Information Gain applied to reduce model-building time in decision-tree-based intrusion detection system

Moisés S. de Sousa[1], Carlos Eduardo Lacerda Veiga[2], Robson de O. Albuquerque[1], William F. Giozza[1]
[1] Professional Post-Graduation Program in Electrical Engineering – PPEE – Electrical Engineering Department, Faculty of Technology, University of Brasília (UnB), Brasília, Brazil, Zip Code 70910-900
[2] Advocacia-Geral da União, SIG-Quadra 06-Lote 800, Brasília, Brazil, Zip Code 70610-460
moisesssousa98@gmail.com, dr.cadu@gmail.com, robson@redes.unb.br, giozza@unb.br

Abstract —**Due to the large amount of sensitive data generated by websites, it is possible to understand the progress of attacks to their databases. This work proposes an intrusion detection system based on data mining and machine learning techniques to detect and mitigate the damage caused by these attacks. It adopts the Information Gain method of selecting attributes in order to reduce the model-building time without affecting the classification performance. Using the CIC-IDS 2017 dataset, this work shows how different decision tree algorithms (Random Forest and J48 Algorithm) behave even if they receive equal parameters and data. Using Information Gain to select attributes, the proposed system achieves a processing time reduction of up to 90%.**

Keywords – Cybersecurity, web attacks, data mining, artificial intelligence, decision tree, J48 Algorithm, Random Forest.

I. INTRODUCTION

Currently, many services are maintained by the Internet. Common in much of the world, these websites, and applications aggregate a large amount of information and there is a need to protect this sensitive data. This process begins with the detection of possible attacks [1]. In this respect, the use of data mining techniques helps in identifying them [2].

This is possible because data mining finds existing patterns in a large volume of data. Thereby, the system can inform if what has been inserted into the database, for example, is expected [3]. Therefore, the emergence of Invasion Detector System (IDS) gain notoriety.

IDS is based on the flow of packets that travel through the network, allowing to distinguish whether the inputs correspond to a normal and expected traffic or an unwanted intrusion.

The IDS implementation is possible through supervised machine learning techniques. In this process, a dataset is loaded with the network flow information [4]. Each flow is classified according to what the IDS is proposed to detect, i.e., normal traffic or intrusion.

This work uses the dataset CIC-IDS 2017 built by the Canadian Cybersecurity Institute [5]. It will be best detailed in the third section of this work. The decision tree machine learning algorithms used are J48 and Random Forest [6]. Both algorithms have the advantages of an easy interpretation of the results and preparation of the data.

In addition, this work is concerned with reducing the time taken to build the model. For this, the Information Gain technique was used for the selection of features, maintaining the ability of the model to correctly classify the classes.

The rest of this paper is structured as follows. In Section II, are presented the most relevant related works regarding the use of data mining techniques and machine learning tools for intrusion detection. Section III presents the modeling of the proposed framework, describing its functionalities and components as dataset, selected features, classifier algorithm, training, and testing. Implementation and the steps used for training the agents are described in Section IV. Section V presents and discusses the results. Finally, the conclusions are presented in Section VI.

II. RELATED WORKS

In the literature, there is a large collection of research that addresses the use of machine learning algorithms to detect intrusion in virtual environments in general.

One of the examples can be seen in the work of Kurniabud et al. [7] where they compared the results obtained with various types of classifiers available. The comparative results end up indicating which classifier would be the best. This research uses Information Gain, as an attribute selection filter, which is a technique of noise reduction in datasets [7]. It creates a ranking of attributes, which are determined by calculating entropy.

On the other hand, Abbas [8] performs Dimensionality Reduction using a Principal Component Analysis (PCA) algorithm to measure the results. Similar to the work of Kurniabud et al., the methodology adopted is to compare results with different classification techniques. Dimensionality reduction, using PCA Algorithm, reduces the dimensions of the dataset without loss of data features.

A. A. Tawil and K. E. Sabri [4] selects the characteristics of interest through Moth Flame Optimization (MFO). Their study also compared with another method of selecting attributes, the Correction Feature Selection (CFS), as well as sending the data without any treatment for training. As proven, part of the need to assign these filters is due to the shorter time spent in the training and classification process.

Ali et al. [6] use two filters for selection of attributes: the CFS and the Classifier Subset Eval. Using two different classifiers – K-Nearest neighbor (IBK) and Multi-LayerPerception (MLP) – this work compares how different selections impact the classification of the results. There are no

2022 17th Iberian Conference on Information Systems and Technologies (CISTI)
22 – 25 June 2022, Madrid, Spain
ISBN: 978-989-33-3436-2

mentions of the construction time of the models, but the results above 99% indicate the good behavior of the use of these classifiers with the selected characteristics.

In Ahmed and Varol [1], five filter technologies were applied to attribute selection. The training procedure uses 19 of the 78 characteristics available. Tests are performed using different machine learning algorithms, including the algorithm PART that combines the algorithm C4.5 (which gives rise to the J48 used in this work) and the algorithm RIPPIER. Besides, Random Forest, Naive Bayes and BayesNet algorithms are also used for comparative purposes.

Shaukat et al. [9], only one attribute selector (Subset Evaluator) was used, resulting in just 8 characteristics taken into consideration in the training and testing procedures. The algorithms used were the Naive Bayes and the J48. Despite the shorter time to build a model, the J48 achieved the best ratings.

Therefore, this work uses the CIC-IDS 2017 dataset. Information Gain is used to select the most relevant features for the training and classification process to reduce the time required for the construction of the models, compared to those who use all of them. In turn, these classifications are made through decision tree algorithms: J48 and Random Forest. The software chosen for the experiment is Weka.

III. METHODOLOGY

This section describes the modeling of the proposed framework for intrusion detection based on data mining and machine learning techniques, detailing their functionalities and main components such as dataset, features selection, classifier algorithm, training, and testing.

A. Experimental setup

The experimental setup is shown in Figure 1.

The first step of the experiment is the correct choice of dataset. The dataset CIC-IDS 2017 is segmented in days, avoiding the need for file partitioning, as is common in works involving machine learning. In this case, only Thursday was chosen.

After choosing the dataset, it is necessary to convert data format from Comma-separated Values (CSV) to ATTRIBUTE-Relation File Format (ARFF) which is the only extension accepted in the software used for the experiments (Weka). Fortunately, Weka does this kind of conversion, avoiding the need to build a second program.

The third step of the experiment is to rank the features with the correct extension for reading the data in Weka. Because there are many features (78), using all of them in the training process will cost time and computational resources. To reduce the number of selected features, the Information Gain was chosen. From the preliminary results, the 10 most relevant characteristics were selected.

The next step was also performed with all the features of the dataset to prove that reduces the time of the experiment.

The fifth step is the training and testing of the data, with the most relevant features selected. The dataset split is 80% for training and 20% for testing. As this is a comparative study, the fourth step was performed twice, once for each type of algorithm used in the experiment (Random Forest and J48). It is important to emphasize that the classes are supervised and all of them were passed by the dataset.

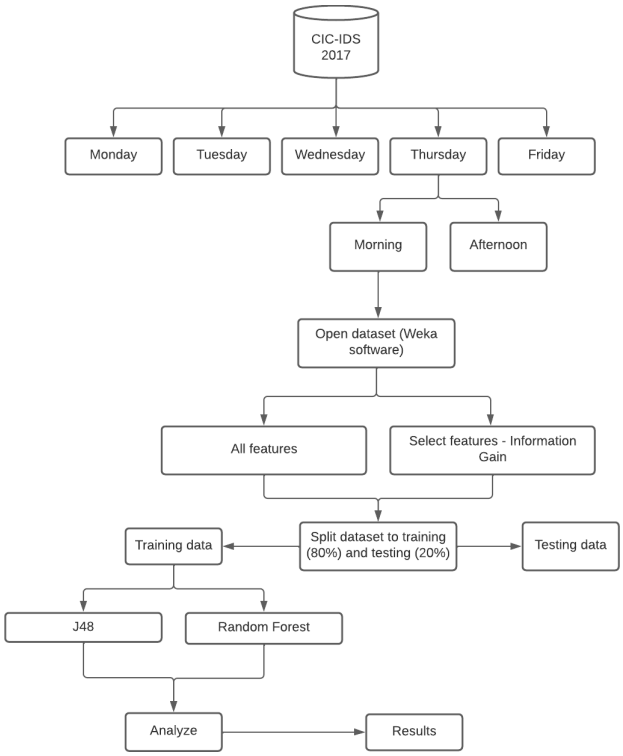

Fig. 1. Experimental setup.

The last step of the experiment is the analysis of the results obtained from training and classification. It involves not only the absolute numbers, but also the comparison between them.

B. Dataset

The dataset chosen for validation of the results is CIC-IDS 2017 [5] assembled by the Canadian Institute for Cybersecurity of the University of New Brunswick [5].

The reason for choosing this dataset is that it is intended for intrusion detection. CIC-IDS 2017 has criteria considered necessary for the construction of datasets focused on IDS [9]. They are: complete network configuration, complete traffic, attack diversity, labelled dataset, complete interaction, complete capture, available protocols, heterogeneity, feature set and metadata. In addition, the organization makes it easy to prepare data for training. In this case, only the selection of features was required.

The organization of this dataset is as follows. The network flow of a given topology is captured five days of the week. For each period, there is a CSV file with all the data collected. For the experiments of this work, the day selected was Thursday (during the morning), since this day collected data regarding web attacks as XSS (Cross-site Scripting), Brute Force and SQL Injection. In all, there are 78 characteristics for each of the 170,366 existing instances in this dataset.

C. Information Gain

As stated in the previous section, there are 78 features in the chosen dataset. This number is high, and not all features are relevant in the training procedure.

Therefore, selecting the best attributes for training is essential for the development of the experiment.

This technique uses entropy values to achieve desirable results to reduce the existing noise in the dataset, selecting the most relevant features [7].

D. J48 Algorithm

The J48 algorithm is an open-source decision tree algorithm implemented by the Weka software [10]. The entropy has a fundamental role in this algorithm because the root of this tree is the characteristic with the highest entropy, among existing ones [6].

E. Random Forest

While the J48 algorithm uses the entropy value for assembling the decision tree, Random Forest creates a set of different trees and performs the combination of them to make more stable and accurate decisions [11].

F. Weka software

Weka is an open-source software built in Java and used for tasks involving data mining and artificial intelligence. Because of this, it contains tools for data preparation, classification, visualization, and more [12].

IV. IMPLEMENTATION

The dataset built by the Canadian Cybersecurity Institute, CIC-IDS 2017, is presented in two different formats: Packet Capture (PCAP) and CSV. Regardless of the extension chosen, both are subdivided according to which days of the week that attacks occurred and which days there was a normal use of the environment.

The extension chosen was the CSV, due to the reduced size and the possibility of converting the file into ARFF directly by the Weka software.

In addition, the purpose of this work is not to investigate all the attacks made available by the dataset. Therefore, only the Thursday morning period was selected, because in this range there is the attack of interest: Brute Force. The classes present in this dataset are listed in Table I.

The experiment proceeds with the conversion of the data to the format that Weka supports. In this case, the ARFF. The dataset was converted using the ARFF Viewer and then uploaded to Weka. Before training and testing, there is a need to filter out some features in this dataset. Therefore, with the use of Information Gain, there was the creation of a ranking with the most relevant features for the experiment.

The less important features are removed of the experiment, remaining only 10 features as listed in Table II. It is possible to notice that many of the remaining features indicate some type of flow, proving that the efficiency in terms of response time is essential for the adoption of machine learning techniques. However, in a second moment, the training is done with all the features. Consequently, it is possible to assess whether the adoption of Information Gain is beneficial.

TABLE I – DATASETINFORMATION

Web Attack	Number of instances
BENIGN	168.186
Brute Force	1.507
XSS	652
SQL Injection	21

TABLE II – INFORMATION GAIN RANKING FILTER

#No	Feature name
1	Flow Packets/s
2	Flow IAT (Inter Arrival Time) Mean
3	Flow IAT (Inter Arrival Time) Std
4	Flow IAT (Inter Arrival Time) Max
5	Flow IAT (Inter Arrival Time) Mean
6	Flow IAT (Inter Arrival Time) Min
7	Flow Header Length
8	Fwd Packets/s
9	Fwd Header Length1
10	Init Win bytes backward

With the filtered dataset, one can perform the training and validations of this process, using 80% of the entire data set to perform the training and 20% for the testing.

The training and classification procedure is performed in two occasions, because of the use of two distinct classifiers: Random Forest and J48 Algorithm.

From the completion of this step, the software itself compiles the results of True Positives (TP), False Positives (FP), Recall, Precision, F-Score. These values are statistical in nature and are necessary for the preparation of the confusion matrix.

The confusion matrix evaluates the performance of the ratings [9]. The output must have at least two distinct values (which in this case would be the classes to be identified).

This experiment was executed on a computer with the following specifications: Intel Core i5 processor with 3.4 GHz, 20 GB RAM, Windows 10 as Operating System. The Weka 3.8 with heap size of 8086 MB, was used as machine learning software.

V. RESULTS

This section presents the results obtained based on the implementation described above.

The first results exposed are TP and FP. They indicate whether the software correctly indicated the situations in which they were classified. The Figures 2 and 3 show, respectively, all the correct classifications for the J48 and Random Forest algorithms. It is noticed that the normal traffic of the network,

represented by the BENIGN class, had a high rate of correct answers. The identifications of the Brute Force web attack, however, were slightly smaller, but still high, from 98%.

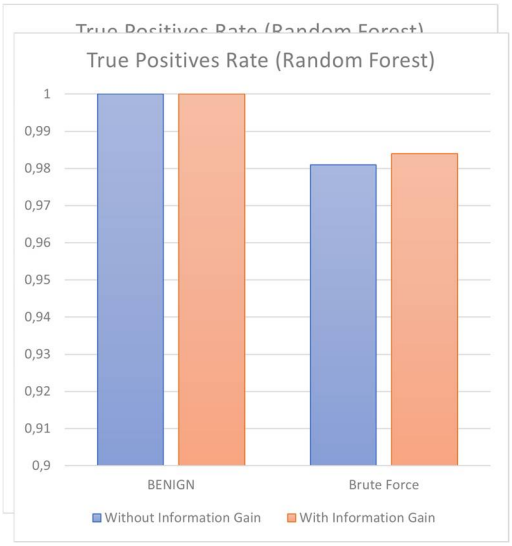

Fig. 2. True positives for J48 Algorithm.

Fig. 3. True positives for Random Forest.

The Figure 4 shows the results of False Positive. The FP occurs when the model identifies an instance of a class that does not belong to it. The closer to 0 this value, the better. The Brute Force class was zero for both decision tree algorithms. From the values of True Positive and False Positive (shown earlier in Figs. 2-4), one can measure the Precision of the design. This means that of all the times the classifier interpreted the data with the respective class, how many of them were true.

In addition to the metric Precision, there is the metric Recall. The difference in Recall is that False Negative (FN) is considered in calculations. This indicates how many are correct among the expected values for a given class. Both Precision and Recall metrics are shown in the Tables III.

Alone, both Precision and Recall do not provide enough data to draw a conclusion. F-Measure then appears to harmonize Precision and Recall values. The higher its value, the more relevant is the precision obtained in the experiment. That is, FP, FN, TP, and True Negative (TN) are not so different from each other, evidencing the balance in the results. The Table III show the F-Measure results for the J48 and Random Forest objects, respectively.

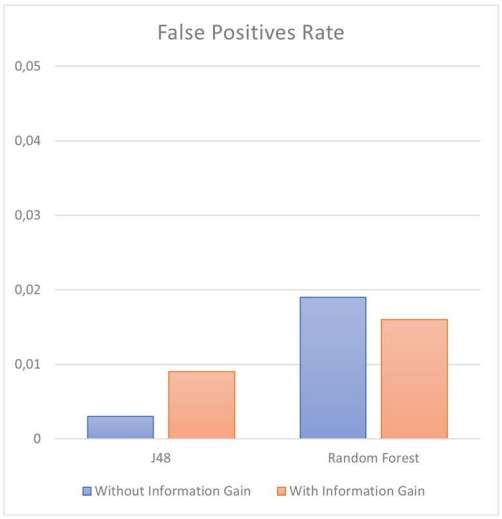

Fig. 4. False Positives Rate.

When looking at the Figure 5, it is possible to notice that there was a significant drop in the time to build the decision tree in both algorithms. This is explained by how each of them assembles these trees. With less information to process, entropy calculations (J48 Algorithm) or combinations between different trees (Random Forest) tend to decrease the time necessary for testing.

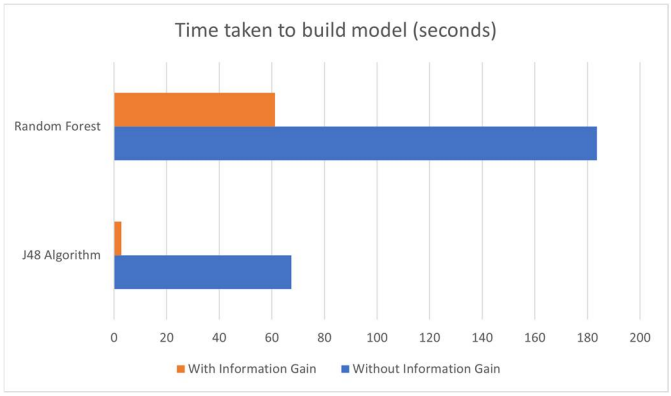

Fig. 5. Time taken to test model on test split.

Finally, the confusion matrices are shown in the Figure 6. It is possible to notice that the confusion matrix behaved as expected. This is due to the high presence of correct answers in the columns that indicate each of the classes.

Despite the time taken to build the model being considerably less, when observing Tables IV and V it is possible to notice that the classifications were not impaired, maintaining the good level of hit in both processes.

TABLE III – RESULTSREGARDING J48 ALGORITHM AND RANDOM FOREST

	J48 Algorithm				Random Forest			
	Precision	Recall	F-Measure	Class	Precision	Recall	F-Measure	Class
Without select features	1	1	1	BENIGN	1	1	1	BENIGN
With select features	1	1	1		1	1	1	
Without select features	0.997	0.997	0.997	Brute Force	1	0.981	0.991	Brute Force

With select features	0.997	0.991	0.994		1	0.984	0.992	

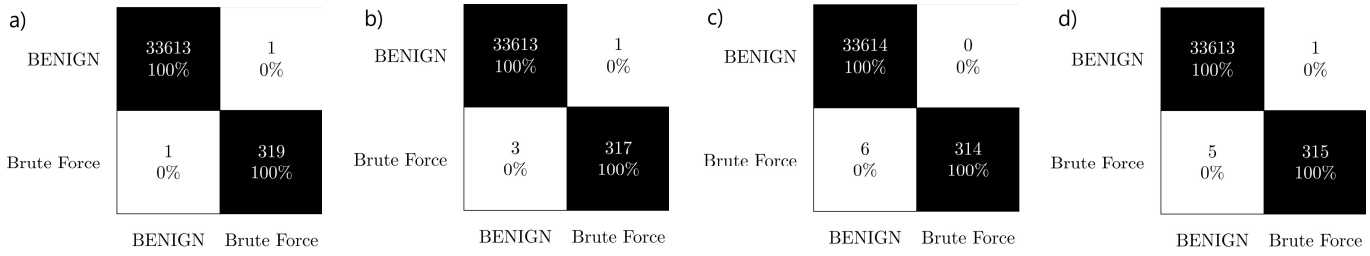

Fig. 6. a) Confusion matrix for J48 Algorithm without Information Gain. b) Confusion matrix for J48 Algorithm with Information Gain. c) Confusion matrix for Random Forest with Information Gain. d) Confusion matrix for Random Forest without Information Gain.

Despite the time taken to build the model being considerably less, when observing Tables IV and V it is possible to notice that the classifications were not impaired, maintaining the good level of hit in both processes.

TABLE IV – J48ALGORITHM RESULTS

	Correctly classified	**Incorrectly classified**
Without select features	99.9941 %	0.0059%
With select features	99.9882 %	0.0118%

TABLE V – RANDOMFOREST RESULTS

	Correctly classified	**Incorrectly classified**
Without select features	99.9823 %	0.0177%
With select features	99.9853 %	0.0147%

A. Comparison with related works

As shown in the Table VI, it is noted that the values for Recall, Precision and F-Measure were slightly better in the present work when compared to Shaukat, et al. [9] when the J48 Algorithm is used.

TABLE VI – RECALLPRECISION AND F-MEASURE COMPARISON

	Precision	**Recall**	**F-Measure**
S. Shaukat, et al. [9]	0.999	0.998	0.999
This work	1	1	1

When comparing the above results with the values obtained by Ahmed and Varol [1] (Table VII), for example, it is noted that there were improvements in building time when using the Random Forest algorithm. The decrease from 99.31s to 61.22s is significant and indicates that the selection of features through the Information Gain technique was more efficient, since the results of the classifications remained at similar levels.

TABLE VII – F-MEASUREAND BUILDING TIME COMPARISON

	F-Measure	**Building time (seconds)**
Ahmed and Varol [1]	0.995	99.31
This work	0.992	61.22

This is explained because Information Gain reduces the entropy of the dataset. This means that the randomness in the decision is decreased, facilitating the process of choosing the correct class by the agent.

VI. CONCLUSION

From the results presented, it is possible to assess that the adoption of Information Gain is important for reducing the time to build decision trees.

In addition, the classifications made by both algorithms remained at considerably similar levels. That is, although there was no improvement in the results, especially of the J48, there was no prejudice in the decisions made with all the features. Therefore, besides choosing a good dataset, preparing well its data also becomes an indispensable step in the machine learning process, as seen in the results obtained in this work. It is important to emphasize that the decrease in building time and maintenance of classification results with the use of Information Gain is conditioned in the terms of this work, not allowing generalization. For this purpose, new experiments must be carried out.

Another important detail is the easy way that Weka software assembles the decision tree. Due to this easy-to-use feature, the time it takes for the developer to create templates is reduced, since the dependency on code creation is virtually nonexistent.

ACKNOWLEDGMENT

R.d.O.A. e W.F.G gratefully acknowledge the General Attorney of the Union—AGU grant 697.935/2019 and the

General Attorney's Office for the National Treasure—PGFN grant 23106.148934/2019-67. R.d.O.A. thanks Nubank for its support and acknowledges this work was partially supported by EC Horizon 2020 HEROES project grant 101021801. W.F.G. acknowledges this work was partially supported by FAP-DF—Brazilian Federal District Research Support Foundation, under Grant 00193-00000229/2021-21 and by the Brazilian Ministry of the Economy under Grant 005/2016 and Grant 083/2016.

REFERENCES

[1] O. I. Ahmed and C. Varol, "Detection of web attacks via part classifier," in 2021 9th International Symposium on Digital Forensics and Security (ISDFS), 2021, pp. 1–4. DOI: 10.1109/ISDFS52919.2021.9486329.

[2] M. Kantarcioglu and B. Xi, "Adversarial data mining: Big data meets cybersecurity," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '16, Vienna, Austria: Association for Computing Machinery, 2016, pp. 1866–1867, ISBN: 9781450341394. DOI: 10.1145/ 2976749.2976753.

[3] V. Desai, K. Oza, and P. Naik, "Data mining approach for cybersecurity," International Journal of Computer Applications Technology and Research, vol. 10, pp. 035–041, Jan. 2021. DOI: 10. 7753 / IJCATR1001. 1007.

[4] A. A. Tawil and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on moth flame optimization," in 2021 International Conference on Information Technology (ICIT), 2021, pp. 377–381. DOI: 10.1109/ICIT52682.2021.9491690.

[5] S. H. Iman Sharafaldin Arash Habibi Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018.

[6] A. Ali, S. Shaukat, M. Tayyab, et al., "Network intrusion detection leveraging machine learning and feature selection," in 2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), 2020, pp. 49–53. DOI: 10. 1109/HONET50430.2020.9322813.

[7] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Idris, A. M. Bamhdi, and R. Budiarto, "Cicids-2017 dataset feature analysis with information gain for anomaly detection," IEEE Access, vol. 8, pp. 132 911–132 921, 2020. DOI: 10.1109/ACCESS.2020.3009843.

[8] M. S. A. Sara Abdalelah Abbas, "Distributed denial of service attacks detection system by machine learning based on dimensionality reduction," Journal of Physics: Conference Series, 2020.

[9] S. Shaukat, A. Ali, A. Batool, et al., "Intrusion detection and attack classification leveraging machine learning technique," in 2020 14th International Conference on Innovations in Information Technology (IIT), 2020, pp. 198–202. DOI: 10.1109/IIT50501.2020.9299093.

[10] E. M. d. Araujo Vieira, N. Neves, A. C. de Oliveira, R. de Moraes, and J. do Nascimento, "Avaliação da performance do algoritmo j48 para construção de mode- los baseados em árvores de decisão," Revista Brasileira de Computação Aplicada, vol. 10, no. 2, pp. 80–90, Jul. 2018. DOI: 10. 5335 / rbca. v10i2. 8078. [Online]. Available: http://seer.upf.br/index.php/rbca/article/view/ 8078.

[11] K. Alpan, "Performance evaluation of classification algorithms for early detection of behavior determinant based cervical cancer," in 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2021, pp. 706–710. DOI: 10. 1109/ISMSIT52890.2021.9604718.

[12] M. A. H. Eibe Frank and I. H. Witten, "Data mining: Practical machine learning tools and techniques," Morgan Kaufmann, 2016.