From Network Traffic to Data Space: Design, Validation, and Multi-Model Benchmarking

Julian Graf*, Murad Hachani*, Christoph Moser*, Sebastian Fischer*, Rudolf Hackenberg*
*Department of Computer Science and Mathematics
University of Applied Sciences Regensburg
Regensburg, Germany
Email: {julian.graf, christoph.moser, rudolf.hackenberg, sebastian.fischer}@oth-regensburg.de;
murad.hachani@st.oth-regensburg.de

***Abstract*—Encrypted communications and the increasing diversity of cyber-threats challenge traditional Intrusion Detection Systems (IDSs). This paper evaluates a novel network traffic analysis model for intrusion detection. The model offers a prioritized packet processing approach for resource-constrained environments and near real-time analysis of encrypted network traffic metadata. The model defines Primary Target Group (PTG) characteristics to process and structure Internet Protocol (IP) network packets. The corresponding model architecture reflects a metadata driven data space. This is used to derive structural and topology related features. The created Polymetric Queueing Topology Space (PQTS) data space is then used for Machine Learning (ML) driven anomaly detection and attack classification algorithms. The approach is evaluated on TON_IoT and CIC-IDS2017 benchmark data sets using tree-based ML models, particularly Random Forest and LightGBM, for binary and multiclass classification. Experimental results demonstrate strong detection performance while employing a lightweight feature set for prioritization, that is consistent across diverse IP-based networks. These results demonstrate that ML algorithms trained on PQTS data space provides an effective foundation for intrusion detection within heterogeneous network environments.**

Keywords-Network Intrusion Detection; Machine Learning; Feature Engineering; Network Security; Encrypted Network Traffic Classification.

I. Introduction

With the increasing complexity of modern computer networks and the rapid growth in both the number and sophistication of cyber-threats, ensuring network security has become a critical challenge. Traditional signature-based Intrusion Detection Systems (IDSs) remain widely deployed due to their effectiveness in detecting known attack patterns. However, an inherent reliance on predefined signatures limits their ability to identify novel and previously unseen attacks, including zero-day exploits and emerging threats, such as Large Language Model (LLM)-generated malware [1]. Consequently, anomaly-based intrusion detection has gained increasing attention as it focuses on identifying deviations from normal network behavior rather than relying on known attack signature [2]. In parallel, critical infrastructures, enterprise networks, and other specialized domains, such as the automotive industry, have become prime targets for cyber-attacks, thereby increasing the demand for reliable and timely threat detection mechanisms. The large volumes of network traffic generated in such environments require IDSs that are computationally efficient and capable of operating in near real time. In this paper, we evaluate Machine Learning (ML) algorithms based on the subset Primary Target Group (PTG) of the PQTS data space. The model-derived data space is designed to address these challenges by providing the necessary information for prioritized intrusion detection in Internet Protocol (IP)-based network environments.

The structure of the paper is as follows: Section II reviews related work. Section III describes the data sets used for evaluation. Section IV details the system architecture. Section V describes the experimental results. Finally, Section VI summarizes the findings and outlines directions for future work.

II. Related Work

Recent research on network intrusion detection in Internet of Things (IoT) environments is increasingly driven by the need to operate under strict computational constraints while analyzing predominantly encrypted network traffic. For related work, we categorized existing work into (i) approaches that emphasize resource-efficient Network Intrusion Detection System (NIDS) designs for encrypted traffic and (ii) studies that evaluate and benchmark ML algorithms for IDSs under such constraints. The following discussion reviews related work along these two categories.

A substantial body of research addresses the protection of IoT environments through lightweight detection mechanisms, often emphasizing feature reduction and computationally efficient learning algorithms for anomaly detection and traffic classification [3]–[5]. Most of these approaches rely on ML-based models to enable efficient analysis of network traffic on resource constrained devices.

Nguyen et al. propose Realguard, a lightweight deep-learning-based IDS designed for deployment on resource-constrained IoT gateways [6]. Their system combines incremental statistical feature extraction with a compact, fully connected neural network to enable packet-level attack detection at the network edge. While Realguard follows a conventional feature-based NIDS pipeline, our approach adopts a heuristic packet processing model that supports scalable analysis of encrypted IoT traffic through adaptive prioritization mechanisms.

Complementary to lightweight model designs, several studies focus on feature aggregation and temporal abstraction mechanisms. Raskovalov et al. introduce a sliding time window–based and queue-oriented NIDS that relies on flow-level aggregation and compact neural networks [7]. Their approach effectively captures temporal traffic characteristics using a predefined feature set and a queuing mechanism of flows. Our model enables adaptive and scalable analysis of encrypted traffic by selectively prioritizing two layered predefined PTGs. This includes Host, Protocol, and the Universal queues. Additionally, connection flows across the Primary Analysis Cycle (PAC) time intervals are reconstructed. The aggregation of queue-based characteristics combined with prioritization mechanisms and conventional flow-based features expands the existing research by adding multiple layers of information. This facilitates novel and resource-efficient evaluations on aggregated subsets, such as PTG $\subset$ PQTS. Beyond efficiency-oriented detection mechanisms, comparability and reproducibility have emerged as important concerns in ML-based NIDS research. Sarhan et al. address these challenges by proposing a standardized NetFlow-based feature set and benchmarking multiple open data sets using a fixed flow-level representation. While their work facilitates cross-dataset comparability, our approach extends standardized flow-level representations with additional structural and topology-aware information, enabling a more expressive yet lightweight abstraction of IP-based encrypted network traffic [8].

III. Data Set Description

To ensure comparability and validate the proposed model, two publicly available data sets were used: TON_IoT and CIC-IDS2017, which represent different application scenarios. Both data sets are standard benchmarks for the research, development, and evaluation of NIDS and broader cybersecurity analysis. Each contains raw captured network traffic complemented by subsequently labeled attacks. Structural differences and their suitability for this study are discussed in the following section.

A. TON_IoT

The TON_IoT data sets were developed to model realistic Industrial Internet of Things (IIoT) and IoT environments comprising heterogenous sensors, edge and fog systems, and hosts running different versions of Windows and Unix-based operating systems [9]–[16]. It includes network traffic protocols, such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP), as well as common services like Hypertext Transfer Protocol (HTTP), Secure Shell (SSH), Domain Name System (DNS), and File Transfer Protocol (FTP) along with sensor telemetry and operating system logs. The data set shows eight different attack categories, including scanning, Denial of Services (DoS), Distributed Denial of Services (DDoS), ransomware, backdoor, data injection, Man-in-the-Middle (MITM), and password-cracking attacks, all executed against vulnerable services to generate realistic threat behavior. The TON_IoT data set is used to evaluate host- and network-based IDS in IoT environments. Unlike traditional IDS data sets, it emphasizes realistic cyber-physical system behavior, enabling cross-layer intrusion detection.

B. CIC-IDS2017

Another popular data set was published by the Canadian Institute for Cybersecurity (CIC). Sharafaldin et al. focused on generating network traffic that reflects a generic office environment [17]. Hence, they put emphasis on realistic background traffic by emulating naturalistic benign behavior of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols, simulating working hours from Monday to Friday. The data set was generated within a fully configured network comprising a modem, firewall, switches, routers, and multiple operating systems, including Windows, Ubuntu, and macOS. Network traffic was collected from 12 machines within the victim network and supplemented with real attacks originating from the attack network, covering both internal LAN communications and Internet traffic. In addition, network traffic data, memory dumps, and system calls from the victim machines were captured during the attacks. More than 80 network flow features were extracted using CICFlowMeter [18]. They recorded 18 different attack classes including different types of Brute Force and DoS, as well as Heartbleed, Web Attacks, Infiltration, Botnet, and DDoS. This data set is widely used for network-based intrusion detection and is often used for supervised and semi-supervised ML. Its increased realism compared to earlier data sets and inclusion of modern attack types make it a standard benchmark in IDS research. With TON_IoT and CIC-IDS2017 we utilize two highly cited benchmark data sets as input for our network analysis model described in the Section IV. The model processes the captured network traffic of TON_IoT and CIC-IDS2017 as pcap format. It utilizes model structure, queue, and topology features to derive the PQTS data space.

IV. Network Analysis Model Description

In a previous study, we presented the architecture of a heuristic packet processing model for network analysis, along with an initial empirical evaluation of its effectiveness [19]. The model derived data space (*Polymetric Queueing Topology Space (PQTS)*) is presented together with tree-based ML approaches. This data space incorporates polymetric features derived from the model architecture to improve ML anomaly detection and classification. This section describes the proposed model, which is to be further evaluated based on predefined PTGs. To create the model data space, the data sets mentioned in Section III were used as input.

Figure 1 shows a simplified architecture image of the model. It presents an illustration of the reconstructed network traffic structure after a complete PAC time interval. After capture in Step 1, the individual packets are sorted into queue-based structures based on device, protocol, and universal PTGs. After the expiration of a fixed PAC time interval, which depends on the network and the communication conditions,

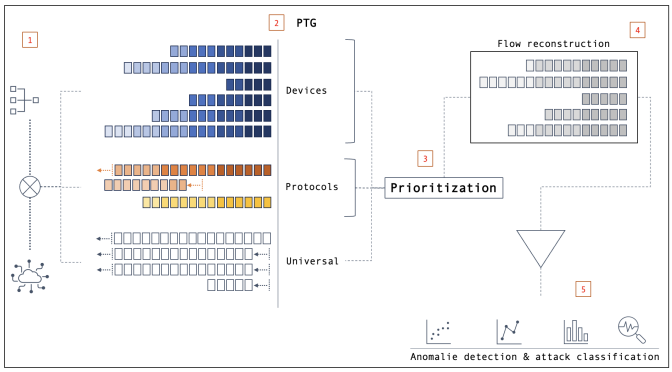

Figure 1. Example model architecture after completed PAC with implicit flow regression.

features are derived from the model structure and stored asPQTS data space. Using these features, a prioritization algorithm calculates, which queue should be marked for further deep attack classification and anomaly detection analysis. The marked queues are then selected and separately analyzed in Step 4. Traffic flows are reconstructed, creating additional features for the PQTS data space. Furthermore, the content of reconstructed flows is analyzed and represented by calculated *Time Series Flow Content Features*. In the final step, the time series feature data and the PQTS model data are merged and made available for further deep ML anomaly detection and intrusion classification methods.

In the following model evaluation, PTG features (Figure 1, designation 1) were used to train the ML models. The configuration parameters are defined with a PAC duration of 30 seconds. For each time interval a classification is performed to determine whether the observed behavior is benign or indicative of an intrusion. In case of multi-class classification, the specific type of attack is additionally identified.

The developed model demonstrator uses network traffic in pcap format as input. The network traffic is replayed and the first part of the model architecture is built as shown in Figure 1 designation 2. The following PTGs are therefore specified:

- **Devices** Including all internal devices.
- **Protocols** Including the captured transport layer protocols used.
- **Universal** Including all network packets in the exact temporal sequence in which they were observed and recorded at the network interface.

For each time interval, 17 different features are calculated. The features used for the ML-models are discussed in more detail in Section V-C.

V. Machine Learning

In this section, a description of the ML-models used for binary and multi-class classification is given. For each of the data sets described in Section III, several models utilizing Random Forest and LightGBM are applied.

A. Preprocessing

This subsection describes the preprocessing steps applied to the output of the model prior to machine learning. The system processes raw network traffic as input. Since features are extracted directly from *pcap*-files or *tshark*, no explicit handling of missing or invalid values is required at the packet level. Traffic is segmented into fixed-length windows of 30 seconds, from which statistical flow-based features are extracted and aggregated. These feature vectors form the input to the subsequent machine-learning models.

Each PAC is assigned a label according to the ground truth information provided by the respective data set. As the data sets differ in structure, data set specific labeling strategies are employed. For the TON_IoT data set, labels are derived from the directory structure provided by the authors, where traffic is organized by attack category and benign activity. For the CIC-IDS2017 data set, labels are assigned based on the attack start and end timestamps specified in the flow reconstructed *csv*-files. A window is labeled as malicious if any packet timestamp overlaps with an attack interval and benign otherwise. No overlapping attack types occur in either data set.

Prior to model training, all numerical features are normalized using z-scores. Finally, the data set is split into training and validation (70/30) subsets, which are used for model selection, hyperparameter tuning, and performance evaluation, respectively.

B. Model Selection

Tree-based learning algorithms are particularly suitable in this context because they can capture non-linear relationships and handle heterogeneous feature distributions, which are typical for network-traffic, while remaining computationally efficient and requiring minimal preprocessing [20]. Random Forest and gradient-boosted decision tree models have been widely adopted in intrusion detection due to their robustness on imbalanced data sets, low inference latency, and limited preprocessing requirements [21]. In addition, their inherent support for feature importance estimation enables transparent analysis of how PQTS derived features contribute to detection performance. Based on these considerations, Random Forest and LightGBM are selected as representative tree-based models for experimental evaluation. Random Forest serves as a robust ensemble baseline with strong generalization properties, while LightGBM represents a gradient-boosted variant optimized for efficiency and scalability. Evaluating both models on the same PQTS feature representation enables a comparative analysis of different tree-based learning paradigms that are not influenced by fundamentally different model classes.

C. Feature Selection

Feature selection is guided by the structure of the PTG subset of PQTS abstraction rather than by explicit feature elimination. All experiments use a fixed set of 17 queue-derived features capturing variability, growth dynamics, and packet processing behavior (Table I), ensuring that observed

performance differences originate from the learning models rather than feature availability.

TABLE I. PQTS FEATURES USED FOR ML TRAINING.

Index	Feature Name
1	gq_mean_protocol_queue_length
2	gq_protocol_queue_length_variance
3	gq_protocol_queue_length_entropy
4	gq_median_protocol_queue_length
5	gq_mean_host_queue_length
6	gq_median_host_queue_length
7	gq_host_queue_length_variance
8	gq_host_queue_length_entropy
9	gq_mean_queue_length
10	gq_median_queue_length
11	gq_popped_pkts_iteration
12	gq_popped_pkts_cumulative
13	gq_protocol_queue_count
14	gq_protocol_queue_count_layer_4_below
15	gq_protocol_queue_count_diff
16	gq_mean_growth_rate
17	gq_median_growth_rate

Feature importance analysis from Random Forest (Table II) shows a consistent ranking across data sets and classification tasks. With respect to the particular PTG queue classification type, entropy- and variance-based queue-features contribute most strongly to detection performance.

TABLE II. BEST HYPERPARAMETERS ON PQTS FOR BINARY AND MULTICLASS CLASSIFICATION.

Set	Model	Best Features	Scores
		Binary Classification	
TON	RF	gq_host_queue_length_entropy	0.2814
		gq_protocol_queue_length_variance	0.1198
		gq_median_host_queue_length	0.1170
CIC	RF	gq_popped_pkts_cumulative	0.2699
		gq_protocol_queue_length_variance	0.0846
		gq_host_queue_length_entropy	0.0844
		Multiclass Classification	
TON	RF	gq_host_queue_length_entropy	0.2352
		gq_median_host_queue_length	0.1133
		gq_protocol_queue_length_variance	0.0869
TON	LGBM	gq_host_queue_length_entropy	0.2779
		gq_popped_pkts_cumulative	0.2330
		gq_mean_growth_rate	0.2259
CIC	RF	gq_popped_pkts_cumulative	0.2970
		gq_host_queue_length_entropy	0.0740
		gq_protocol_queue_length_variance	0.0716
CIC	LGBM	gq_popped_pkts_cumulative	0.7153
		gq_host_queue_length_entropy	0.1406
		gq_protocol_queue_length_variance	0.1045

Overall, the results demonstrate that a compact, model derived PTG feature set is sufficient for effective intrusion detection across data sets without data set specific feature engineering.

D. Model Training and Evaluation

All models are trained on the PQTS subset PTG shown in Table I. For TON_IoT and CIC-IDS2017, we use the same data set split for both binary and multiclass experiments, with data set statistics reported in Table III.

TABLE III. DATA SET OVERVIEW FOR PTG-BASED EXPERIMENTS.

Data Set	Classes	Features	Train	Test
TON_IoT	5	17	11,749	2,938
CIC-IDS2017	18	17	3,873	969

Hyperparameter optimization is performed using grid search with 5-fold cross-validation for all evaluated models. The search spaces applied for the multiclass and binary classification of the corresponding algorithms are shown in IV and V. For Random Forest classifiers, optimization focuses on ensemble size, tree depth, and split criteria, while LightGBM tuning additionally considers learning rate and framework related optimization parameters.

TABLE IV. RANDOM FOREST PARAMETER GRID FOR GRIDSEARCHCV.

Hyperparameter	RF
n_estimators	[100, 200]
max_depth	[10, 20, 30, None]
min_samples_split	[2, 5]
min_samples_leaf	[1, 2]
max_features	[sqrt, log2]
bootstrap	[True]

TABLE V. LIGHTGBM PARAMETER GRID FOR GRIDSEARCHCV.

Hyperparameter	LGBM
n_estimators	[100, 200]
max_depth	[10, 20, -1]
learning_rate	[0.05, 0.1]
num_leaves	[31, 63]
min_child_samples	[20, 50]
subsample	[0.8]
colsample_bytree	[0.8]

All reported classification results are obtained using the selected hyperparameter configurations identified during this optimization process and summarized in Tables VI and VII.

Tables VIII and IX summarize the classification performance of the evaluated models on the PTG feature representation across binary and multiclass intrusion detection tasks. Overall, the results demonstrate that tree-based models can effectively adapt the PTG abstraction, achieving high detection performance even on encrypted network traffic.

For binary classification on the TON_IoT data set, the Random Forest model achieves an F1-score of 0.9947 and

TABLE VI. BEST RANDOM FOREST HYPERPARAMETERS ON PTG FOR BINARY AND MULTICLASS CLASSIFICATION.

Feature	TON		CIC	
	Bin	Mul	Bin	Mul
n_estimators	200	100	150	100
max_depth	10	20	None	20
min_samples_split	5	2	5	5
min_samples_leaf	1	1	2	1
max_features	sqrt	sqrt	sqrt	sqrt
bootstrap	True	True	True	True

TABLE VII. BEST LIGHTGBM HYPERPARAMETERS ON PTG FOR BINARY AND MULTICLASS CLASSIFICATION.

Feature	TON	CIC
n_estimators	100	100
max_depth	20	-1
learning_rate	0.1	0.05
num_leaves	63	63
min_child_samples	50	50
subsample	0.8	0.8
colsample_bytree	0.8	0.8

TABLE VIII. BINARY CLASSIFICATION RESULTS OF PTG.

Model	Set	Acc	F1	AUC
RF	TON	0.9966	0.9947	0.9999
RF	CIC	0.8534	0.5439	0.8979

an Area under the ROC curve (AUC) of 0.9999. The balanced accuracy exceed 99%, indicating robust detection capability under moderately imbalanced class distributions. This highlights the suitability of the PTG features for distinguishing benign and malicious traffic without deep packet inspection. In contrast, binary classification on CIC-IDS2017 yields noticeably lower performance, particularly in terms of recall and F1-score. While the model maintains a reasonable AUC of 0.8979, the reduced balanced accuracy reflects the higher class imbalance and increased attack diversity in terms of volume-based characteristics of the data set. These results suggest that the performance degradation is primarily data set driven rather than caused by limitations of the PTG feature representation.

TABLE IX. MULTICLASS CLASSIFICATION RESULTS OF PTG.

Model	Set	BalAcc	$F1_w$	MeanAUC
RF	TON	0.9915	0.9915	0.9998
LGBM	TON	0.9935	0.9936	0.9998
RF	CIC	0.8627	0.8376	0.9753
LGBM	CIC	0.6549	0.8612	0.9866

For multiclass classification on TON_IoT, both Random Forest and LightGBM achieve strong results. The Random Forest model attains a weighted F1-score of 0.9915, while LightGBM slightly improves upon this with a weighted F1-score of 0.9936. The consistently high MeanAUC values across both models indicate stable class separation and balanced performance across attack categories. Multiclass classification on CIC-IDS2017 remains challenging. Nevertheless, the Random Forest model achieves an approximate weighted F1-score of 0.8376, demonstrating that the PQTS-based abstraction retains discriminative power even under more complex multiclass conditions. Overall, the results indicate that the PTG feature space enables intrusion detection across multiple datasets and learning models, particularly for volume-based attacks, such as DoS, while maintaining low computational cost and leveraging novel features derived from queue-structure statistics.

E. Performance of Comparable Approaches

To put these results into perspective, a comparison with other IDSs, introduced in Section II, is provided. It is important to note that the results obtained on a small subset of PQTS are competitive across both datasets. For the TON dataset, the proposed model achieves performance comparable to the related work; however, on CIC it performs less strongly for this reduced PQTS subset. The performance metrics reported in the literature for binary and multiclass classification are summarized in Tables X and XI.

TABLE X. COMPARISON OF BINARY CLASSIFICATION RESULTS

Model	Set	Acc	F1	Features
CL-SKD [5]	CIC	0.9980	0.9980	-
Realguard [6]	CIC	0.9964	-	100
NetFlow [8]	TON	0.9964	1.00	43

TABLE XI. COMPARISON OF MULTICLASS CLASSIFICATION RESULTS

Model	Set	BalAcc	$F1_w$	Features
BT-TPF [4]	CIC	0.9960	0.9960	78
CL-SKD [5]	CIC	0.9984	0.9984	-
Realguard [6]	CIC	0.9993	-	100
BT-TPF [4]	TON	0.9945	0.9944	43
NetFlow [8]	TON	0.9805	0.98	43

Overall, the comparison suggests that the proposed approach is already competitive with established IDSs, particularly on TON, while the weaker results on CIC for this reduced PQTS subset indicate clear potential for improvement for certain attack classes in subsequent full-space evaluations.

VI. CONCLUSION AND FUTURE WORK

This paper presents a multi-model evaluation of a heuristic network analysis model. It abstracts captured packet streams into a complex, structurally motivated, and metadata-driven data space. The PQTS derives from polymetric queue states, model architecture and traffic prioritization mechanisms. The proposed approach was evaluated using the PTG feature subset of the PQTS representation on two widely used and well-established benchmark datasets. This focus allows an assessment of the model's robustness and applicability under realistic and comparable experimental conditions, while addressing key challenges of modern network environments, including high traffic volumes, encryption, and resource constrained environments.

The experimental results demonstrate that a single subset of the PQTS data space enables competitive intrusion detection for volume based attacks across different data sets and classification tasks. Tree-based learning models, such as Random Forest and LightGBM, achieved strong performance on both binary and multiclass scenarios, particularly on the TON_IoT data set. Feature importance further confirm that dynamic queue-related characteristics, such as entropy, variance, and growth behavior, are highly discriminative for both data sets. These findings underline that meaningful security relevant information can be extracted from network traffic without reliance on deep packet inspection or extensive feature engineering.

Before deployment in production environments, targeted validation steps are necessary. These include validation of robustness and generalization capabilities, assessment under realistic traffic loads, and evaluation against previously unseen attack patterns. In summary, the results presented indicate that the PTG subset of the PQTS data space already constitutes a promising foundation for anomaly-based intrusion detection in encrypted and resource-constrained network environments.

REFERENCES

[1] K. Ahi and S. Valizadeh, "Large language models (LLMs) and generative AI in cybersecurity and privacy: A survey of dual-use risks, AI-generated malware, explainability, and defensive strategies," in *2025 Silicon Valley Cybersecurity Conference (SVCC)*, 2025, pp. 1–8. DOI: 10.1109/SVCC65277.2025.11133642.

[2] J. Graf, K. Neubauer, S. Fischer, and R. Hackenberg, "Architecture of an intelligent intrusion detection system for smart home," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6. DOI: 10.1109/PerComWorkshops48775.2020.9156168.

[3] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the internet of things," *IEEE Access*, vol. 7, pp. 42 450–42 471, 2019. DOI: 10.1109/ACCESS.2019.2907965.

[4] Z. Wang *et al.*, "A lightweight IoT intrusion detection model based on improved BERT-of-Theseus," *Expert Systems with Applications*, vol. 238, p. 122 045, 2024. DOI: 10.1016/j.eswa.2023.122045.

[5] Z. Li and W. Yao, "A two stage lightweight approach for intrusion detection in internet of things," *Expert Systems with Applications*, vol. 257, p. 124 965, 2024. DOI: 10.1016/j.eswa.2024.124965.

[6] X.-H. Nguyen, X.-D. Nguyen, H.-H. Huynh, and K.-H. Le, "Realguard: A lightweight network intrusion detection system for IoT gateways," *Sensors*, vol. 22, no. 2, 2022. DOI: 10.3390/s22020432.

[7] A. Raskovalov, N. Gabdullin, and I. Androsov, *NIDS neural networks using sliding time window data processing with trainable activations and its generalization capability*, 2024. DOI: 10.48550/ARXIV.2410.18658.

[8] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 357–370, 2022. DOI: 10.1007/s11036-021-01843-0.

[9] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustainable Cities and Society*, vol. 72, p. 102 994, 2021. DOI: 10.1016/j.scs.2021.102994.

[10] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_iot telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, 2020. DOI: 10.1109/ACCESS.2020.3022862.

[11] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. D. Hartog, "ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, 2022. DOI: 10.1109/JIOT.2021.3085194.

[12] N. Moustafa, M. Keshky, E. Debiez, and H. Janicke, "Federated TON_IoT windows datasets for evaluating AI-based security applications," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou, China, 2020, pp. 848–855. DOI: 10.1109/TrustCom50675.2020.00114.

[13] N. Moustafa, M. Ahmed, and S. Ahmed, "Data analytics-enabled intrusion detection: Evaluations of ToN_IoT linux datasets," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 727–735. DOI: 10.1109/TrustCom50675.2020.00100.

[14] N. Moustafa, *New generations of internet of things datasets for cybersecurity applications based machine learning: TON_IoT datasets*, 2019. DOI: 10.26190/5D7AC9BFE8487.

[15] N. Moustafa, *A systemic IoT-fog-cloud architecture for big-data analytics and cyber security systems: A review of fog computing*, 2019. DOI: 10.48550/ARXIV.1906.01055.

[16] J. Ashraf *et al.*, "IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities," *Sustainable Cities and Society*, vol. 72, 2021. DOI: 10.1016/j.scs.2021.103041.

[17] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization:" in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116. DOI: 10.5220/0006639801080116.

[18] A. Habibi Lashkari, G. Draper Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, 2017, pp. 253–262. DOI: 10.5220/0006105602530262.

[19] J. Graf, M. Hachani, S. Fischer, and R. Hackenberg, "A heuristic packet processing model for improved encrypted network analysis," in *Proceedings of the 2025 Cyber Security in CarS Workshop*, 2025, pp. 1–12. DOI: 10.1145/3736130.3764510.

[20] K. S. Adewole, A. Jacobsson, and P. Davidsson, "Intrusion detection framework for internet of things with rule induction for model explanation," *Sensors*, vol. 25, no. 6, 2025. DOI: 10.3390/s25061845.

[21] O. Achbarou, T. Datsi, O. Bourkoukou, and A. M. El Kiram, "Enhanced intrusion detection system using feature selection and hybrid learning models for high performance and efficiency in an IoT environment," *Journal of Engineering Research*, 2025. DOI: 10.1016/j.jer.2025.10.016.