance because the risk associated with poor activity and poor tracking is higher than the best of threats. more serious (for example, remote access to Trojan horses). ransomware, APT). The problem and importance of the class of unexpected vulnerabilities in cybersecurity was, to our knowledge, first mentioned by Axelsson [7] in 2000. Now, more than many years later, an incredible group is still one of the most important. which makes it difficult to acquire cybersecurity knowledge [5], [27].

Although the difference of a little elegance usually has no impact, as soon as it reaches the truth, the gadget has experienced class with the appropriate measurements that cannot be scientifically reliable from the data [31]. In this case, the classifiers will often turn out to be beasts for the greater part of the magnificence and neglect the underrepresented one, which makes the situation more correct, because the classifier predicts most of the people's elegance at all times. However, on the other hand, additional performance measures that reproduce the performance of each instruction are negative.

Over the years, this problem has attracted a lot of interest. Many specific techniques have been proposed to sequence all the important levels of machine learning design. These steps are [6]: 1) truth checking, 2) model training, and 3) model analysis. The practices in the first stage are formerly called data-level processes, while the process performed at the second level is called algorithm-level methods [34]. A number of literature reviews [15], [35], [54], [31], [34] documenting popular concepts and techniques in each phase have been published over time.

In this article, we note the statistical level approach required for the study

ISSN: 2366-1313

unbalanced in terms of sophistication. The idea behind this process is to focus on improving the distribution of educational materials to make it more unequal. This is done, in principle, either by oversampling the minority class or by sampling the magnificent population. Many such ideas have been published over the years, and each time their purpose has been controversial. The current situation regarding which strategies are appropriate to use at this time and which need not be difficult for little or no effect is uncertain. At worst, it can be hoped, the artistic process has been stopped with the help of the favorite field of less trouble or more usual. Our goal in this article is to develop information on the strengths, weaknesses, and various changes (all estimates and calculations) of many of the best-known degree ideas.

To achieve this, we carry out a quantitative analysis of the truth level approach of numerous documents through specialized software with particular dedication to the field of cybersecurity. We aim to evaluate ideas as objectively as possible on the ground, which we support.

II RELATED WORK

Over the years, many preliminary ideas suitable for intellectual conflicts have been published, but on the other hand, there is only a small amount of information that includes many of ideas and information. In general, every report introducing a new system has a test, but the resources of these tests are usually small. For example, a paper presents ADASYN [30] as a test on 5 data sets and compares the model only for SMOTE [16] and the simple selection tree root.

That said, there are already specialized classes that are needed to compare the prior methods, but most of them prefer to know which methods are more efficient than up sampling methods. Most of these studies [26], [3], [10] have also been conducted on large and small-scale datasets. An exception is the study by Kova'cs [36], which is voluminous in terms of comparing techniques and reference materials. However, it only focuses on the oversampling process and there are no tests in the field of cybersecurity. In addition, none of the above studies have done as well as researching the hyper parameters and the complete model as we do.

In the cybersecurity industry, Wheels et al. [59] compared several prior methods to the UNSW-NB15 data set [45]. Bagui and Li [9] compared five prior methods of six input networks to detect information and used a feed-forward neural network with a

ISSN: 2366-1313

hidden layer for classification. Also, the maximum advance of known data Techniques are known and used in cybersecurity [1], [43], [2], [53], [8], but to our knowledge, larger comparative studies are not available.

Finally, previous research adds elements of individual processes into a multidimensional one. In general, this is the grade or average score of the process obtained across all the documents. In this article, we provide a density distribution plot instead of a single number. These charts show a more complete picture because the rankings tend to have high variance and overlapping data.

III BENCHMARKED METHODS

This step includes a set of predefined methods used in testing. For the sake of space, we refrain from reasoning and discuss with the real classes.

A. Oversampling methods

The oversampling method represents a useful way to solve the problem of randomness. The main objective of oversampling techniques is to modify the empirical distribution by increasing the value of the minority sample. Empirical distributions are modified by both copying existing models or creating new models until the desired parameters are satisfied. The most accurate method is called random oversampling, which, as it is called, randomly duplicates already present the sample in the data set.

One of the first and most widely used oversampling techniques that produces accurate samples is SMOTE [16]. He creates new synthetic models taking the lines of existing models starting from the less elegant ones. SMOTE considers, however, that all minimum standards are of equal importance. It does not include previous samples and does not take care of approximating the neighborhood of the sample. Various improvements have been proposed to overcome the shortcomings of the original SMOTE algorithm. We include 4 of these modifications in our evaluation, specifically Borderline SMOTE [28], SVM SMOTE [46], K stands for SMOTE [38].

Border SMOTE, unlike SMOTE, selects models of the smallest people with at least half of their neighbors who are public elegance. The idea behind this approach is that the few samples surrounded by the majority of samples are close to the so-called selection limit and are therefore important in the distribution.

ISSN: 2366-1313

SVM SMOTE builds on the same concept but uses SVM rules in favor of the kNN algorithm to find fewer samples near the decision region.

K means that SMOTE tries to create new synthetic models in areas where the models are weak and prevent further increase in density. It uses K means clustering to find clusters that contain fewer samples than most samples. This avoids interference from non-noisy samples. Then a new pattern is created in each selected group based on its speed, i.e. Larger patterns are created in random groups.

ADASYN differs from SMOTE by giving weight to lower standards based on their learning difficulty. Difficulty mastering, in this example, the way of good close friends who participate in other elegance's. Many statistics are produced in areas where it is difficult to study small samples, and fewer statistics are produced in other areas of study that are less difficult to use.

B. In sampling methods

By sampling most people's consciousness, elegance favors the oversampling strategy, to solve the problem of unbalanced distribution. This strategy reduces the number of samples in most classes to create a more balanced sample of classes.

Most of the following model ideas mentioned here are called model selection methods. The sample selection process reduces the number of samples by eliminating unnecessary samples from the data and using the best primary statistics methods. The Cluster Centrists method is the only example of a clustering method used in statistical analysis. Prototype technology reduces the number of prototypes by creating new models,

For example. Cluster centrists are obtained using the K-means algorithm, instead of using a subset of the true ones.

Again, the best manual method based entirely on selecting and eliminating most of the people sampled is called random sub sampling. The following multiple strategies are based on the kNN algorithm and manipulate it to achieve specific results.

Condensed Nearest Neighbors - CNN [29] reduces the large data set to a fixed set of data which, when used in the 1-NN rule, is divided into all samples from the designed specified data.

Updated Neighbors - ENN [60] divides all samples into a beautiful sub sample by calculating the nearest neighbors for each of the first complete sets.

ISSN: 2366-1313

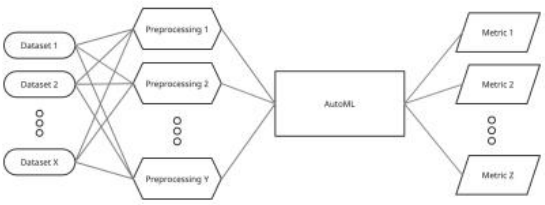

Fig. 1. **High-level architecture of the bench marking framework.**

Tomek Links [56] is a data cleaning method used to make the location near the selected area using a popular tool called Tomek Links. A pair of points xi Smin, xj Smaj belongs to the opposite training as a Tomek Link if there is no sample xk such that d(xi, xk) ≤ d(xi, xj) or d(xj , xk) ≤ d (xi, xj).

One-sided selection - OSS [37] divides most models into noise models, boundary models, recursive models and security models. OSS detects and removes irregular patterns using CNN. Next, Tomek Links removed the noise and border patterns, leaving us with the best patterns left.

Neighborhood Cleaning Rule - NCL [39] tries to solve one of the main problems of OSS methods by replacing CNN with ENN, because CNN tends to keep noisy samples in training [39]. In addition, NCL cleans the community of low standards. For each small sample, it counts its three closest neighbors. If the neighbors classify the sample of the minority, this removes the neighbors from that sample which is the best of the majority.

Cluster Centrists [49] is our closed method and is the best way to represent the time of the sample under the sample. It uses the K Means algorithm to group the most samples in a class into clusters and create the centrists of the clusters based on the most new samples.

IV EXPERIMENT SETUP

We have developed a framework for comprehensive and robust testing with many different priorities across multiple datasets based on multiple metrics. The central idea of the framework is illustrated in Figure 1. Each execution combines the data, a predefined path and the influence of its hyper parameters determined using the target search. On each run, a per-process is used to learn a portion of the data, thereby providing an updated version of the school, which is then sent to the Auto ML component of the framework. We use the modern Auto ML framework, Auto-Sklearn [23] to select, train and tune appropriate classes for the given data. We provide more information about Auto-Sklearn in Section IV-A1. Once the classifier is intelligent, we make predictions using random examples from testing and test results.

A. Define the reference

ISSN: 2366-1313

We performed a benchmark covering the 16 predefined methods mentioned in Section III and a non-initial test. We have mentioned several hyper parameter settings for each verification process in Table I. All implementations of the per-processes used in the benchmarks come from the Imbalanced Learn library [40].

All previous methods have been transformed to work on the 23 public records and members highlighted in Table II. Non-cybersquatting public information was downloaded from Open ML [57]. We carefully selected the datasets based on several criteria, including data set size, missing values, and inconsistencies. We require that each Open ML data set be binary and contain at least 5,000 samples; A maximum of 20% of the sample can have missing values, and the minimum odds ratio must be 1:10. Although we primarily focus on the binary format, data conflicts also arise in many environmental classes. However, for the sake of simplicity and compatibility with different authors and classes, we will retain the best of the binary case. Extension to multiple optimizations can be done easily by working one-on-one or one-on-one relationships for per-processes and micro and macro supports for measurement. We used seventy-five percent of the statistical samples of all data as study and the remaining 25% as test. The separation was made to preserve the true disparity between the two pieces.

We used Auto-Sklearn IV-A1 to find, train, and correct the training program's first-class results using 5-fold validation as a validation method. Auto-Sklearn switches to the optimal configuration for the ROC AUC IV-B2 parameter. Each run was converted into a full half-hour set for training public data; A single learning model has 10 minutes to complete the training. Failed executions are not repeated. Because of their size, that becomes a good five minutes for Auto-Sklearn on proprietary data, and doesn't want to repeat itself. We do not limit the duration of processioning steps in any way in order to collect information on the effectiveness of prepossessing techniques on datasets of different sizes.

1) AutoSklearn: Auto-Sklearn [23] is a library for automated model selection and hyper parameter tuning. Auto-Sklearn allows us to explore many models without presenting ourselves in a biased way in the process. We chose Auto-Sklearn for its superior overall performance compared to various competing Auto ML models [23]. Although the second version of Auto-Sklearn, bringing large-scale

ISSN: 2366-1313

advancements [22], was available in 2020, we chose to no longer use it because it became an experimental part during the simulation era .

Auto-Sklearn extends existing Auto ML architectures using a Bayesian optimizer using the meta-acquired knowledge and architecture to improve the tool's performance. In short, we provide an explanation of how each of the add-nos works and give the timely message that we need to control the behavior of Auto-Sklearn so that the code is successfully manipulated during testing.

Bayesian optimization works as an intelligent random search for hyper parameter tuning. It's a powerful process.

Method	Hyperparameter Configurations
Baseline	1
Random Oversampling	2
SMOTE	4
Borderline SMOTE	16
SVM SMOTE	8
KMeans SMOTE	4
ADASYN	4
Random Undersampling	2
CNN	2
ENN	4
Repeated ENN	4
All KNN	4
Near Miss	12
Tomek Links	1
One-Sided Selection	2
NCL	8
Cluster Centroids	4
Σ	82

TABLE I
HYPERPARAMETER CONFIGURATIONS FOR PREPROCESSING METHODS. THE TABLE SHOWS THE NUMBER OF AVAILABLE HYPERPARAMETER CONFIGURATIONS IN THE BENCHMARK.

Name	Maj. Size	Min. Size	Imbalance
Asteroid	125,975	156	807.532
Credit Card Subset [17]	14,217	23	618.130
Credit Card [17]	284,315	492	577.876
PC2 [50]	5,566	23	242.000
MC1 [50]	9,398	68	138.206
Employee Turnover	33,958	494	68.741
Satellite [25]	5,025	75	67.000
BNG - Solar Flare	648,320	15,232	42.563
Mammography	10,923	260	42.012
Letter [24]	19,187	813	23.600
Relevant Images	129,149	5,582	23.137
Click Prediction V1	1,429,610	66,781	21.407
Click Prediction V2	142,949	6,690	21.368
Amazon Employee	30,872	1,897	16.274
BNG - Sick	938,761	61,239	15.329
Sylva Prior	13,509	886	15.247
BNG - Spect	915,437	84,563	10.826
CIC-IDS-2017 [51]	227,132	5,565	40.814
UNSW-NB15 [45]	164,673	9,300	17.707
CIC-Evasive-PDF [33]	4,468	555	8.050
Ember [4]	200,000	26,666	7.500
Graph - Embedding [20]	394	154	2.558
Graph - Raw [20]	394	154	2.558

TABLE II
DATASETS. THE TABLE SHOWS BASIC INFORMATION ABOUT THE DATASETS USED IN THE BENCHMARK. THE UPPER PART OF THE TABLE SHOWS PUBLICLY AVAILABLE NON-CYBERSQUATTING DATASETS; THE LOWER PART SHOWS CYBERSECURITY DATASETS AND TWO PROPRIETARY DATASETS CONCERNING THE CLASSIFICATION OF NODES IN NETWORK GRAPHS.

Necessary for finding the extreme of expensive-to-observe target features, with tuning hyper parameters in the machine learning version, in a few sampling steps as possible [14]. Bayesian optimization fits a probabilistic model providing a relationship between hyper parameters and overall output performance. The

ISSN: 2366-1313

probabilistic version shows an expected hyper parameter configuration based on its current input. It evaluates the reference version of these hyper parameters and applies the result to put your faith back in the loop. It can explore new areas and use established areas to strengthen overall performance [23].

The meta-analysis uses the previous experience recorded in the knowledge base that includes pairs of data set features and gadget to know the model + hyper parameters that show the overall performance of this data to show the model will work well on the new data. The features are considered as vectors in the environment meta-feature vector that allows us to use the remote sensing of the data to find the data comparison and use the models that are good at data set as a starting point for similar. Auto-Sklearn can quickly recommend settings for the Bayesian optimizer that will work well on new data. Unfortunately, we encountered many errors during the meta-learning process and failed to make it work properly. Therefore, we have completely failed our test.

The data prepossessing stage in Auto-Sklearn includes zero-value imputation, one-bit encoding, domain normalization, scaling, and centering. Feature prepossessing attempts to create new features using polynomial features or select a set of features using PCA or ICA. Auto-Sklearn also sometimes choose to be equal. The weighting index is used to penalize classifications of classes more than classifications of others. We have missed all three steps to maintain full maintenance of the test. We practice equal prioritization procedures for all data and do not perform prioritization.

Auto-Sklearn uses a list of 15 algorithms in its search. The list can be found on their GitHub. We did not include the Multi-Layer Perceptron from the list because it uses a lot of resources during the training and the data set used in the test does not require the use of neural networks [52].

B. Performance measurement

We now provide a list of performance measures that we used as part of the evaluation. We did not use measures including accuracy and balance, because they are not appropriate measures of overall performance in non-balance problems [12]. In addition to the measures shown below, we also measured the F-level and Matthews Correlation Coefficients (MCC) [44]. We do not include them in the result because of local influence and because F-score and MCC

ISSN: 2366-1313

examine the best performance of the work alone and not the measures listed below no.

1) Area under the PR curve (PR AUC): The precision-recall curve plots the precision and recall values of the successful selection. Remember = T P is drawn horizontally

Area under the ROC curve (ROC AUC): The receiver operating characteristic curve represents the false positive rate, FPR =

F P, on the horizontal line in opposition to the Positive Value,

TPR = T P, on the vertical line, the calculation of the possible option. Again, an overall performance given the performance of a distribution can be obtained by calculating the neighborhood under the curve. When FR AND TPR intersection, the perfect performance of the classification of the point (zero, 1). ROC curve is not rooted in inequality of elegance. However, in the presence of high uncertainty, even a high ROC AUC operator may not be useful (12). The following diploma addresses this issue.

	PR AUC	ROC AUC	P-ROC AUC
Baseline	6.196 // 23	6.761 // 23	10.891 // 23
Random Oversampling	9.238 // 21	9.024 // 21	6.929 // 21
SMOTE	6.283 // 23	6.174 // 23	4.087 // 23
Borderline SMOTE	5.935 // 23	6.239 // 23	4.500 // 23
SVM SMOTE	4.841 // 22	4.909 // 22	3.545 // 22
KMeans SMOTE	4.500 // 04	2.625 // 04	4.000 // 04
ADASYN	7.955 // 22	7.977 // 22	5.386 // 22
Random Under sampling	10.318 // 22	9.818 // 22	5.773 // 22
CNN	11.964 // 14	11.821 // 14	9.036 // 14
ENN	6.310 // 21	6.452 // 21	9.524 // 21
Repeated ENN	7.222 // 18	6.583 // 18	11.056 // 18
All KNN	8.273 // 22	8.273 // 22	11.159 // 22
Near Miss	11.023 // 22	11.341 // 22	7.068 // 22
Tomek Links	7.667 // 21	7.857 // 21	11.810 // 21
One-Sided Selection	8.455 // 22	9.091 // 22	12.000 // 22
NCL	9.023 // 22	8.886 // 22	11.068 // 22
Cluster Centroids	10.706 // 17	10.235 // 17	6.529 // 17

TABLE III
MEAN RANK ACROSS ALL DATASETS. THE TABLE CONTAINS AVERAGE RANKS FOR EACH COMBINATION OF PREPROCESSING METHOD AND EVALUATION METRIC COMPUTED ACROSS ALL DATASETS. THE SECOND NUMBER AFTER // INDICATES THE NUMBER OF DATASETS USED TO COMPUTE THE AVERAGE.

V RESULTS

Figures 2, three and four show the distribution of count levels for each pr-processing across all index data. The ranking was calculated by taking measurements for each procedure one after the other. The dark symbols indicate the minimum, maximum and grade level of each method, and the 3 blue symbols mean the 25th, 50th and 75th percentiles of each standard. The average level should also be indicated in Table III. Table IV shows the average levels calculated only for the cybersecurity datasets. We do not currently publish classifications for cybersecurity datasets due to space constraints and small sample sizes. Table V zooms in on the SMOTE code and its variants and compares the relative differences between the methods. Due to the small number of performance factors, we ignore K Means

ISSN: 2366-1313

SMOTE in Table V. Finally, Figure five shows the total running time of each method.

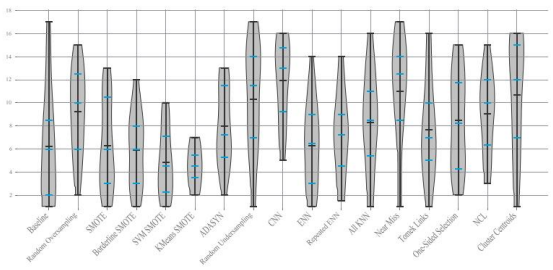

Fig. 2. **Area under PR Curve (PR AUC).** Ranks for each method were measured across all datasets in the benchmark.

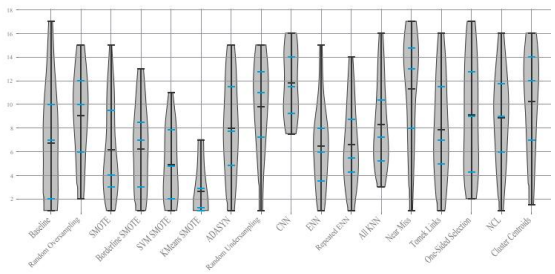

Fig. 3. **Area under ROC Curve (ROC AUC).** Ranks for each method were measured across all datasets in the benchmark.

VI CONCLUSION

We have conducted a research on 16 ideas before 23 documents, six of which come from the cybersecurity domain. We have learned all the forecasting and calculation operations. To this end, we performed a large-scale experiment using Auto ML to determine various distributions and included hyper parameter search to eliminate potential biases present in higher standards.

Our main finding is that using data set prepossessing to deal with a group of unequal performance is often beneficial. However, at the same time, many ideas do not follow the solution of doing nothing. In general, oversampling techniques perform better than sampling, but there are exceptions. Among the up sampling methods, the traditional SMOTE algorithm achieves the best overall performance, while its maximum change will lead to the improvement of the simple incremental nature.

When we limit our analysis to cybersecurity datasets that span multiple cybersecurity domains, we come to the same conclusion as above.

Finally, it is important to remember that the evaluation method is applied by measuring performance. We include some measures of effectiveness that can be achieved and appropriate in appropriate situations when dealing with class inequality. Although the specifics of the tests vary by grade, the main points mentioned above are constant.

REFERENCES

1. Mostofa Ahsan, Rahul Gomes, and Anne Denton. Smote implementation on phishing data to enhance cybersecurity. In 2018 IEEE International Conference on Elector/Information Technology (EIT), pages 0531–0536. IEEE, 2018.

2. Bathini Sai Akash, Pavan Kumar Reddy Yannam, Bokkasam Venkata Sai Ruthvik,

ISSN: 2366-1313

Lov Kumar, Lalita Bhanu Murthy, and Aneesh Krishna. Predicting cuber-attacks on iot networks using deep-learning and different variants of smote. In International Conference on Advanced Information Networking and Applications, pages 243–255. Springer, 2022.

3. Adnan Amin, Sajid Anwar, Awais Adnan, Muhammad Nawaz, Newton Howard, Junaid Qadir, Ahmad Hawalah, and Amir Hussain. Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. IEEE Access, 4:7940–7957, 2016.

4. Hyrum S Anderson and Phil Roth. Ember: an open data set for training static pe malware machine learning models. Ar Xiv preprint arXiv:1804.04637, 2018.

5. Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. Dos and don'ts of machine learning in computer security. In 31st USENIX Security Symposium (USENIX Security 22), pages 3971–3988, Boston, MA, August 2022. USENIX Association.

6. Rob Ashmore, Radu Calinescu, and Colin Paterson. Assuring the machine learning life cycle: Desiderata, methods, and challenges. 54(5), May 2021.

7. Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions on Information and System Security (TISSEC), 3(3):186–205, 2000.

8. Salahuddin Azad, Syeda Salma Naqvi, Fariza Sabrina, Shaleeza Sohail, and Sweta Thakur. Iot cybersecurity: On the use of machine learning approaches for unbalanced datasets. In 2021 IEEE Asia-Pacific Confer- ence on Computer Science and Data Engineering (CSDE), pages 1–6. IEEE, 2021.

9. Sikha Bagui and Kunqi Li. Resampling imbalanced data for network intrusion detection datasets. Journal of Big Data, 8(1):1–41, 2021.

10. Ricardo Barandela, Rosa M Valdovinos, J Salvador Sa´nchez, and Francesc J Ferri. The imbalanced training sample problem: Under or over sampling? In Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR), pages 806–814. Springer, 2004.

11. Jan Brabec, Toma´sˇ Koma´rek, Vojteˇch Franc, and Luka´sˇ Machlica.

ISSN: 2366-1313

On model evaluation under non-constant class imbalance. In International Conference on Computational Science, pages 74–87. Springer, 2020.

12. Jan Brabec and Lukas Machlica. Bad practices in evaluation methodology relevant to class-imbalanced problems. arXiv preprint arXiv:1812.01388, 2018.

13. Prasadu Peddi, and Dr. Akash Saxena. "studying data mining tools and techniques for predicting student performance" International Journal Of Advance Research And Innovative Ideas In Education Volume 2 Issue 2 2016 Page 1959-1967