The Impact of Data Scaling Approaches on Deep Learning, Random Forest and Nearest Neighbour-based Network Intrusion Detection Systems for DoS Detection in IoT Networks

Marek Pawlicki, Rafal Kozik, and Michal Choraś

Bydgoszcz University of Science and Technology, Bydgoszcz, Poland
{marek.pawlicki, rafal.kozik, chorasm}@pbs.edu.pl

Abstract

In the rapidly expanding realm of the Internet of Things, network security is of paramount importance, especially in the face of an increasing number of DoS attacks leveraging IoT devices. This paper examines the underexplored area of the impact of data scaling approaches on the effectiveness of machine learning-based Network Intrusion Detection Systems in detecting DoS attacks in IoT networks. Specifically, it evaluates the performance of three classifier algorithms, K-Nearest Neighbour, RandomForest, and Deep Neural Networks, on three different datasets, focusing on how distinct feature scaling methods influence detection capabilities. Through a comprehensive experiment, the paper finds that the choice of scaling method can significantly impact the performance of the NIDS. Results vary across datasets and algorithms; for example, the 'Standard' scaling generally outperforms others for ANNs in one dataset, while the 'Quantile' and 'Power' scalings are more effective for ANNs in another. This work fills the gap in the existing research on the machine-learning-based network intrusion detection and has the potential to guide the development of intrusion detection systems, particularly in the complex and vulnerable landscape of IoT.

Keywords: Network Intrusion Detection, Data Scaling, Machine Learning, Preprocessing, Feature Engineering

1 Introduction

In the evolving landscape of the Internet of Things (IoT), where an ever-growing number of interconnected devices participate in data exchange—from personal smartphones to industrial control systems—the urgency of robust network security measures cannot be overstated [1]. A 2023 report reveals that the Denial of Service (DoS) attacks involving IoT bots have seen a staggering five-fold increase within a year [2], spotlighting IoT devices as susceptible vectors for a gamut of network threats such as data theft, phishing, and spoofing [2][3]. Notably, cyber incidents like the Mirai, Hajime, BusyBox, BrickerBot [4], Reaper and Persirai Botnets, and many others [5] have emphasized these vulnerabilities, exploiting IoT devices to launch devastating DDoS attacks or, in some cases, rendering them entirely unusable [6].

These events emphasize not just the potential risks but also the breadth of the attack surface that IoT introduces. Thus, they signify an urgent necessity for innovative security solutions. With a range of motivations for cyberattacks [7], as the complexity and volume of network data escalate, machine learning techniques offer promising avenues for enhancing network intrusion

detection systems (NIDS), particularly in identifying DoS attacks perpetrated through vulnerable IoT devices [8]. While machine learning algorithms have exhibited significant promise in enhancing the performance of NIDS [9], also in IoT settings [10], their effectiveness is heavily influenced by the quality of the utilised data [11]. Unprocessed or poorly scaled data can impact the algorithms' capacity to detect network intrusions significantly. For example, some machine learning algorithms are sensitive to the scale of feature values [12]; a feature that ranges between 0 and 1 can disproportionately influence the outcome if another feature ranges between 0 and 1000. This underscores the need for a rigorous understanding of the data scaling techniques in the context of NIDS. Effective data scaling is not merely a preprocessing step; it is a fundamental aspect that could greatly influence the detection capabilities of the entire system [13], thus bearing direct implications for IoT security. Despite the pervasive adoption of machine learning in NIDS, there remains a gap in research focusing on the role of data scaling in this domain, especially in the increasingly vulnerable landscape of IoT. Most existing studies have predominantly centred on feature selection, algorithmic optimization, or hyperparameter tuning, often overlooking the initial yet critical phase of data scaling. This void in research is all the more pertinent given the recent spike in DoS attacks involving IoT bots and the distinct vulnerabilities that IoT devices introduce into network systems. A nuanced understanding of how different data scaling techniques impact the performance of machine learning-based NIDS could provide valuable insights for both researchers and practitioners. It can also guide the development of more robust and adaptive intrusion detection systems that are well-suited to the dynamic and complex nature of IoT networks.

This paper provides a comprehensive evaluation of five feature scaling methods tested on DoS/DDoS samples and Benign samples extracted from three different NetFlow-based Network Intrusion Detection open benchmark datasets applied as applied to three different ML algorithms: K-Nearest Neighbour (KNN), RandomForest(RF) and Deep Neural Networks (DNN). By providing a comprehensive evaluation of various feature scaling methods, specifically focusing on DoS/DDoS and Benign samples across multiple open benchmark datasets and machine learning algorithms, this paper contributes to the contemporary state of the art and addresses a significant gap in the current understanding of ML-based NIDS effectiveness in an IoT context. This work has the potential to shift the way researchers and practitioners approach the development and optimisation of NIDS, particularly in the increasingly complex and vulnerable domain of IoT.

While the study focuses on NIDS, the methodologies and findings are also applicable to Host-based Intrusion Detection Systems (HIDS). The scalability and adaptability of the evaluated feature scaling techniques make them suitable for deployment in HIDS, where data characteristics and attack patterns may differ from network traffic.

The paper is structured as follows: Section 2 provides the Related Works with emphasis on previous work on machine learning in NIDS and existing data preprocessing techniques in NIDS, identifying the existing research gaps. Section 3 describes the used benchmarks, ML methods and data scaling approaches. The experimental setup and results are showcased in Section 4, the paper wraps up with the conclusions in Section 5.

2 Related Works

While ML-based NIDS is a widely researched domain, with 21900 search results on Google Scholar as of the time of writing this paper, the topic of the influence of data scaling as a preprocessing step is a relatively unexplored topic. In [14], the authors evaluate the effect of three different scalers on the classification accuracy of one-class classifiers in intrusion detection.

In [15], the authors experiment with using different scalers after applying a log transformation of the data in their fast intrusion detection approach. The authors of [16] indicate that their experiment points to the normalization steps having a positive effect on the accuracy of most classifiers. A comprehensive evaluation of the impact of fourteen data normalization techniques on classification is presented in [17], where a set of best and worst methods is proposed.

In contrast to the related works, this study focuses on testing the influence of scaling algorithms on a range of algorithms (KNN, RF, ANN) and specifically using real-world NIDS data. On top of that, since most real-world NIDS datasets suffer from the imbalance problem, solely reporting accuracy may be misleading. This study evaluated metrics suited to classification in an unbalanced scenario, most importantly the Balanced Accuracy (BAC) and Matthews Correlation Coefficient (MCC). Those are derived from the confusion matrix, which contains the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This study also includes categorical features. Finally, the achieved outcome is rigorously tested with the Wilcoxon signed-rank test to establish the significance of the results.

3 Materials and Methods

Three datasets were utilised in this study; two come from the NetFlow Datasets collection [18], the third one is the SIMARGL2021 Dataset [9].

The **NF-UQ-NIDS**, commonly known as the NetFlow Dataset Collection, came about in the pursuit of fostering a more robust NIDS. By amalgamating multiple smaller datasets and calculating the aggregate NetFlow features, this comprehensive collection augments the scope and applicability of NIDS research, offering a universal dataset encompassing flows from diverse network and attack configurations, and facilitating comparative analyses across different testbed networks for similar attack scenarios. In this paper, the NF-BoT and NF-ToN subsets were used.

The **SIMARGL2021** dataset is sourced from a real-world academic network; it adheres to the Netflow v9 format. It incorporates a comprehensive array of 44 unique features along with labels for each frame. The dataset encompasses various attack types in addition to normal traffic, offering a diversified landscape for NIDS research. The dataset involves vulnerable systems running on different operating systems, an attacker network managed with Kali Linux, and legitimate client traffic. This dataset comprises over 6.5 million frames of non-infected base traffic and approximately 5.6 million frames labelled as attacks. Since SIMARGL2021 is not specifically an IoT dataset, it will serve as a cross-check of the study findings.

In the presented study, the following data scaling approaches were adopted:

3.0.1 Standard Scaler

The Standard Scaler [19] removes the mean from the data, to centre it on zero, and then divides by the standard deviation to get to the unit variance, as described in Equation 1, where X_{scaled} is the standardized value, X is the original value of the data point, μ is the mean of the feature and σ is the standard deviation of the feature from which the data point comes.

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma} \tag{1}$$

3.0.2 Min-Max Scaler

This approach removes the minimum value from the data and divides by the max-min range, effectively scaling the data to the [0,1] range. Then, the standardised value can be multiplied by the desired (max-min) range and shifted to min, as seen in Equation 2.

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \times (\text{max} - \text{min}) + \text{min} \tag{2}$$

3.0.3 MaxAbs Scaler

This approach divides each element in a feature by the absolute maximum value of that feature, as seen in Equation 3. This has the effect of restricting the range to [-1,1], without shifting and centring the values.

$$\text{scaled\_x} = \frac{x}{|\text{max\_abs}|} \tag{3}$$

3.0.4 Power Transformer

This approach applies a mathematical function (usually some power function) to a feature to get its distribution into a bell curve. SciKit-Learn offers two options, Box-Cox Transformation and the Yeo-Johnson Transformation. This paper uses the latter as it can work with both positive and negative values. The Yeo-Johnson transformation of a single data point X is defined as Equation 4, where X refers to a sample of data for a particular feature and λ controls the "power" in the power transformation. The objective is to find the best λ that makes the transformed data y as close as possible to a normal distribution. In Yeo-Johnson, λ is estimated from the data using maximum likelihood estimation, which will optimise the parameter iteratively maximising the function in Equation 5 to arrive at a normal distribution, where n is the number of data points and *SSE* is the sum of squared errors between the transformed data and its mean.

$$y = \begin{cases} \frac{(X+1)^{\lambda}-1}{\lambda} & \text{if } X \geq 0, \lambda \neq 0 \\ \log(X+1) & \text{if } X \geq 0, \lambda = 0 \\ -\frac{(-X+1)^{\lambda}-1}{\lambda} & \text{if } X < 0, \lambda \neq 2 \\ -\log(-X+1) & \text{if } X < 0, \lambda = 2 \end{cases} \tag{4}$$

$$L(\lambda) = -\frac{n}{2}\log(\text{SSE}) + (\lambda - 1)\sum_{i=1}^{n}\log(|x_i| + 1) \tag{5}$$

3.0.5 Quantile Transformer

This scaler breaks up the data into quantiles and uses the cumulative distribution function (CDF) to map the values to a desired output distribution. Because the transformation is based on rank and quantiles rather than actual values, it is less sensitive to outliers than the scalers based on extremes and standard deviation. The transformation does not depend on the scale of the features and it is non-linear, which may distort linear relationships but preserves the monotonicity.

The work presented in this paper utilizes the following machine learning algorithms:

k-Nearest Neighbour is an instance-based learning algorithm where the function is approximated locally and all computation is deferred until classification. The algorithm measures the distance of a datapoint to each datapoint in the set, sorts the distances it in an ascending

order and uses a majority vote of the closest k datapoints to assign the label of the test sample [20].

Random Forest is an ensemble technique which trains an array of simpler classifiers - decision trees - on randomly subsampled parts of the training set and aggregates the decision of this array of decision trees (e.g. with a majority vote) to arrive at the classification of the test sample [21][22].

Deep Neural Networks are a set of popular learning algorithms named to describe the multi-layered structure of computational nodes connected by adjustable weights. The weights can be trained with the error backpropagation algorithm [23] to fit a function to a set of data, a procedure which produces a classifier with astonishing learning ability [24][25].

Evaluation Metrics In the presented work, the standard performance metrics in NIDS research were measured: Accuracy (ACC), Balanced Accuracy (BAC), Precision, Recall, F1-Score, Matthews Correlation Coefficient (MCC), as defined in [25]. For brevity reasons, this paper presents the BAC and MCC.

Wilcoxon Test is a non-parametric statistical test used to compare two sample distributions, applied when the data violate the assumptions of the paired t-test. It ranks the absolute differences between paired datapoints and calculates a test statistic to find if the differences are significant. This study opted for the Wilcoxon test over the standard ANOVA test due to its suitability for non-parametric data and its effectiveness in handling the non-normal distributions.

4 Experiments

4.1 Experimental Setup

The experimental setup consisted of evaluating the performance of three ML algorithms: ANN, k-NN and RF, on three distinct datasets: NF-BoT, NF-ToN, and SIMARGL2021. The datasets contain varying types of network flow information, however for the experiment only DoS and Benign traffic was used. The classifiers were selected for different characteristics: KNN is a simple classifier, RF is relatively robust against overfitting, ANN has the potential to find very complex relationships among features. A standard data preprocessing pipeline was employed. This involved filling missing values and encoding categorical variables using frequency encoding.

A 10-fold stratified cross-validation approach was used for the experiment, with the training folds balanced with SMOTE for BoT and ToN datasets and Random Subsampling the SIMARGL dataset, since it has significantly more Benign samples. Then, the balanced folds were scaled with the evaluated five methods and for each method, a separate instance of each of the three classifiers was trained, and tested on the test fold. The metrics were collected for each fold. Having finished the 10-fold CV, when obtained metrics for each Algorithm/Dataset combination were pitted against one another in a Wilcoxon test.

4.2 Results and Statistical Evaluation

Table 1: The pairs of scalers which reject the null hypothesis during the Wilcoxon test, classifier: ANN

No.	Algorithm	Dataset	Metric	Rejected Hypothesis	Which is Better
1	ANN	BoT	BAC	Standard vs Quantile	Standard
2	ANN	BoT	BAC	Standard vs Power	Standard
3	ANN	BoT	BAC	Min-Max vs Quantile	Min-Max
4	ANN	BoT	BAC	Min-Max vs Power	Min-Max

Table 1: The pairs of scalers which reject the null hypothesis during the Wilcoxon test, classifier: ANN

No.	Algorithm	Dataset	Metric	Rejected Hypothesis	Which is Better
5	ANN	BoT	BAC	Max-Abs vs Quantile	Max-Abs
6	ANN	BoT	BAC	Max-Abs vs Power	Max-Abs
7	ANN	BoT	BAC	Quantile vs Power	Quantile
8	ANN	BoT	MCC	Standard vs Quantile	Standard
9	ANN	BoT	MCC	Standard vs Power	Standard
10	ANN	BoT	MCC	Min-Max vs Quantile	Min-Max
11	ANN	BoT	MCC	Min-Max vs Power	Min-Max
12	ANN	BoT	MCC	Max-Abs vs Quantile	Max-Abs
13	ANN	BoT	MCC	Max-Abs vs Power	Max-Abs
14	ANN	BoT	MCC	Quantile vs Power	Quantile
15	ANN	ToN	BAC	Standard vs Quantile	Quantile
16	ANN	ToN	BAC	Standard vs Power	Power
17	ANN	ToN	BAC	Min-Max vs Quantile	Quantile
18	ANN	ToN	BAC	Min-Max vs Power	Power
19	ANN	ToN	BAC	Max-Abs vs Quantile	Quantile
20	ANN	ToN	BAC	Max-Abs vs Power	Power
21	ANN	ToN	BAC	Quantile vs Power	Power
22	ANN	ToN	MCC	Standard vs Quantile	Quantile
23	ANN	ToN	MCC	Standard vs Power	Power
24	ANN	ToN	MCC	Min-Max vs Quantile	Quantile
25	ANN	ToN	MCC	Min-Max vs Power	Power
26	ANN	ToN	MCC	Max-Abs vs Quantile	Quantile
27	ANN	ToN	MCC	Max-Abs vs Power	Power
28	ANN	ToN	MCC	Quantile vs Power	Power
29	ANN	SIMARGL	BAC	Standard vs Power	Standard
30	ANN	SIMARGL	BAC	Min-Max vs Power	Min-Max
31	ANN	SIMARGL	MCC	Standard vs Power	Standard
32	ANN	SIMARGL	MCC	Min-Max vs Quantile	Quantile
33	ANN	SIMARGL	MCC	Min-Max vs Power	Min-Max
34	ANN	SIMARGL	MCC	Max-Abs vs Quantile	Quantile
35	ANN	SIMARGL	MCC	Max-Abs vs Power	Max-Abs
36	ANN	SIMARGL	MCC	Quantile vs Power	Quantile

Table 2: The pairs of scalers which reject the null hypothesis during the Wilcoxon test, classifier: KNN

No.	Algorithm	Dataset	Metric	Rejected Hypothesis	Which is Better
37	KNN	ToN	BAC	Standard vs Min-Max	Standard
38	KNN	ToN	BAC	Standard vs Max-Abs	Standard
39	KNN	ToN	BAC	Standard vs Quantile	Standard
40	KNN	ToN	BAC	Standard vs Power	Power
41	KNN	ToN	BAC	Min-Max vs Quantile	Quantile
42	KNN	ToN	BAC	Min-Max vs Power	Power
43	KNN	ToN	BAC	Max-Abs vs Quantile	Quantile
44	KNN	ToN	BAC	Max-Abs vs Power	Power
45	KNN	ToN	BAC	Quantile vs Power	Power
46	KNN	ToN	MCC	Standard vs Min-Max	Standard
47	KNN	ToN	MCC	Standard vs Max-Abs	Standard
48	KNN	ToN	MCC	Standard vs Quantile	Standard
49	KNN	ToN	MCC	Standard vs Power	Power
50	KNN	ToN	MCC	Min-Max vs Quantile	Quantile
51	KNN	ToN	MCC	Min-Max vs Power	Power
52	KNN	ToN	MCC	Max-Abs vs Quantile	Quantile
53	KNN	ToN	MCC	Max-Abs vs Power	Power
54	KNN	ToN	MCC	Quantile vs Power	Power
55	KNN	SIMARGL	BAC	Standard vs Min-Max	Min-Max
56	KNN	SIMARGL	BAC	Standard vs Max-Abs	Max-Abs
57	KNN	SIMARGL	BAC	Standard vs Quantile	Quantile
58	KNN	SIMARGL	BAC	Standard vs Power	Standard
59	KNN	SIMARGL	BAC	Min-Max vs Max-Abs	Min-Max
60	KNN	SIMARGL	BAC	Min-Max vs Quantile	Min-Max
61	KNN	SIMARGL	BAC	Min-Max vs Power	Min-Max
62	KNN	SIMARGL	BAC	Max-Abs vs Quantile	Max-Abs
63	KNN	SIMARGL	BAC	Max-Abs vs Power	Max-Abs
64	KNN	SIMARGL	BAC	Quantile vs Power	Quantile
65	KNN	SIMARGL	MCC	Standard vs Min-Max	Min-Max
66	KNN	SIMARGL	MCC	Standard vs Max-Abs	Max-Abs

Table 2: The pairs of scalers which reject the null hypothesis during the Wilcoxon test, classifier: KNN

No.	Algorithm	Dataset	Metric	Rejected Hypothesis	Which is Better
67	KNN	SIMARGL	MCC	Standard vs Quantile	Quantile
68	KNN	SIMARGL	MCC	Standard vs Power	Standard
69	KNN	SIMARGL	MCC	Min-Max vs Max-Abs	Min-Max
70	KNN	SIMARGL	MCC	Min-Max vs Quantile	Min-Max
71	KNN	SIMARGL	MCC	Min-Max vs Power	Min-Max
72	KNN	SIMARGL	MCC	Max-Abs vs Quantile	Max-Abs
73	KNN	SIMARGL	MCC	Max-Abs vs Power	Max-Abs
74	KNN	SIMARGL	MCC	Quantile vs Power	Quantile

Table 3: The pairs of scalers which reject the null hypothesis during the Wilcoxon test, classifier: RF.

No.	Algorithm	Dataset	Metric	Rejected Hypothesis	Which is Better
75	RF	BoT	BAC	Max-Abs vs Quantile	Quantile
76	RF	BoT	BAC	Quantile vs Power	Quantile
77	RF	BoT	MCC	Max-Abs vs Quantile	Quantile
78	RF	BoT	MCC	Quantile vs Power	Quantile
79	RF	ToN	BAC	Standard vs Min-Max	Standard
80	RF	ToN	BAC	Standard vs Max-Abs	Standard
81	RF	ToN	BAC	Min-Max vs Quantile	Quantile
82	RF	ToN	BAC	Min-Max vs Power	Power
83	RF	ToN	BAC	Max-Abs vs Quantile	Quantile
84	RF	ToN	BAC	Max-Abs vs Power	Power
85	RF	ToN	MCC	Standard vs Min-Max	Standard
86	RF	ToN	MCC	Standard vs Max-Abs	Standard
87	RF	ToN	MCC	Min-Max vs Quantile	Quantile
88	RF	ToN	MCC	Min-Max vs Power	Power
89	RF	ToN	MCC	Max-Abs vs Quantile	Quantile
90	RF	ToN	MCC	Max-Abs vs Power	Power
91	RF	SIMARGL	BAC	Min-Max vs Max-Abs	Min-Max
92	RF	SIMARGL	BAC	Min-Max vs Power	Min-Max
93	RF	SIMARGL	BAC	Max-Abs vs Quantile	Quantile
94	RF	SIMARGL	BAC	Quantile vs Power	Quantile
95	RF	SIMARGL	MCC	Min-Max vs Power	Min-Max
96	RF	SIMARGL	MCC	Max-Abs vs Quantile	Quantile
97	RF	SIMARGL	MCC	Quantile vs Power	Quantile

Tables 1, 2 and 3 gather the results of the Wilcoxon test for RF. Out of 180 pairs of calculated tests, the results of the 98 significant results were found with the Wilcoxon test. A sample Boxplot for the BAC of the KNN classifier over the BoT dataset is showcased in Fig. 1. For ANNs used on the BoT dataset, 'Standard' scaling consistently outperforms 'Quantile' and 'Power' in terms of both BAC and MCC metrics. On the other hand, for the ToN dataset, the 'Quantile' and 'Power' scaling techniques have the upper hand. For KNN, the 'Standard' scaling seems generally superior in the ToN dataset, especially when compared with 'Min-Max' and 'Max-Abs', but the results shift when it comes to the SIMARGL dataset. In the case of RF, the 'Quantile' scaling has a slight edge in many of the comparisons across datasets, especially in the BoT and SIMARGL datasets.

The behaviour of the Random Forest algorithm is especially worth emphasising, as it is generally considered robust to the scale of input variables. The experiment suggests that the particular preprocessing transformations of the input data can introduce nuances which impact RF performance. The authors speculate that the 'Quantile' transformer's ability to force data into a bell curve shape might indirectly affect RF's effectiveness at partitioning the data. The 'Quantile' transformer also handles outliers, which might be helpful for RF.

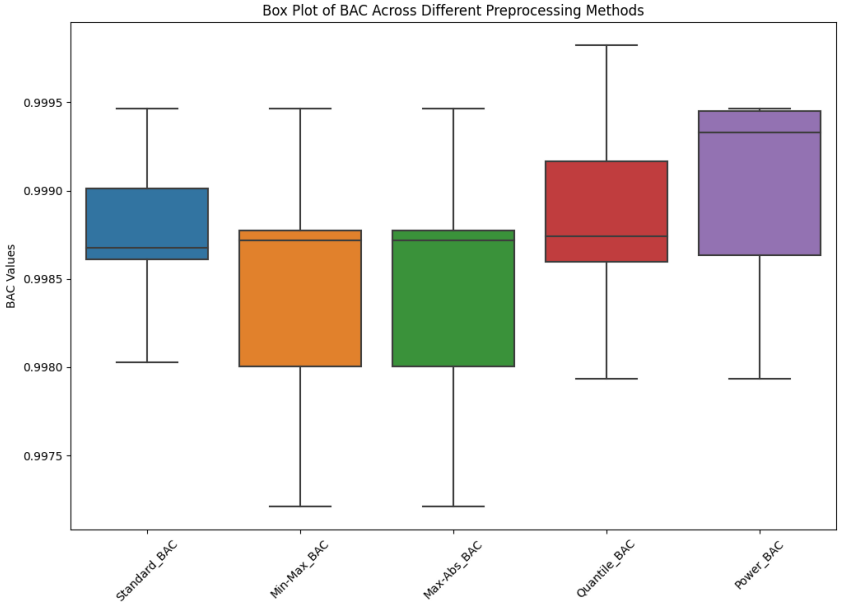

Figure 1: Boxplot of achieved Balanced Accuracy for all the scalers, classifier: KNN, dataset: BoT.

5 Conclusion

The conducted experiments and the subsequent statistical tests showcase that there is no one-size-fits-all solution: the optimal scaling method can depend on the selected classifier and the characteristics of the used dataset. Therefore, when developing an ML model, the experiments suggest that it is beneficial to evaluate multiple scaling techniques to identify which one is the most appropriate for the specific needs of the processing pipeline. The comprehensive experiments reveal the distinct impact of each scaling technique on the performance of the evaluated machine learning algorithms. The results of the Wilcoxon test further substantiate the findings, highlighting the statistical significance of the observations and providing valuable insights for future research in scaling methods for intrusion detection systems.

Acknowledgement

This work is funded under the ELEGANT project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957286.

References

[1] Famous ddos attacks — biggest ddos attacks — cloudflare. https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/. (Accessed on 09/09/2023).

[2] Ddos attacks using iot bots have jumped five-fold in 12 months, says report. https://www.rcrwireless.com/20230607/internet-of-things-4/ddos-attacks-using-iot-bots-have-jumped-five-fold-in-12-months-says-report. (Accessed on 09/09/2023).

[3] Iot security: 5 cyber-attacks caused by iot security vulnerabilities. https://www.cm-alliance.com/cybersecurity-blog/iot-security-5-cyber-attacks-caused-by-iot-security-vulnerabilities. (Accessed on 09/09/2023).

[4] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[5] Pierre-Antoine Vervier and Yun Shen. Before toasters rise up: A view into the emerging iot threat landscape. In *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21*, pages 556–576. Springer, 2018.

[6] Muhammad Waqas, Kamlesh Kumar, Asif Ali Laghari, Umair Saeed, Muhammad Malook Rind, Aftab Ahmed Shaikh, Fahad Hussain, Athaul Rai, and Abdul Qayoom Qazi. Botnet attack detection in internet of things devices over cloud environment via machine learning. *Concurrency and Computation: Practice and Experience*, 34(4):e6662, 2022.

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

[8] Nadia Chaabouni, Mohamed Mosbah, Akka Zemmari, Cyrille Sauvignac, and Parvez Faruki. Network intrusion detection for iot security based on learning techniques. *IEEE Communications Surveys & Tutorials*, 21(3):2671–2701, 2019.

[9] Maria-Elena Mihailescu, Darius Mihai, Mihai Carabas, Mikołaj Komisarek, Marek Pawlicki, Witold Hołubowicz, and Rafał Kozik. The proposition and evaluation of the roedunet-simargl2021 network intrusion detection dataset. *Sensors*, 21(13):4319, 2021.

[10] Vibekananda Dutta, Michal Choras, Marek Pawlicki, and Rafal Kozik. Detection of cyberattacks traces in iot data. *J. Univers. Comput. Sci.*, 26(11):1422–1434, 2020.

[11] Gints Engelen, Vera Rimmer, and Wouter Joosen. Troubleshooting an intrusion detection dataset: the cicids2017 case study. In *2021 IEEE Security and Privacy Workshops (SPW)*, pages 7–12. IEEE, 2021.

[12] Xing Wan. Influence of feature scaling on convergence of gradient iterative algorithm. In *Journal of physics: Conference series*, volume 1213, page 032021. IOP Publishing, 2019.

[13] Dilber Uzun Ozsahin, Mubarak Taiwo Mustapha, Auwalu Saleh Mubarak, Zubaida Said Ameen, and Berna Uzun. Impact of feature scaling on machine learning models for the diagnosis of diabetes. In *2022 International Conference on Artificial Intelligence in Everything (AIE)*, pages 87–94. IEEE, 2022.

[14] Siti-Farhana Lokman, Abu Talib Othman, Muhamad Husaini Abu Bakar, and Shahrulniza Musa. The impact of different feature scaling methods on intrusion detection for in-vehicle controller area network (can). In *Advances in Cyber Security: First International Conference, ACeS 2019, Penang, Malaysia, July 30–August 1, 2019, Revised Selected Papers 1*, pages 195–205. Springer, 2020.

[15] Abhilash Singh, J Amutha, Jaiprakash Nagar, Sandeep Sharma, and Cheng-Chi Lee. Lt-fs-id: Log-transformed feature learning and feature-scaling-based machine learning algorithms to predict the k-barriers for intrusion detection using wireless sensor network. *Sensors*, 22(3):1070, 2022.

[16] Abhijeet Sahu, Zeyu Mao, Katherine Davis, and Ana E Goulart. Data processing and model selection for machine learning-based network intrusion detection. In *2020 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, pages 1–6. IEEE, 2020.

[17] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.

[18] Mohanad Sarhan, Siamak Layeghy, Nour Moustafa, and Marius Portmann. Netflow datasets for machine learning-based network intrusion detection systems. In *Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10*, pages 117–135. Springer, 2021.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Evelyn Fix and JL Hodges. Usaf school of aviation medicine. *Discriminatory analysis, nonparametric discrimination: consistency properties," Randolph Field, Tex.: USAF School of Aviation Medicine*, 1951.

[21] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[22] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[23] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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