SHAP happens: an Explainable IDS for Industrial IoT Networks

Pierangelo Loi
Università degli Studi di Cagliari
Cagliari, Italy
pierangelo.loi@unica.it

Daniele Canavese
IRIT, CNRS
Toulouse, France
daniele.canavese@irit.fr

Leonardo Regano
Università degli Studi di Cagliari
Cagliari, Italy
leonardo.regano@unica.it

Davide Maiorca
Università degli Studi di Cagliari
Cagliari, Italy
davide.maiorca@unica.it

Giorgio Giacinto
Università degli Studi di Cagliari
Cagliari, Italy
CINI - Consorzio Interuniversitario Nazionale per l'Informatica
Italy
giorgio.giacinto@unica.it

***Abstract*—Industrial Internet of Things (IIoT) technologies have been increasingly leveraged across various industry sectors, due to their benefits in terms of automation, monitoring, and operational efficiency. However, the increased connectivity and heterogeneity of IIoT devices have also broadened the attack surface, making these systems attractive targets for cyber threats. In this context, machine learning–based Intrusion Detection Systems (IDS) have emerged as promising solutions due to their ability to detect complex patterns in network traffic without relying on static rules or deep packet inspection. A key limitation of such systems, however, lies in their lack of interpretability, posing challenges for adoption in safety-critical industrial settings.**

In this work, we propose an explainable IDS that leverages a Random Forest classifier for accurate traffic classification and integrates SHAP (SHapley Additive Explanations) to provide transparent explanations of model decisions. We evaluate our system using the CIC IoT-DIAD 2024 dataset, which includes a broad spectrum of network attacks. Our approach demonstrates good detection performance while also delivering intuitive explanations for each prediction. By analyzing the specific network features, such as inter-arrival times and packet sizes, that most influence each alert, security analysts may better assess, validate, and act upon IDS outputs.

***Index Terms*—Internet of Things, Intrusion detection, Explainable AI**

I. Introduction

Industrial Internet of Things (IIoT) technologies have seen rapid adoption in many industrial sectors, including manufacturing [5], agriculture [11], and transportation [7]. Indeed, businesses are implementing IIoT-based solutions in industrial processes to enhance operational efficiency, enable real-time monitoring and decision-making, reduce downtime through predictive maintenance, and optimize resource utilization across complex and distributed systems [4].

However, adopting such solutions comes at a cost: IoT devices are well-known to be insecure. For example, IoT devices' firmwares typically exhibit multiple vulnerabilities [32], which attackers may leverage to hijack such devices or endanger the confidentiality of user data. Furthermore, many devices still use default credentials or weak authentication schemes, making them easy prey for attackers [3]. Thus, considering the vulnerable state of many IIoT endpoints and their deployment in critical processes, securing IIoT-enabled industrial operations has become paramount. The SANS Institute, in a recent whitepaper[1], highlighted network-based attack detection among the five most critical security controls to be implemented in IIoT scenarios.

In this context, Intrusion Detection System (IDS) based on Machine Learning (ML) models have emerged as a promising solution for network attack detection in IIoT. Rather than relying on hand-crafted signatures or full packet payload inspection, an ML-based IDS can learn to recognize patterns of normal vs. malicious traffic using features of network flows or protocol behaviors. ML and Deep Learning (DL) techniques can analyze large volumes of network data, detect complex attack patterns, and even adapt to new threats faster than static rule-based systems [37].

A significant challenge with ML-driven IDS is their lack of interpretability [38], as advanced models often function as "black boxes." While these models may have high detection accuracy, their decision-making process is unclear, undermining trust and hampering incident response. Security professionals require insights into why an IDS raises alarms to differentiate between actual threats and false positives effectively. Emerging Explainable Artificial Intelligence (XAI) techniques, which aim to provide human-interpretable explanations for ML model decisions, may be leveraged to empower ML-based IDS with such needed transparency. Indeed, methods like SHAP (SHapley Additive Explanations) and Local Interpretable Model-agnostic Explanations (LIME) can be applied

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

[1]https://www.sans.org/white-papers/five-ics-cybersecurity-critical-controls/

in network IDS to clarify why specific traffic is flagged as malicious. Additionally, XAI facilitates a feedback loop that allows developers to refine and debug models by identifying spurious patterns or biases [12].

In this paper, we present an ML-based IDS for IIoT networks that addresses the above challenges by combining an ensemble detection approach with explainable AI. Specifically, we employ a Random Forest (RF) classifier to identify malicious network activities. To make the IDS decisions transparent to users, we integrate the SHAP explainability technique, which produces clear feature-importance explanations for each detection. Such interpretability allows security practitioners to trust and verify the system's outputs, and aids in troubleshooting misclassifications by revealing the model's reasoning. Our results show that the proposed IDS can achieve good detection performance while outputting meaningful explanations for its decisions, facilitating its adoption in critical industrial contexts where both security and interpretability are paramount.

The remainder of this paper is organized as follows. Section II reviews related work in intrusion detection and explainable AI. Section III briefly introduces RF and SHAP, while Section IV describes the employed dataset and the attacks considered. Section V presents the experimental setup and results, and Section VI concludes the paper.

II. Related Works

Early IDS in both IT and OT environments relied on expert-defined rules and simple statistical models to identify anomalies in network traffic. Denning et al. [10] developed a model that formalized anomaly detection through statistical profiles of system metrics, and signature-based tools like Roesch's Snort [31] matched packet headers and payloads to known attack patterns. These classical approaches monitored low-level features, source/destination IPs and ports, protocol flags, and basic packet statistics, to flag deviations from an established baseline. For example, in Giacinto et al. [14], features containing information about the payload, e.g., errors reported by the operating system, root access attempts, are used. While effective when port mappings were static and most traffic unencrypted, such heuristics have become increasingly brittle. Modern networks use dynamic port allocations and protocol tunneling, undermining port-based detection. Pervasive encryption, like TLS, also renders payload-based signatures impractical [9], [23].

To overcome these limitations, the field shifted toward ML classifiers. Supervised algorithms, support vector machines, decision trees, and particularly ensemble methods, learn complex decision boundaries from labeled data, improving detection of novel attacks compared to static rules [20]. RF are an ensemble method that aggregates hundreds of randomized trees, reducing overfitting and handling high-dimensional, noisy feature spaces typical of IoT/OT traffic [8]. Several works have implemented this approach. For example, Farnaaz et al. [13] proposed an RF-based modelling approach for IDS, while Jabbar et al. [19] proposed a combination of RF and the Average One-Dependence Estimator (AODE) to create a more robust classifier. Resende and Drummond [29] presented a comprehensive survey of the application of RF models to IDS. They concluded that, at the time of publication, despite unexplored problems, there were significant advantages in implementing such a technique.

Latest studies show that RF nowadays is still a viable option. Ali et al. [6] perform a comprehensive comparison of DL models (Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM)) and classical ML algorithms (logistic regression, naïve Bayes, RF, k-NN, decision tree) on the CICIDS2017 dataset, enhanced with SMOTE and correlation-based feature selection, and show that RF achieves the highest detection accuracy (99.9%) while maintaining lower computational complexity.

DL automates feature extraction by learning hierarchical, latent representations of traffic data. Autoencoders, for instance, detect anomalies by measuring reconstruction error, as demonstrated in the Kitsune NIDS [24], which leverages an ensemble of autoencoders to flag zero-day attacks without manual feature engineering. Recurrent neural networks and convolutional architectures have also been applied to model temporal and spatial dependencies in traffic streams, yielding improved detection of subtle or multi-stage intrusions [25]. Racherla et al. [28] implement a lightweight LSTM network in their so-called "Deep-IDS " framework, deployed on IoT edge nodes for real-time detection and classification of Denial of Service (DoS), Distributed Denial of Service (DDoS), brute-force, Man-in-the-Middle, and replay attacks, obtaining high detection rates.

In general, the advent of DL has brought many advantages but has also introduced some issues. Tsimenidis et al. [35] provide a comprehensive survey of DL techniques for IoT intrusion detection and conclude that DL models' practical deployment requires overcoming challenges related to real-time processing on resource-constrained devices, the need for continual online learning to handle concept drift, and the development of representative, high-quality IoT datasets for robust validation. DL models also introduce opacity: without interpretability, security analysts cannot easily understand why a flow was classified as malicious. This gap has spurred interest in XAI techniques. SHapley Additive exPlanations (SHAP) provide per-instance feature attributions based on Shapley values from cooperative game theory, fairly distributing the "payout" (model prediction) among input features [22]. Tree-SHAP algorithms enable efficient, exact computation of Shapley values for RF, generating positive or negative contributions for each feature that sum to the model's output [22]. In the IDS domain, SHAP can reveal which traffic characteristics, such as atypical packet-size spikes or rare flag combinations, drove a malicious verdict, enhancing analysts' trust and aiding in the diagnosis of false positives [33], [39]. Visualizations like force plots or summary beeswarm charts make these explanations accessible and actionable during incident response.

III. Background

To support the design of our explainable IDS for IoT and Operational Technology (OT) networks, we briefly review two core techniques: RF classifiers for high-performance anomaly detection on tabular network traffic, and SHapley Additive exPlanations (SHAP) for interpreting their outputs. In IoT/OT environments, where device heterogeneity and safety-critical operations prevail, an IDS must not only detect attacks accurately but also provide clear, actionable explanations to operators [20], [36]. The following sections summarize the general mechanisms of these methods and then discuss their specific applicability to security monitoring.

A. Random forests

RF are an ensemble learning method that constructs a multitude of decision trees and aggregates their outputs to improve predictive performance and robustness [8]. A single decision tree partitions the feature space by recursively splitting on feature values, yielding an intuitive set of if–then rules that map inputs to class labels. However, individual trees can overfit noisy data and exhibit high variance. RF overcomes these issues through two key mechanisms:

- Bootstrap Aggregation (Bagging): Each tree is trained on a random sample (with replacement) of the training data;
- Random Feature Selection: A random subset of features is considered at each split, further decorrelating the trees.

During inference, each tree casts a vote for the predicted class, and the forest's output is determined by majority vote (classification) or average prediction (regression). This ensemble strategy reduces variance and mitigates overfitting, resulting in models that generalize well to unseen data [8]. RFs are particularly well-suited for intrusion detection in IoT/OT contexts for several reasons: they handle high-dimensional network traffic data with many potentially noisy or irrelevant features, they are relatively fast to train and evaluate (enabling near real-time operation), and they inherently provide measures of global feature importance that help identify which traffic attributes (e.g., flow duration, packet counts, protocol flags) most influence detection decisions [19]. Empirical studies show that RF-based IDS achieve high accuracy and low false-alarm rates on public IoT/OT datasets, often matching or exceeding the performance of more complex DL models while requiring less parameter tuning [25].

B. SHAP

Despite the transparency of decision trees, the aggregation of many trees in an RF can obscure the rationale for individual predictions. SHapley Additive exPlanations (SHAP) addresses this by attributing the prediction of any black-box model to its input features based on Shapley values from cooperative game theory [22]. In this framework, each feature is treated as a "player" in a game where the model's output is the "payout." The Shapley value for a feature quantifies its average marginal contribution to the prediction across all possible feature subsets, ensuring a fair and consistent attribution.

SHAP produces a vector of feature attributions for each instance, where positive values indicate features that push the prediction toward the positive class (e.g., "malicious") and negative values indicate features that push it toward the negative class (e.g., "benign"). Crucially, TreeSHAP exploits the tree structure of RFs to compute exact Shapley values in polynomial time, rendering SHAP explanations efficient even for large ensembles [22].

In intrusion detection, SHAP explanations allow security analysts to see precisely which traffic characteristics (such as an unusual spike in flow byte count or the presence of rare protocol flags) led the RF to flag a flow as malicious. This per-alert insight not only enhances operator trust but also aids in diagnosing false positives and refining detection logic [16]. A typical visualization (e.g., SHAP force plot) can illustrate how each feature's contribution combines to yield the final RF prediction. By integrating SHAP with RF-based IDS, one can achieve both high detection performance and human-interpretable explanations, a combination essential for effective, accountable security monitoring in IoT/OT deployments.

IV. Dataset

For our experiments, we decided to use the CIC IoT-DIAD 2024 dataset[2], which was designed for both IoT device identification and anomaly detection. To create it, at the Canadian Institute for Cybersecurity, Rabbani et al. [27] conducted 33 distinct attacks using malicious IoT devices targeting other IoT devices, within an IoT topology comprising 105 nodes. The dataset provides PCAP files, which contain actual traffic packets, and CSV files containing the extracted features from the traffic, using CICFlowMeter[3]. For the training, we used the data in the CSV containing the already extracted flow-based features.

A. Dataset Content

Following, we briefly explain the attacks we considered in our study, subdividing them in terms of the ISO/OSI networking layer of the protocol employed to perform such attacks.

1) Network layer attacks:

- *DDoS ICMP flood* - The attacker overwhelms the target's network by sending a massive stream of ICMP echo requests (ping packets), consuming bandwidth and forcing the victim to expend resources replying to the requests or dropping them [18], [26], [27];
- *DDoS ICMP fragmentation* - The adversary sends a continuous flood of oversized ICMP packets that must be reassembled by the target, exhausting memory and CPU cycles in the IP defragmentation process [15], [27];
- *Mirai Greeth flood* - Compromised devices from the Mirai botnet emit a high rate of GREETH (Generic Routing Encapsulation over Ethernet) probe packets to

[2]https://www.unb.ca/cic/datasets/iot-diad-2024.html

[3]https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter

saturate the victim's link and evade simple packet-filter defenses [1], [27].

2) Transport layer attacks:

- *DDoS ACK fragmentation* - Similarly to the DDoS ICMP fragmentation, a distributed set of bots sends TCP ACK segments that are deliberately split across multiple IP fragments, forcing the target to waste resources defragmenting and validating each fragment [21], [27];
- *DoS SYN flood* - The attacker opens many half-open TCP connections by sending a stream of SYN requests without completing the three-way handshake, filling the server's connection table and preventing legitimate clients from connecting [17], [27];
- *DoS UDP flood* - A continuous flow of UDP packets floods the victim's network and forces the host to repeatedly check for listening services or send ICMP "port unreachable" replies [2], [27].

3) Application layer attacks:

- *DDoS HTTP flood* - A network of bots issues a very large number of seemingly valid HTTP GET or POST requests to web servers, tying up application-layer resources (threads, database connections) [26], [27], [34];
- *DoS HTTP flood* - Continuous requests for heavy or dynamic web content (e.g., large file downloads, complex API calls) to exhaust server and backend resources, causing legitimate requests to time out [26], [27], [34].

B. Supports

The dataset comprises 27107157 packet flows, specifically labeled as shown in Table I, each of which makes the model classification units. The used flow-based features contained statistics such as flow duration, packet length mean, packet length variance, FIN, SYN, ACK flag counts, down/up ratio, average packet size, and so on. We did not use source/destination IPs/ports due to their dynamic and volatile nature.

LEVEL	CLASS	FLOWS
network	DDoS ICMP flood	201728
	DDoS ICMP fragmentation	320596
	Mirai Greeth flood	174518
transport	DDoS ACK fragmentation	2449972
	DoS SYN flood	16287923
	DoS UDP flood	5206402
application	DDoS HTTP flood	504597
	DoS HTTP flood	1563223
-	benign	398198

TABLE I
FLOW COUNTS FOR EACH CLASS.

V. EXPERIMENTAL RESULTS

In this section, we present a brief description of how we trained and tested our IDS, along with a discussion on the explainability of its decisions.

A. Training

In our experiments we used scikit-learn 1.6.1[4], Python 3.13.3[5], and optuna 4.3.0[6].

To facilitate the model training and evaluation, the dataset was divided into two subsets. Specifically, 80% of the samples were allocated to the training set, while the remaining 20% were reserved for the test set. This split resulted in a total of 21,685,725 flows in the training set and 5,421,432 flows in the test set. This partitioning strategy ensures that the model has sufficient data to learn from while also providing a representative and unbiased portion for evaluating its generalization performance.

We trained a RF classifier and performed hyperparameter optimization with the goal of maximizing the Matthews Correlation Coefficient (MCC), a robust metric particularly suitable for imbalanced classification tasks. To efficiently explore the hyperparameter space, we employed the Optuna optimization framework. The optimization process was run for 32 iterations, during which Optuna systematically searched for the combination of hyperparameters that yielded the highest MCC on the validation set. This approach allowed us to fine-tune the model for improved performance and generalization.

We optimized the following hyperparameters[7]:

- `n_estimators` - Number of trees in the forest;
- `max_depth` - Maximum depth of the tree;
- `min_samples_split` - Minimum number of samples required to be at a leaf node.

The best hyperparameters found by Optuna were `n_estimators` = 144, `max_depth` = 20, and `min_samples_split` = 2.

B. Testing

Table II reports several metrics of our classifier. While exhibiting good accuracy, several misclassifications can be noticed in the confusion matrix reported in Table III. However, an attack is frequently misclassified as another similar attack. For example, the most common misclassifications are caused by DDoS HTTP Flood attacks classified as DoS HTTP Flood attacks (19611 flows) and viceversa (10950 flows). While different in the context of execution (i.e., with a single or multiple coordinated attackers), these attacks are substantially similar in terms of performed HTTP requests, thus exhibiting similar network flow-based features.

Concerning the explainability of the decisions taken by our model, we discuss the SHAP waterfall plots generated for three of the attacks included in the dataset we employed. These plots visually present how features cumulatively influence the model's decision for each detection, starting from the expected value (the average model output over the training data) and adding/subtracting feature contributions to reach the final prediction.

[4]https://github.com/scikit-learn/scikit-learn

[5]https://www.python.org/downloads/release/python-3133/

[6]https://optuna.readthedocs.io/en/stable/index.html

[7]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

METRIC	VALUE
MCC	0.929
accuracy	95.770
balanced accuracy	77.628
(macro) F1-score	73.540
(macro) precision	70.797
(macro) recall	77.628

TABLE II
CLASSIFICATION METRICS ON THE TEST SET.

Looking at Figure 1, the most influential feature characterizing DDoS ICMP flood attacks is the average Inter-Arrival Time (IAT), which shows very low values, and a high number of packets per second. This is expected, as this attack typically involves a high number of packets arriving in quick succession.

Average IAT remains a dominant feature also in the case of the DDoS ICMP Fragmentation attack, as reported in Figure 2. Additionally, a noticeably low maximum packet length significantly influences the decision. This reflects the nature of fragmentation attacks, where the adversary sends numerous small, fragmented packets that collectively overload the packet reassembly logic of the target system.

As a last example, we consider the Mirai GREETH flood attack, reported in Figure 3. The most impacting features are all related to the rather small packet size in both flow directions. This is consistent with the typical small size of the GRE probe packets, used in this attack by Mirai-infected devices to flood the target victim endpoint.

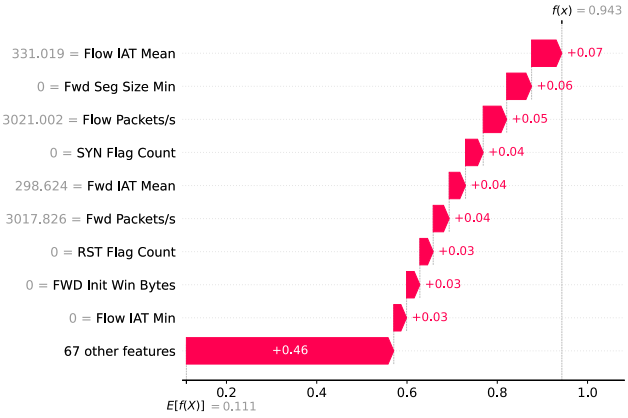

Fig. 1. DDoS ICMP flood waterfall plot.

VI. CONCLUSIONS

In this study, we introduced an explainable ML-based IDS suitable for Industrial Internet of Things (IIoT) environments. By integrating a RF classifier with SHAP explanations, our system achieves good detection performance while ensuring transparency in its decision-making process. Experimental results showed the effectiveness of the classifier across various types of attacks, while SHAP visualizations offered clear insights into the contributions of different features. In future research, we plan to employ adaptive learning techniques to address evolving threats [30]. Furthermore, we plan to leverage a digital twin of the monitored network in order to make the underlying ML learn the normal behaviour of the protected endpoints, consequently adapting the IDS decisions for the specific safeguarded network.

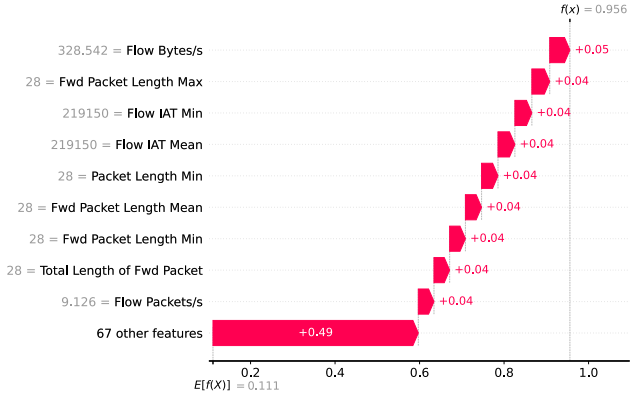

Fig. 2. DDoS ICMP fragmentation waterfall plot.

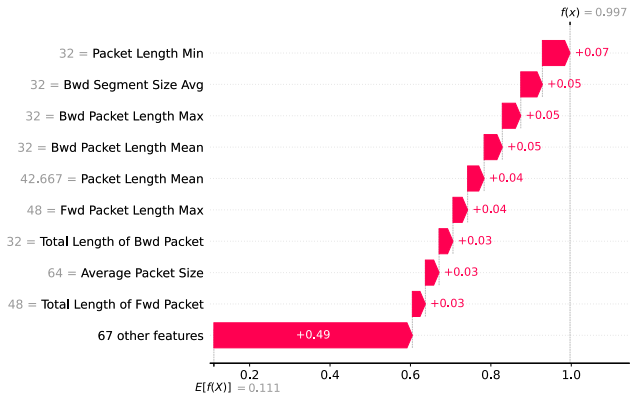

Fig. 3. Mirai Greeth flood waterfall plot.

REFERENCES

[1] Abbas, S.G., Hashmat, F., Shah, G.A., Zafar, K.: Generic signature development for iot botnet families. Forensic Science International: Digital Investigation **38**, 301224 (2021)

[2] Acharya, A.A., Arpitha, K., Kumar, B., et al.: An intrusion detection system against udp flood attack and ping of death attack (ddos) in manet. International Journal of Engineering and Technology (IJET) **8**(2) (2016)

[3] Ahmed, S., Khan, M.: Securing the internet of things (iot): A comprehensive study on the intersection of cybersecurity, privacy, and connectivity in the iot ecosystem. AI, IoT and the Fourth Industrial Revolution Review **13**(9), 1–17 (2023)

[4] Ahmed, S.F., Alam, M.S.B., Hoque, M., Lameesa, A., Afrin, S., Farah, T., Kabir, M., Shafiullah, G., Muyeen, S.: Industrial internet of things enabled technologies, challenges, and future directions. Computers and Electrical Engineering **110**, 108847 (2023). https://doi.org/https://doi.org/10.1016/j.compeleceng.2023.108847

[5] Ahmetoglu, S., Che Cob, Z., Ali, N.: Internet of things adoption in the manufacturing sector: A conceptual model from a multi-theoretical perspective. Applied Sciences **13**(6) (2023). https://doi.org/10.3390/app13063856

[6] Ali, M.L., Thakur, K., Schmeelk, S., Debello, J., Dragos, D.: Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study. Applied Sciences **15**(4), 1903 (2025)

TABLE III
CONFUSION MATRIX

		Predicted								
		benign	**ddos_ack_frag**	**ddos_http_fld**	**ddos_icmp_fld**	**ddos_icmp_frag**	**dos_http_fld**	**dos_syn_fld**	**dos_udp_fld**	**mirai_greeth_fld**
Actual	**benign**	65669	86	483	2535	7017	550	191	559	2550
	ddos_ack_frag	2534	469907	355	3879	6758	543	263	579	5176
	ddos_http_fld	345	37	87420	723	708	10950	168	244	324
	ddos_icmp_fld	4675	291	649	19418	6820	913	654	1762	5164
	ddos_icmp_frag	9415	503	861	8061	33955	1161	573	1472	8118
	dos_http_fld	2487	67	19611	2131	2000	284001	418	1202	728
	dos_syn_fld	6158	180	792	11338	7246	2040	3220813	5414	3604
	dos_udp_fld	8086	288	535	16015	10542	2560	1504	994374	7376
	mirai_greeth_fld	3874	407	423	5257	6508	544	349	1004	16538

[7] Bhargava, A., Bhargava, D., Kumar, P.N., Sajja, G.S., Ray, S.: Industrial iot and ai implementation in vehicular logistics and supply chain management for vehicle mediated transportation systems. International Journal of System Assurance Engineering and Management **13**(Suppl 1), 673–680 (2022)

[8] Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)

[9] Canavese, D., Regano, L., Basile, C., Ciravegna, G., Lioy, A.: Encryption-agnostic classifiers of traffic originators and their application to anomaly detection. Computers & Electrical Engineering **97**, 107621 (2022)

[10] Denning, D.E.: An intrusion-detection model. IEEE Trans. Software Engineering **13**(2), 222–232 (1987)

[11] Duguma, A.L., Bai, X.: How the internet of things technology improves agricultural efficiency. Artificial Intelligence Review **58**(2), 63 (2024)

[12] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., Ranjan, R.: Explainable ai (xai): Core ideas, techniques, and solutions. ACM Comput. Surv. **55**(9) (Jan 2023). https://doi.org/10.1145/3561048

[13] Farnaaz, N., Jabbar, M.: Random forest modeling for network intrusion detection system. Procedia Computer Science **89**, 213–217 (2016)

[14] Giacinto, G., Perdisci, R., Roli, F.: Network intrusion detection by combining one-class classifiers. In: International Conference on Image Analysis and Processing. pp. 58–65. Springer (2005)

[15] Gilad, Y., Herzberg, A.: Fragmentation considered vulnerable. ACM Transactions on Information and System Security (TISSEC) **15**(4), 1–31 (2013)

[16] Gummadi, A.N., Napier, J.C., Abdallah, M.: XAI-IoT: An explainable AI framework for enhancing anomaly detection in IoT systems. IEEE Access **12**, 71024–71054 (2024)

[17] Guo, X., Gao, X.: A syn flood attack detection method based on hierarchical multihead self-attention mechanism. Security and Communication Networks **2022**, 1–13 (09 2022). https://doi.org/10.1155/2022/8515836

[18] Harshita, H.: Detection and prevention of icmp flood ddos attack. International Journal of New Technology and Research **3**(3), 263333 (2017)

[19] Jabbar, M.A., Aluvalu, R., Reddy, S.S.: RFAODE: A novel ensemble intrusion detection system. In: Proc. of Int. Conf. on Computational Intelligence and Data Science (ICCIDS 2017). Procedia Computer Science, vol. 115, pp. 226–234 (2017)

[20] Khraisat, A., Alazab, A.: A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity **4**, 18 (2021)

[21] Kumari, P., Jain, A.K.: A comprehensive study of ddos attacks over iot network and their countermeasures. Computers & Security **127**, 103096 (2023)

[22] Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems (NIPS). vol. 30 (2017)

[23] Masukawa, R., Yun, S., Jeong, S., Huang, W., Ni, Y., Bryant, I., Bastian, N.D., Imani, M.: Packetclip: Multi-modal embedding of network traffic and language for cybersecurity reasoning. arXiv preprint arXiv:2503.03747 (2025)

[24] Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A.: Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089 (2018)

[25] Mubarak, S., Habaebi, M.H., Islam, M.R., Rahman, F., Tahir, M.: Anomaly detection in ICS datasets with machine learning algorithms. Computer Systems Science & Engineering **37**(1), 33–46 (2021)

[26] Pakmehr, A., Aßmuth, A., Taheri, N., Ghaffari, A.: Ddos attack detection techniques in iot networks: a survey. Cluster Computing **27**(10), 14637–14668 (2024)

[27] Rabbani, M., Gui, J., Nejati, F., Zhou, Z., Kaniyamattam, A., Mirani, M., Piya, G., Opushnyev, I., Lu, R., Ghorbani, A.A.: Device identification and anomaly detection in iot environments. IEEE Internet of Things Journal **12**(10), 13625–13643 (2025). https://doi.org/10.1109/JIOT.2024.3522863

[28] Racherla, S., Sripathi, P., Faruqui, N., Kabir, M.A., Whaiduzzaman, M., Shah, S.A.: Deep-ids: A real-time intrusion detector for iot nodes using deep learning. IEEE Access **12**, 63584–63597 (May 2024). https://doi.org/10.1109/ACCESS.2024.3396461

[29] Resende, P.A.A., Drummond, A.C.: A survey of random forest based methods for intrusion detection systems. ACM Computing Surveys (CSUR) **51**(3), 1–36 (2018)

[30] Rizqullah, N.Z., Alekhine, J., Yonia, D.L., Purnomo, R.M.R., Shiddiqi, A.M.: Enhancing intrusion detection systems with adaptive learning techniques. In: 2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS). pp. 1–6. IEEE (2024)

[31] Roesch, M.: Snort: Lightweight intrusion detection for networks. In: Proc. of the 13th USENIX System Administration Conference (LISA '99). pp. 229–238 (1999)

[32] Selvaraj, M., Uddin, G.: A large-scale study of iot security weaknesses and vulnerabilities in the wild. ACM Trans. Softw. Eng. Methodol. **34**(2) (Jan 2025). https://doi.org/10.1145/3691628

[33] Siganos, M., Radoglou-Grammatikis, P., Kotsiuba, I., Markakis, E., Moscholios, I., Goudos, S., Sarigiannidis, P.: Explainable ai-based intrusion detection in the internet of things. In: Proceedings of the 18th international conference on availability, reliability and security. pp. 1–10 (2023)

[34] Sreeram, I., Vuppala, V.P.K.: Http flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm. Applied computing and informatics **15**(1), 59–66 (2019)

[35] Tsimenidis, S., Lagkas, T., Rantos, K.: Deep learning in iot intrusion detection. Journal of network and systems management **30**(1), 8 (2022)

[36] Varghese, S.A., Dehlaghi Ghadim, A., Balador, A., Alimadadi, Z., Papadimitratos, P.: Digital twin-based intrusion detection for industrial control systems. In: Proc. of IEEE PerCom Workshops. pp. 611–617 (2022)

[37] Walling, S., Lodh, S.: Enhancing iot intrusion detection through machine learning with an-sfs: a novel approach to high performing adaptive feature selection. Discover Internet of Things **4**(1), 16 (2024)

[38] Wang, M., Zheng, K., Yang, Y., Wang, X.: An explainable machine learning framework for intrusion detection systems. IEEE Access **8**, 73127–73141 (2020). https://doi.org/10.1109/ACCESS.2020.2988359

[39] Younisse, R., Ahmad, A., Abu Al-Haija, Q.: Explaining intrusion detection-based convolutional neural networks using shapley additive explanations (shap). Big Data and Cognitive Computing **6**(4), 126 (2022)