Enhancing Network Security Through Granular Computing: A Clustering-by-Time Approach to NetFlow Traffic Analysis

Mikołaj Komisarek
ITTI Sp. z o.o.
Poznań, POLAND

Marek Pawlicki
Bydgoszcz University of Science and Technology
Bydgoszcz, POLAND
ITTI Sp. z o.o.
Poznań, POLAND

Salvatore D'Antonio
Naples University Parthenope
Naples, Italy

Rafał Kozik
Bydgoszcz University of Science and Technology
POLAND
ITTI Sp. z o.o.
Poznań, POLAND

Aleksandra Pawlicka
University of Warsaw
Warsaw, POLAND
ITTI Sp. z o.o.
Poznań, POLAND

Michał Choraś
Bydgoszcz University of Science and Technology
Bydgoszcz, POLAND
ITTI Sp. z o.o.
Poznań, POLAND

ABSTRACT

This paper presents a study of the effect of the size of the time window from which network features are derived on the predictive ability of a Random Forest classifier implemented as a network intrusion detection component. The network data is processed using granular computing principles, gradually increasing the time windows to allow the detection algorithm to find patterns in the data at different levels of granularity. Experiments were conducted iteratively with time windows ranging in size from 2 to 1024 seconds. Each iteration involved time-based clustering of the data, followed by splitting into training and test sets at a ratio of 67% - 33%. The Random Forest algorithm was applied as part of a 10-fold cross-validation. Assessments included standard detection metrics: accuracy, precision, F1 score, BCC, MCC and recall. The results show a statistically significant improvement in the detection of cyber attacks in network traffic with a larger time window size (p-value 0.001953125). These results highlight the effectiveness of using longer time intervals in network data analysis, resulting in increased anomaly detection.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Security and privacy** → **Intrusion detection systems**.

KEYWORDS

feature engineering, granular computing, NetFlow, network intrusion detection

ARES 2024, July 30–August 02, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3670882

ACM Reference Format:
Mikołaj Komisarek, Marek Pawlicki, Salvatore D'Antonio, Rafał Kozik, Aleksandra Pawlicka, and Michał Choraś. 2024. Enhancing Network Security Through Granular Computing: A Clustering-by-Time Approach to NetFlow Traffic Analysis. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3664476.3670882

1 INTRODUCTION

In the present day, infrastructure must be protected to keep sensitive information and critical assets confidential [11]. With multiple reasons why malicious users attack infrastructure [14], as the number of network attacks grows and the attacks become more sophisticated, new techniques are needed that can dynamically respond and strengthen defences. This article examines the application of granular computing to network security, specifically exploring the use of granular approaches in analyzing NetFlow traffic. NetFlow, a protocol that is widely used as a means of monitoring the behaviour of network traffic, contains data that, when grouped, can be used to detect network attacks. This research studies the impact of different levels of granulation, according to the principle of grouping by time, to reveal its influence on detection metrics obtained by ML-based network intrusion detectors. The premise of using different levels of granularity to look for suspicious network traffic patterns comes from the fact that different granularity levels can emphasize different aspects of network behaviour. In the context of network security, granular computing can enable more accurate detection of anomalies that may indicate potential threats. Temporal clustering adds an additional level of complexity, taking into account temporal relationships between network traffic patterns and enabling a different approach to attack detection. The main goal of this research is to contribute to improving network security by evaluating the performance of granular computing applied to NetFlow traffic. The following research paper presents an examination of granularity for different types of network attacks using machine learning. The results of this study can serve as a guideline for creating and implementing new network security systems in

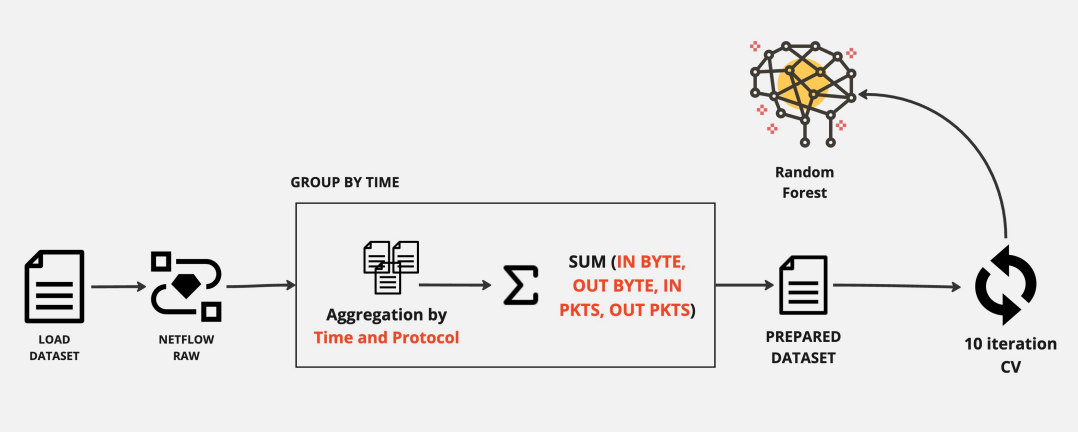


Figure 1: The concept of the proposed solution

the future, providing a basis for more effective countermeasures against the ever-changing threat landscape.

Granular computing [4], a method for processing information at multiple levels of abstraction, offers a promising option in this area of network intrusion detection because it allows for flexible and adaptive processing of data.

This paper evaluates the possibility that by using granular computing, the level of effectiveness in detecting anomalous traffic patterns that may indicate potential security risks can be increased. Validation of this work will support the development of anomaly detection methods, expanding the detection of network attacks and providing more in-depth insights into the dynamics of network data.

Thus, the major contributions of this paper are as follows

- A novel use case of granular computing to real-time network intrusion detection systems.
- Evaluation of the impact of temporal granularity on enhancing predictive capabilities of detection systems.
- Providing a set of experiments with time windows ranging from 2 to 1024 seconds, with empirical evidence on optimal granularity.
- Statistically significant results demonstrating that larger time windows improve the accuracy of machine learning models in detecting network anomalies.

The paper is structured as follows: The text opens with an introduction then the following section provides an overview of existing solutions related to network intrusion, anomaly detection and data granularity. The third section describes the dataset used and the Netflow protocol, as well as the methodology employed for granulating the traffic. Section four introduces the description of the individual experiments. The article concludes with the results and a summary of the research.

2 RELATED WORKS

Network security and intrusion detection have gained considerable momentum in recent years, with most research efforts focusing on designing effective strategies to prevent emerging cyberthreats. While there are multiple studies in the recent literature on utilising the techniques of machine learning in intrusion detection and traffic classification[1][17][16], it should be noted that there has not been a significant focus on the employment of granular computing in that domain, with a limited number of works. The following are some notable research works in the field of detecting suspicious network activity with a mix of ML techniques and granular computing.

In Zhang et al. (2020) [20], a real-time network attack detection system based on the combination of deep learning and flow estimation has been proposed. This method includes two algorithms: One is for real-time detection using flow feature calculations and common patterns, while the second uses a support vector machine and deep belief network (DBN-SVM). This strategy (the DBN-SVM combination) achieved higher results than the traditional machine learning methods; however, the experiment has been performed concerning detection accuracy. The enhancement of attack categorization accuracy shows that this technology can be used for real-time network intrusion detection.

Lin et al. [12] proposed Granular AutoEncoders (GAE) in their 2023 work as a solution to the overfitting and vanishing gradients problems in traditional autoencoders used for intrusion detection. Granular vectors are generated through a dedicated granulation method that concentrates on the similarity of separate traits by GAE. This way the anomaly detection rates increase greatly in comparison to the original granular vectors with their reconstructed counterparts. When tested GAE outperformed other algorithms at classifying intrusion detection datasets into anomalies.

Pawlicki et al. (2018, [15]) studied the application of Granular Computing (GrC) to network traffic classification and intrusion detection. GrC concerns turning data collections into granules that are, compact and have meaning. This paper presented a starting point for GrC and surveyed recent implementations, assessing their applicability for addressing real-world issues. The study showed the possibility of using GrC for improving efficiency, clarify data and the speed-up of classification algorithms. Through improved data abstraction and computational cost reduction, GrC shows potential as an enabler for enhanced network intrusion cybersecurity measures.

The authors of [8] (2018) introduce a novel approach to the field of network security, extending its depth of knowledge by proposing a new method to improve the detection of Denial-of-Service (DoS) attacks. It uses entropy computation to discover relevant features and granular computation to select features accurately. This helps to efficiently deal with complex and intelligent DoS attacks in a manner similar to intrusion detection systems by optimising feature selection. By integrating entropy-based metrics with granular computing, the proposed method improves the accuracy and adaptability of DoS attack detection, indicating the prospect of developing more efficient network security systems.

The paper "Machine Learning-based Intrusion Detection System for IoBT" [2] examines applications of Internet of Battlefield Things (IoBT), a military version of the Internet of Things (IoT), to improve the efficiency of military operations. IoBT connects various military assets for better battlefield decisions. The modularity and scale of IoBT still entail cybersecurity concerns, like data manipulation and unlawful access. These shortcomings are fixed by our work that innovatively proposes an IDS model integrating ensemble learning and supervised machine learning. Given the CIC-IDS-2017 and CIC-IDS-2018 datasets as benchmarks, the IDS aims for a high anomaly detection rate with few false positives. The methodology shows the possibilities of machine learning in increasing the cybersecurity architecture of the IoBT.

Wang et al. [18] proposed a lightweight botnet detector, BotCapturer, with a two-level analysis mechanism based on graph anomaly detection and packet traffic clustering for detecting the abnormal nodes corresponding to the C&C servers ([18]). Deng et al. [6] proposed a security anomaly detection method for big data platforms using quantum optimization and clustering which presents a framework for a big data platform anomaly detection system that is based on a distributed software architecture and uses server log data to detect network anomalies.

Six different types of machine learning (ML) algorithms were used in [3]: logistic regression (LR), decision trees (DT), Random Forest (RF), naive bayes (NB), K_Nearest_Neighbors (K-NN), and support vector machines (SVM). The CICDDoS2019 dataset was used for the testing, and the decision tree (DT) and random forest (RF) algorithms produced the best results, with 99% and 99%, respectively.

Yaqoob et al. in [19] dealt with the challenges of security threats and abnormal network communication in Fa-IoVs containing the problem of identifying malicious nodes by proposing a dynamic deep learning-based scheme, CAaDet, which uses convolutional layers and a customized autoencoder for feature extraction and anomaly detection.

The authors of [9], describe a new approach using a graph-based neural network (AEN-GNN) model. AEN (Activity and Event Network) can detect both these types of attacks and long-term threats that traditional tools cannot capture. The study used the AEN-GNN model to classify network traffic into normal and anomalous (e.g., traffic from TOR networks). A high detection rate was achieved (76% for DDoS attacks and 88% for TOR/non-TOR traffic).

In [7] the authors considered the application of deep learning and clustering methods to enhance the traditional network traffic detection algorithm in order to increase its efficiency in the detection of anomalies.

The related works demonstrate the complexity and diversity of approaches to detecting anomalies in network traffic, ranging from clustering algorithms to deep learning methods to the most edge processing and IoT-based techniques, with an emphasis on protecting critical infrastructure.

3 METHODOLOGY

3.1 Description of the network datasets used in the research.

This research utilises one of the most recent NetFlow datasets: LUFlow Network Intrusion Detection Data Set [13]. The considered set includes both regular (underlay/benign) and anomalous (suspicious/attack) traffic. These gathered data cover a wide range of attack types, such as malware infections, port scanning, DDoS, etc.

The dataset is a free open benchmark, allowing for reproducibility of the approach presented in this paper.

LUFlow [13] provides a wealth of information on modern suspected activities. It has data collected via honeypot networks on the premises of the University of Lancaster. The traffic was collected with the aim of identifying new attack methods. The employed process ensures the collection of telemetric data. The dataset identifies flows that differ from typical network activity but cannot be definitively labelled as malevolent, classifying them as anomalies to encourage further research. The dataset also features records of normal traffic from different running services, like SSH and database activities.

The collection includes the following flow labels (roughly balanced, imbalance ratio = 1.18):

(1) benign - 516220 flows
(2) outlier - 365385 flows
(3) malicious - 78116 flows

The dataset is structured as Netflow v9, which was developed by Cisco. NetFLow v9 provides flow information, containing attributes such as source and destination IP addresses, port numbers, protocols, packet length, flow time and other important data. An important feature of NetFlow is that because it formulates aggregations of traffic, it is lightweight relative to PCAP data, making the protocol highly useful for near real-time data verification. In addition, Netflow V9 supports a configurable data schema according to the specific needs of the network environment or task for which it is to be used, thus facilitating the collection of detailed network traffic information. What distinguishes the NetFlow version 9 format from other versions is its template-based design. A template allows the format to be customised. One or more FlowSet templates are placed after the package header in the NetFlow version 9 record format.

The main advantage of using this protocol is its ability to represent the behaviour of the network traffic without performing a deep analysis of every packet. This offers the major advantage of preserving user privacy, which is critical from the perspective of many privacy-preserving laws and regulations, like the GDPR.

The research presented in this paper focuses on a subset of fields from the NetFlow traffic. Following our previous work [10], the features were selected for their informativeness in detecting network attacks, and useing a small number of features, influences

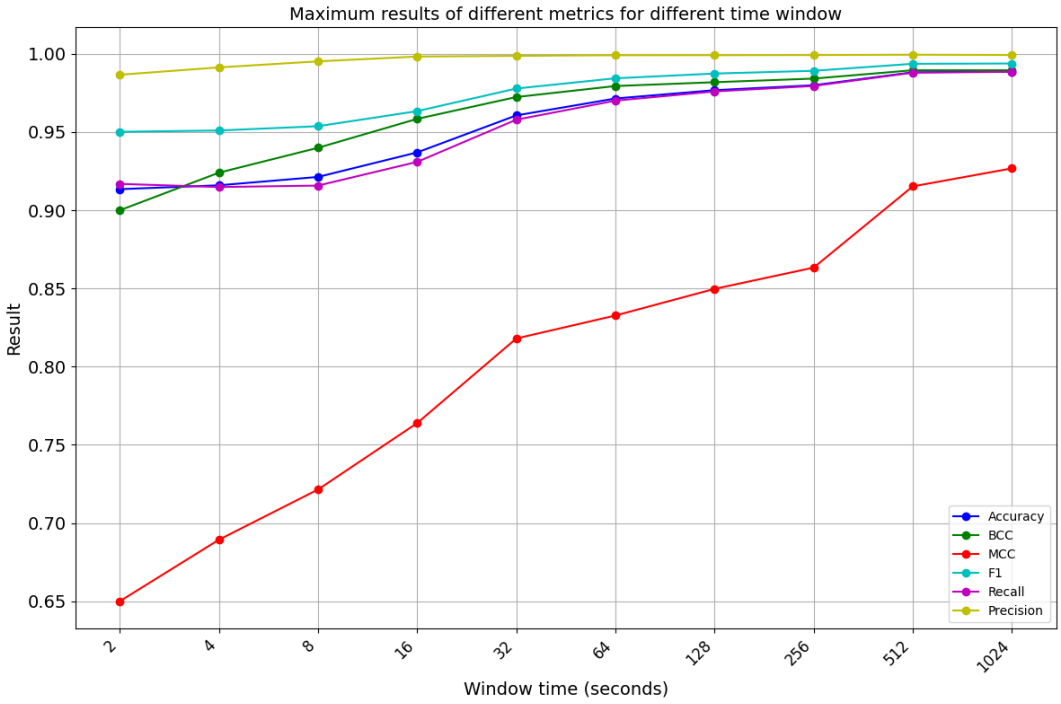


Figure 2: Results of granularization process after the time window

the speed of the model, both when learning and subsequently at inference time.

The following features were used:

(1) bytes_in - The number of bytes transmitted from source to destination
(2) bytes_out - The number of bytes transmitted from destination to source.
(3) num_pkts_in - The packet count from source to destination
(4) num_pkts_out - The packet count from destination to source
(5) time_start - The start time of the flow in seconds since the epoch.
(6) proto - The protocol number associated with the flow.

3.2 The proposed Granulation-by-Time Process

The process of granulating data by time is the main contribution of this paper. The presented approach is based on a systematic application of clustering methods for time windows that organise and group the data based on the time of the datapoint occurrence. As will be showcased in the experimental part, this approach to the process of granulation enables the detection and separation of temporal patterns in network data that may indicate potential security risks.

This research set out to find if temporal granularity provides a more accurate representation of network behaviour, thus enabling the separation of anomalous activities with different effectiveness at different granularity levels.

The following experiments seek to answer if, with proper grouping by time, the detection metrics of attacks, as detected by ML models, improve significantly. Thus, if using the granular approach it is possible to strengthen network security.

The proposed approach is to systematically analyze the impact of granularity on machine learning models for Netflow attacks detection. To achieve that, this work uses time window-based coarse-to-fine groupings. The method can be described as follows: the input features are calculated starting with small time windows - called T_{fine} (e.g., 2-seconds intervals), and gradually increasing the windows to larger lengths, denoted as T_{coarse}. Each experiment aggregates Netflow frames within each interval. The aggregation process is expressed as Eq. 1:

$$X_{\text{window}} = \sum_{i=1}^{n} X_i \tag{1}$$

X_{window} in the formula Eq. 1, is the result of the Netflow data aggregation over the predefined time window. It is obtained by grouping the traffic in a specified time frame, e.g., a 10-second window. Key metrics represented by the NetFlow data, including 'bytes in', 'bytes out', 'num pkts in', 'num pkts out' etc., are summed for each grouping.

3.3 Detection model - Random Forest

Random Forest [5] is an ensemble algorithm that uses aggregates of decision trees for inference. The method has high prediction accuracy and robustness to over-fitting, resulting in its wide application in various areas, including network intrusion detection. Random Forest selects a random subset of the training set to formulate each decision tree. At inference time, each of the trees in the

forest produces a result, and the entire forest produces a result by majority voting. The subsampling with replacement of the training set for the formulation of each classification tree is called 'Bagging" (Bootstrap Aggregating).

This randomness introduced by bagging reduces makes the model less prone to over-fitting.

When growing the decision trees Random Forest repeatedly partitions the dataset into smaller and smaller subsets, using either Entropy (a measure of information uncertainty) or the Gini Index (which is a measure of set impurity) for the decision of there to split the set, which would minimize those metrics.

Thanks to its effectiveness, Random Forest is one of the most popular and widely used machine learning algorithms, with the field of data analysis and prediction being among its many areas of application.

3.4 Evaluation criteria for anomaly detection

Accuracy, Eq. (2) is defined as the proportion of accurately classified datapoints in the test set in relation to the total number of instances.

Precision, Eq. (3), is another standard classification metric. It is the ratio of detected positive cases to all expected positive instances.

Recall, Eq. (4), is the ratio of samples classified as positive to all expected positive instances.

The F1-score, Eq. (5), is a harmonic mean of recall and precision. Better performance is indicated by a higher value of F1, with a value range of 0 to 1.

The MCC (Matthews Correlation Coefficient), Eq. (7), takes into consideration the model's true positives, true negatives, false positives, and false negatives. MCC determines the correlation coefficient between the anticipated and actual classifications. It is a metric well suited to imbalanced data.

The BCC (Balanced Accuracy), Eq. (6), BCC is used for situations with data imbalance.

It represents the arithmetic average of sensitivity (true positive rate) and specificity (true negative rate)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{5}$$

$$BCC = \frac{\frac{TP}{TP+FN} + \frac{TN}{TN+FP}}{2} \tag{6}$$

$$MCC = \frac{TN * TP - FN * FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}$$

3.5 Statistical Analysis

The Wilcoxon test (also referred to as the rank-sum test for two correlated variables) is a nonparametric statistical method used to compare the medians of two groups where the normality of the data distribution is not assumed.

The p-value (p-value) is a measure allowing to evaluate to what extent one can reject the null hypothesis. For the Wilcoxon test, the null hypothesis is that there are no meaningful differences between measured groups. The p-value represents the probability of observing the data in case the null hypothesis was true, meaning there both groups are similar. The smaller the p-value, the more statistically significant the difference between groups is.

4 EXPERIMENTS AND RESULTS

4.1 Description experiment

The purpose of the experiment was to investigate how increasing the size of the time window used for feature aggregation affects the performance of ML-based network intrusion detectors. An overview image showing the data processing pipeline can be found in Figure 1, The loaded data is aggregated time and used to train and test a Random Forest classifier in a 10-fold CV procedure.

The main contribution of this experiment is in the evaluation of different sizes of time windows on the detection results obtained by the classifier. Each subsequent experiment increased the size of time window, as expressed in seconds. The subsequent value was determined by doubling the preceding one, ranging from 2 seconds to 1024 seconds. This results in 10 distinct window sizes, as follows: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024.

To properly assess the reliability of the model, a 10-fold cross-validation was used. This procedure involves dividing the data into ten equal-sized parts, nine of which are used to train the model and the remainder is used for testing.

To determine the statistical significance of differences between results for different granularity levels, the Wilcoxon test was applied. It is a nonparametric statistical test which is used to compare the medians of two independent groups of peered data. It checks to see whether the disparities between the medians are statistically significant.

Results of the experiment were aggregated into various aspects of model quality including accuracy, precision, F1 score, BCC, MCC and recall. They are often employed in the assessment of the efficiency of classification and regression models as they offer a more detailed account of the performance.

Eventually the analysis of the results of the experimentation and the test Wilcoxon allowed to evaluate the influence of the extension of the time window on the performance of data clustering and the quality of prediction of the Random Forest model. The outcomes of this experiment suggest that the practical application of network data analysis may be of much significance in relation to the selection of ideal model parameters for optimal results.

4.2 Results

The results for each time window are found in the summary in Table 1, the table presents the detection metrics in terms of precision, recall and f1 for each class of flows (normal or anomalous). The results are also displayed in Fig. 2 for each measured metric in every measured granularity. It can be seen that as the size of the time window increases, the detection metrics improve. Fig. 3a and 3 display the confusion matrix for the granularity of 1024 seconds. Precision is 0.99, Recall: 0.98, F1: 0.99, MCC: 0.92, BCC: 0.98 and ACC : 0.98 All of the obtained sets of results of each metric for all granularities

were compared against one another using the Wilcoxon test. The comparative analysis of the results for each window showed that among 270 pairs in total, 261 pairs were statistically significantly different from one another. Selected pairs are displayed in Tab.2.

5 CONCLUSION

The purpose of the study was to investigate what effect different time windows have on the detection results of the Random Forest model in network intrusion detection. The results indicate the positive effect of a larger time window on the efficiency of detecting and identifying anomalous traffic. As the time window increased, the model became more suitable for separating different classes of network activity, which corresponded to an increase in detection quality.

The differences proved statistically significant for each reported time window. The p-values of 0.001953125 for all comparisons indicate that the extended time window had an impact on the quality of data clustering and model prediction. The results of the evaluation have important implications for the use of larger time windows in network data analysis. If applied, this could lead to better detection of anomalies and more accurate prediction of events.

The study proves that the length of the time window has an impact on the ability of the Random Forest classifier to detect network intrusions. The results of this experiment can be applied by practitioners working in the field of network data analysis to optimize model parameters for better predictive performance.

The paper showcases the critical role of temporal granularity in enhancing ML-driven NIDS, with an innovative granular computing approach, significantly improving the predictive accuracy of Random Forest classifiers. This research not only advances our understanding of network security dynamics but also sets the stage for future explorations aimed at refining and implementing more robust cybersecurity measures.

ACKNOWLEDGMENTS

The work described in this paper is performed in the H2020 project STARLIGHT ("Sustainable Autonomy and Resilience for LEAs using AI against High priority Threats"). This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101021797.

REFERENCES

[1] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. 2021. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies* 32, 1 (2021), e4150.

[2] Basmh Alkanjr and Thamer Alshammari. 2023. IoBT Intrusion Detection System using Machine Learning. In *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*. 0886–0892. https://doi.org/10.1109/CCWC57344.2023.10099340

[3] Rami J. Alzahrani and Ahmed Alzahrani. 2021. Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic. *Electronics* 10, 23 (2021). https://doi.org/10.3390/electronics10232919

[4] Andrzej Bargiela and Witold Pedrycz. 2022. Granular computing. In *HANDBOOK ON COMPUTER LEARNING AND INTELLIGENCE: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation*. World Scientific, 97–132.

[5] Gérard Biau. 2010. Analysis of a Random Forests Model. https://doi.org/10.48550/ARXIV.1005.0208

[6] Lijuan Deng, Long Wan, and Jian Guo. 2022. Research on Security Anomaly Detection for Big Data Platforms Based on Quantum Optimization Clustering. *Mathematical Problems in Engineering* 2022 (26 Aug 2022), 4805035. https://doi.org/10.1155/2022/4805035

[7] Qian He. 2021. Research on Network Traffic Anomaly Detection Based on Deep Learning. In *2021 International Conference on Networking, Communications and Information Technology (NetCIT)*. 50–53. https://doi.org/10.1109/NetCIT54147.2021.00017

[8] Suleman Khan, Abdullah Gani, Ainuddin Wahid Abdul Wahab, and Prem Kumar Singh. 2018. Feature Selection of Denial-of-Service Attacks Using Entropy and Granular Computing. *Arabian Journal for Science and Engineering* 43, 2 (01 Feb 2018), 499–508. https://doi.org/10.1007/s13369-017-2634-8

[9] Patrice Kisanga, Isaac Woungang, Issa Traore, and Glaucio H. S. Carvalho. 2023. Network Anomaly Detection Using a Graph Neural Network. In *2023 International Conference on Computing, Networking and Communications (ICNC)*. 61–65. https://doi.org/10.1109/ICNC57223.2023.10074111

[10] Mikołaj Komisarek, Marek Pawlicki, Rafał Kozik, Witold Hołubowicz, and Michał Choraś. 2021. How to effectively collect and process network data for intrusion detection? *Entropy* 23, 11 (2021), 1532.

[11] Tim Krause, Raphael Ernst, Benedikt Klaer, Immanuel Hacker, and Martin Henze. 2021. Cybersecurity in power grids: Challenges and opportunities. *Sensors* 21, 18 (2021), 6225.

[12] Sihong Lin, Kunbin Zhang, Dun Guan, Linjie He, and Yumin Chen. 2023. An intrusion detection method based on granular autoencoders. *Journal of Intelligent & Fuzzy Systems* 44 (2023), 8413–8424. https://doi.org/10.3233/JIFS-223649 5.

[13] Ryan Mills. 2024. LUFlow Network Intrusion Detection Data Set. https://doi.org/10.34740/KAGGLE/DSV/7594192

[14] Aleksandra Pawlicka, Michał Choraś, and Marek Pawlicki. 2021. The stray sheep of cyberspace aka the actors who claim they break the law for the greater good. *Personal and Ubiquitous Computing* 25, 5 (2021), 843–852.

[15] Marek Pawlicki, Michał Choraś, and Rafał Kozik. 2018. Recent Granular Computing Implementations and its Feasibility in Cybersecurity Domain. In *Proceedings of the 13th International Conference on Availability, Reliability and Security* (Hamburg, Germany) *(ARES '18)*. Association for Computing Machinery, New York, NY, USA, Article 61, 6 pages. https://doi.org/10.1145/3230833.3233259

[16] Marek Pawlicki, Rafał Kozik, and Michał Choraś. 2022. A survey on neural networks for (cyber-) security and (cyber-) security of neural networks. *Neurocomputing* 500 (2022), 1075–1087.

[17] Nasrin Sultana, Naveen Chilamkurti, Wei Peng, and Rabei Alhadad. 2019. Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications* 12 (2019), 493–501.

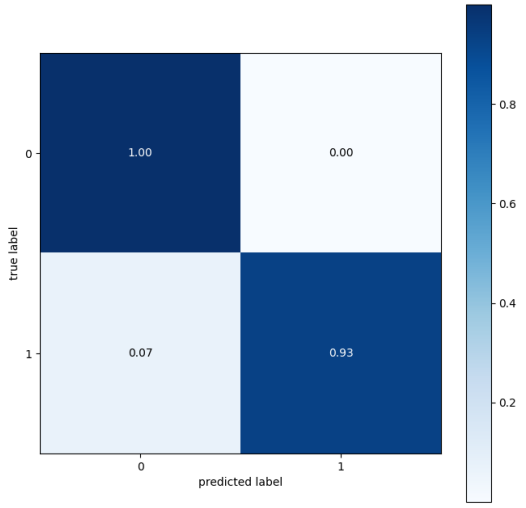
[18] Wei Wang, Yang Wang, Xinlu Tan, Ya Liua, and Shuangmao Yang. 2018. BotCapturer: Detecting Botnets based on Two-Layered Analysis with Graph Anomaly Detection and Network Traffic Clustering. *International Journal of Performability Engineering* (2018). https://doi.org/10.23940/ijpe.18.05.p24.10501059

[19] Shumayla Yaqoob, Asad Hussain, Fazli Subhan, Giuseppina Pappalardo, and Muhammad Awais. 2023. Deep Learning Based Anomaly Detection for Fog-Assisted IoVs Network. *IEEE Access* 11 (2023), 19024–19038. https://doi.org/10.1109/ACCESS.2023.3246660

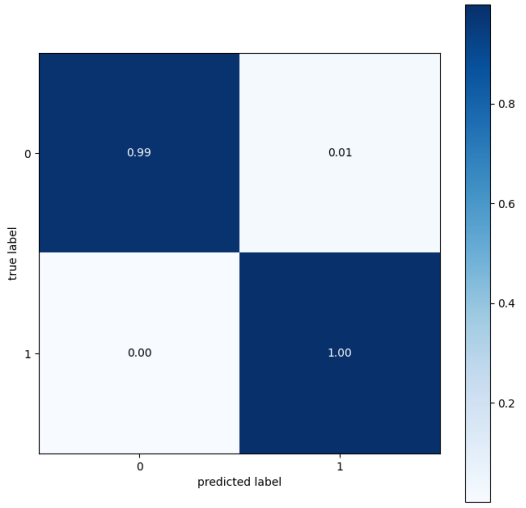
[20] Hao Zhang, Yongdan Li, Zhihan Lv, Arun Kumar Sangaiah, and Tao Huang. 2020. A real-time and ubiquitous network attack detection based on deep belief network and support vector machine. *IEEE/CAA Journal of Automatica Sinica* 7, 3 (2020), 790–799. https://doi.org/10.1109/JAS.2020.1003099

Table 1: Summary presentation of results for each time window

Time Window (seconds)	0 - normal 1 - anomaly	precision	recall	f1-score	support
2	0	0.92	0.87	0.89	481357
	1	0.99	0.99	0.99	4315721
4	0	0.94	0.89	0.92	328850
	1	0.99	0.99	0.99	2690547
8	0	0.94	0.92	0.93	214201
	1	0.99	0.99	0.99	1657442
16	0	0.94	0.94	0.94	122220
	1	0.99	0.99	0.99	987930
32	0	0.94	0.95	0.94	58508
	1	0.99	0.99	0.99	560737
64	0	0.93	0.95	0.94	24113
	1	1.00	0.99	0.99	303399
128	0	0.93	0.95	0.94	11426
	1	1.00	0.99	1.00	156271
256	0	0.95	0.95	0.95	5577
	1	1.00	1.00	1.00	79235
512	0	0.98	0.97	0.97	2899
	1	1.00	1.00	1.00	40032
1024	0	0.99	0.99	0.99	1694
	1	1.00	1.00	1.00	20231



(a) The result of Random Forest - Confusion Matrix for 8 second window granular grouping



(b) The result of Random Forest - Confusion Matrix for 1024 second window granular grouping

Figure 3: Results of granularization process after the time window

Table 2: Pairs of windows with statistically significant differences in the during the Wilcoxon Test, for brevity reasons, only pairings of windows sie 512 and 1024 are presented

Pair No.	Windows	Metric	Average Values in 10f CV	Which is Better
45	512 vs 1024	MCC	0.9153 vs 0.9268	1024
90	512 vs 1024	ACC	0.9882 vs 0.9886	1024
135	512 vs 1024	BCC	0.9895 vs 0.9895	512
180	512 vs 1024	F1	0.9936 vs 0.9938	1024
225	512 vs 1024	RECALL	0.988 vs 0.9885	1024
270	512 vs 1024	PRECISION	0.9994 vs 0.9992	512