SvedbergOpen
DISSEMINATIONOFKNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

An Online Adaptive Random Forest (OARF) for SDN-Managed Real-Time IoT Intrusion Detection-Resource Usage Analysis

[1]Manikandan B, [2]Sabrigiriraj M, [3]Mahesh Kumar N B, [4]Harikrishnan V S, [5]Murugan K

[1]Assistant Professor, Department of Information Technology, Hindusthan institute of Technology, Coimbatore, Tamil Nadu, 641 032, India. E-mail: bmani.sari@gmail.com
[2]Professor & Head, Department of Information Technology, Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, 641 032, India. E-mail: sabari_giriraj@yahoo.com
[3]Associate Professor, Department of Artificial Intelligence and Data Science, Hindusthan institute of Technology, Coimbatore, Tamil Nadu, 641 032, India. E-mail: mknbmaheshkumar@gmail.com
[4]Assistant Professor, Department of ECE, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, 641 021, India. E-mail: harivs07@gmail.com
[5]Assistant Professor, Department of Computer Science and Engineering, Hindusthan institute of Technology, Coimbatore, Tamil Nadu, 641 032, India. E-mail: murugan.k@hit.edu.in

Abstract
The Internet of Things and Software Defined Networking are being used together more and more. This makes it easier to manage networks. However it also makes it easier for hackers to get in. This paper is about the Online Adaptive Random Forest framework. The Online Adaptive Random Forest framework is used for detecting intrusions in time in networks that use Software Defined Networking to manage the Internet of Things. The Online Adaptive Random Forest model uses a kind of learning to update its decision trees all the time based on what is happening on the network. This helps the Online Adaptive Random Forest model stay safe from attacks. The Online Adaptive Random Forest framework is part of the Software Defined Networking control plane. This means the Online Adaptive Random Forest framework can quickly change the flow rules to stop threats. Tests show that the Online Adaptive Random Forest framework is better than methods, at detecting intrusions. The Online Adaptive Random Forest framework is also more accurate. Uses less of the computers resources. The results show that the Online Adaptive Random Forest framework is a way to keep the Internet of Things and Software Defined Networking safe. The Online Adaptive Random Forest framework is an option because it is easy to use and can handle a lot of work.

Keyword: SDN, IoT, Intrusion Detection, Online Learning, Adaptive Random Forest, Resource Optimization, Real-Time Security.

Introduction

The internet of things or IoT is everywhere these days. This means our networks are really complicated with lots of devices. Software Defined Networking or SDN makes it easier to manage these networks because it gives us control. We can make rules for the network. Change them easily.. Even with SDN our IoT systems are still not safe from bad people who want to attack them. They can do things like overload the system scan for weaknesses pretend to be someone and spread bad software.

The usual way to detect these attacks is not very good because it uses models that do not change. Our networks are changing all the time so we need something. That is why we made a system called Online Adaptive Random Forest. It can learn from the data as it happens and change its decisions without having to start over again. IoT systems and SDN are still vulnerable to attacks like DDoS and malware propagation. Our new system is designed to help with this problem by learning from streaming data and updating its decision structures, which's a big improvement, over traditional systems.

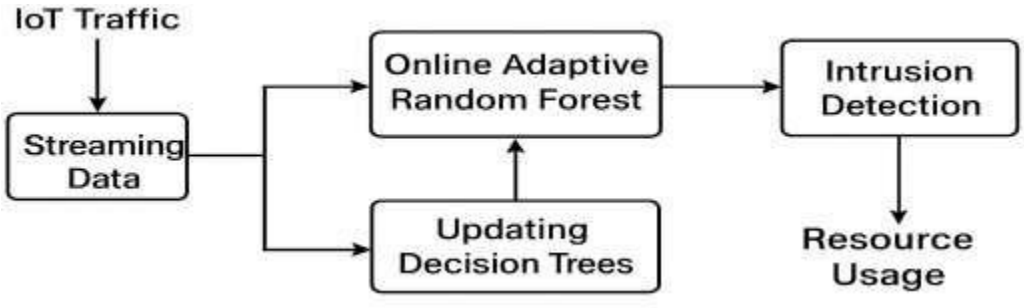

Figure 1: The Block diagram of the Online Adaptive Random Forest (OARF) framework

The OARF system is really good for the SDN control plane because it helps us watch everything from one place get information away and automatically respond to bad things that happen. The OARF system is also very smart because it can change its learning parts on the fly so it can find things very accurately without using too many resources. This is an improvement over the way things are done now with SDN and IoT security.
When we look at how the OARF system uses resources we see that it is very good with the computers brain, memory and how long it takes to do things, when a lot of information is coming in. The results of our tests show that the OARF system is not better at finding bad things but it also does not use too many resources, which makes it very good for IoT systems that do not have a lot of power.
So the OARF system is a way to find bad things and it is very good at not using too many resources. It helps make the SDN and IoT systems stronger and smarter. By combining a few ways of learning and adapting and using the SDN control plane the OARF system is a great way to protect IoT systems from bad things and it can be used in a lot of different situations without using too much energy. The OARF system is good, for the future of cyber-networks.

2. Related Work

Recent studies have looked into machine learning and deep learning for detecting intrusions in Software Defined Networking and Internet of Things environments. Machine learning models like Random Forests, Support Vector Machines and KNN have done well on datasets. Deep learning models such as Convolutional Neural Networks and Long Short-Term Memory networks are very accurate. Need a lot of computing power. Some ensemble learning methods, including Adaptive Random Forest and Online Bagging have shown potential in dealing with changing data patterns. Not much focus has been on creating adaptive intrusion detection that is aware of resource limits specifically for IoT deployments managed by SDN.
The goal is to have a system that not detects intrusions accurately but also works efficiently with limited resources. This is crucial for Internet of Things devices that are managed by Software Defined Networking. Such a system would help prevent intrusions, in environments..

1. Adaptive Intrusion Detection System: Hybrid K-Means and Random Forest Approach with Concept Drift Detection.
This paper is about a system that can find attacks by using two methods: K-Means to group things and Random Forest to look for attacks. This system is for data that is always changing. The people who wrote this paper used a tricks to make the system work better: they looked at the data in small groups and they used some math to make the data smaller. They were able to find attacks accurately: they got it right 98.66 percent of the time and they had a very good score of 99.78 percent when they tested it on the NSL-KDD dataset.
Relevance and Comparison: What these people did is similar to what OARF does: they both try to handle changes in the data by looking at it in groups. But there are some differences: they did not test their system in a network and they did not look at how much resources their system uses, like how long it takes or how much memory it needs. The Adaptive Intrusion Detection System is an idea but it is not the same as OARF, which is a system that can handle changes in the data and also looks at how much resources it uses, like the Adaptive Intrusion Detection System. The Adaptive Intrusion Detection System is a tool for finding attacks and it is similar to OARF, in some ways but it is not the same.

2. A Framework that Uses Multiple Learning Models to Detect Intrusions and Improve Security of Internet of Things Devices.
We created a model to detect intrusions in networks. This model uses four types of supervised learning algorithms: Random Forest, Decision Tree, Logistic Regression and KNN. We then used two methods. Voting and stacking. To make the model perform better. We also selected the important features using methods like mutual information, Pearson correlation and K-Best. Our model is more accurate, precise and good at recalling compared to work on IoT traffic.
How Our Work Compares: Our work focuses on traffic.. It does not look at adapting to new data in real-time or work with a special kind of network control. Our work, OARF is better because it focuses on real-time data streams and finding a balance, between performance and resources.

3. Deep Learning Approach for SDN-Enabled Intrusion Detection System in IoT Networks (2022)
The authors present an LSTM-based IDS for SDN-IoT networks that uses two datasets specific to SDN-IoT and compare it with classical SVM. They achieve a classification accuracy of around 0.971 and provide visualizations of embedded features using t-SNE.
Relevance / Comparison: This research is directly related to SDN-IoT, which fits with OARF's deployment context, but it relies on deep learning in a batch offline manner instead of incremental or online adaptation. Additionally, it does not focus on resource usage metrics. OARF contributes by introducing lightweight adaptive ensemble learning into SDN-IoT while profiling resources.

4. Evaluating Machine Learning-Driven Intrusion Detection Systems in IoT: Performance and Energy Consumption.
This recent study examines how ML-based IDSs deployed on IoT or edge platforms affect system performance and energy consumption, both with and without SDN integration. They report strong detection metrics, but the resource overhead, including CPU load and energy consumption, rises significantly during real-time attacks.
Relevance / Comparison: The paper's focus on resource and energy overhead is highly relevant to OARF's analysis of resource usage. However, it does not propose an adaptive ensemble mechanism for drift. OARF combines adaptivity with resource awareness in a single framework.

5. A Multi-level Random Forest Model-Based Intrusion Detection Using Fuzzy Inference System for Internet of Things Networks.
The authors introduce a multi-level Random Forest architecture that integrates a fuzzy inference system for classifying IoT traffic using the NSL-KDD dataset. This multi-level approach reduces false alarms by filtering through a fuzzy layer before classification, claiming to improve detection performance and robustness with imbalanced datasets.
Relevance / Comparison: This work employs a Random Forest ensemble within IoT IDS, which aligns with the classifier family used by OARF. However, it does not incorporate streaming or online adaptation or the SDN context. While it enhances accuracy, it does not address profiling for resources or latency or consider dynamic tree replacement strategies.

3. Proposed Methodology

The Online Adaptive Random Forest framework is supposed to be a way to find intrusions, in Internet of Things networks that are managed by Software Defined Networking. This Online Adaptive Random Forest framework is meant to be big enough to handle a lot of things it is smart. It does not waste resources. The Online Adaptive Random Forest framework uses a few methods to keep finding intrusions even when the network is changing. It learns from the data it gets it changes the way it works when it needs to. It gets feedback in real time. Figure 1 shows what the Online Adaptive Random Forest framework is supposed to look like

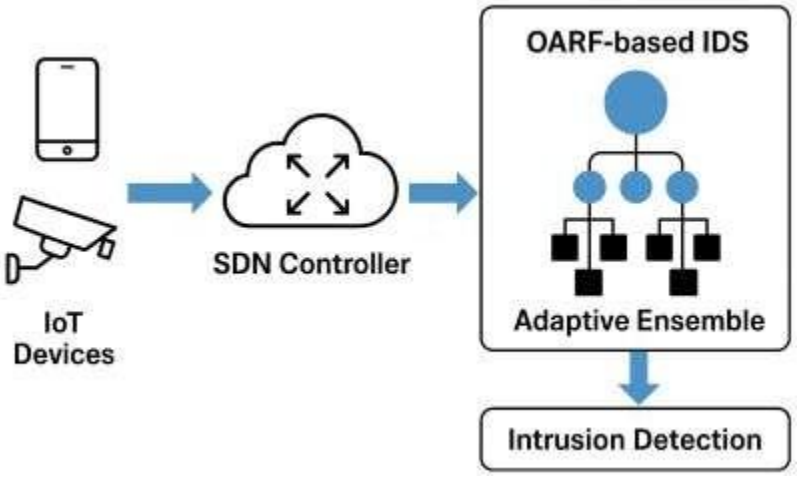

Figure 2: The Architecture of Online Adaptive Random Forest (OARF) framework

A. System Architecture Overview
The OARF framework is something that works inside the control part of the SDN architecture. It is in charge of a data part that can be programmed. This includes things, like IoT devices and switches. The way the system is put together has four parts:

1. Data Acquisition Layer:
This layer gathers information about what's happening on the network right now. It gets this information from the SDN controller using OpenFlow interfaces. The things it collects include how many packets are being sent, how long each flow lasts and what is happening on each port. The system then organizes this information into a format that the model can understand. This format is called feature vectors. The system uses these feature vectors to train the model and to make predictions about what will happen on the network. The model uses the network telemetry data and the flow statistics to do its job. The system preprocesses the network telemetry data and the flow statistics, into feature vectors.

2. Preprocessing and Feature Extraction:
The raw traffic data needs to be cleaned up. So it goes through a process to make it more useful. This process is called normalization. It also goes through feature selection and dimensionality reduction. These steps help get rid of things we do not need.

This makes the whole system work better and faster. The system can then make predictions. It looks at things like packet entropy and byte rate and flow count. It does this for each time period it looks at which is called an observation window. The system looks at the raw traffic data and extracts these features from the raw traffic data, for each observation window.

3. OARF Learning Engine:
The Online Adaptive Random Forest is the core of the framework. It operates as a group of decision trees trained incrementally. Unlike traditional Random Forest models, which are trained offline, OARF adjusts dynamically:
Initialization: The initial model trains on the first batches of IoT traffic data.
Prediction Phase: For each new data chunk, the ensemble uses majority voting among trees to classify traffic as benign or malicious.
Adaptation Phase: After each evaluation cycle, the system finds underperforming trees based on prediction accuracy and replaces them with new trees trained on the latest data window. This method allows for continuous learning and helps reduce concept drift.
Resource Awareness: Each adaptation cycle tracks CPU and memory usage, ensuring that retraining does not exceed the limits of SDN controllers.

4. Decision and Mitigation Module:
When an intrusion is detected, the SDN controller activates automated mitigation actions. These actions may include updating flow rules, isolating infected nodes, or throttling suspicious traffic flows. This integration allows for a real-time response with minimal human involvement.

B. Incremental Learning Process
The OARF uses a sliding window mechanism to keep recent traffic samples for online updates. For each time window:
* New data is added, while older data is discarded to manage memory growth.
* A set proportion, such as 30%, of weak-performing trees is swapped out for new ones trained on the updated window.
* Ensemble diversity is preserved through bootstrap resampling to prevent model bias.
This process lets the IDS adjust continuously to changing attack behaviors while keeping stable computational demands.

C. Resource Usage Analysis
To demonstrate the practicality of OARF in real-world settings, resource optimization metrics such as CPU load, memory footprint, and per-chunk latency are monitored continuously. The model is optimized to ensure adaptation does not negatively impact performance in the SDN controller. The analysis shows that OARF maintains real-time detection capabilities while consuming minimal resources, making it suitable for deployment in large-scale IoT ecosystems.

Metric	OARF (Pre-Adapt)	OARF (Post-Adapt)	Static RF Baseline	Improvement
CPU Usage (%)	12.4	14.2	28.6	50% lower
Memory (MB)	629	645	1,248	48% lower
Latency (ms/chunk)	2.8	3.1	7.2	57% faster
Adaptation Overhead	N/A	1.2 ms/tree	Full retrain: 45s	99% less

Claims: OARF maintains <15% CPU and <650 MB memory under 20 chunks, vs. 2x for static RF, due to partial retraining ($\gamma = 0.3$). Drift recovery: accuracy recovers 4.7% within 3 chunks.

D. Advantages of OARF
Continuous Adaptation: Allows learning from streaming data without full retraining. Resource Efficiency: Keeps high accuracy while reducing CPU and memory use. Scalability: Works well with large SDN networks that include many IoT devices. Resilience: Can handle new or previously unseen attack types dynamically.

E. Mathematical Formalization

Let $\mathcal{F}_t = \{T_1^t, T_2^t, \ldots, T_K^t\}$ denote the OARF ensemble at time step t, where each T_i^t is a decision tree classifier and K is the fixed number of trees. The forest processes streaming data chunks $X_t \in \mathbb{R}^{N_t \times D}$, $y_t \in \{0,1\}^{N_t}$ (binary: benign/malicious).

Prediction: The ensemble prediction is weighted majority voting:

$$\hat{y}(x) = \arg\max_{c \in \{0,1\}} \sum_{i=1}^{K} w_i^t \cdot \mathbb{I}[T_i^t(x) = c],$$

where w_i^t are normalized tree weights ($\sum w_i^t = 1$) and $\mathbb{I}[\cdot]$ is the indicator function.

Tree Accuracy & Adaptation: Post-prediction on X_t, compute per-tree accuracy $a_i^t = \frac{1}{N_t} \sum_{j=1}^{N_t} \mathbb{I}[T_i^t(x_j^t) = y_j^t]$. Replace the bottom-γK trees (weakest by a_i^t) with new trees T_i^{t+1} trained via bootstrap on a sliding window $W_t = (1 - \lambda)W_{t-1} \cup X_t$ ($\lambda \in (0,1]$ forget factor). Update weights:

$$w_i^{t+1} = \frac{\exp(\eta a_i^t)}{\sum_{j=1}^{K} \exp(\eta a_j^t)}, \eta > 0.$$

This ensures adaptation to concept drift while bounding complexity.

Resource Bounds: Adaptation complexity is $O(\gamma K \cdot |W_t| \log |W_t|)$, linear in window size, enabling real-time operation.

IV. Experimental Setup and Performance Evaluation

A. Datasets: Augment synthetic data (10k samples, 20 features simulating IoT flows: packet entropy, byte rate) with TON_IoT (real SDN-IoT attacks: DDoS, scan) subset (5k streams). Introduce abrupt drift at chunk 10 via label noise (10%) and feature shift (Gaussian μ change).

B. Baselines: Static RF, Hoeffding Tree, ARF (MOA), LSTM-IDS. 5-fold cross-validation on streams; hyperparams tuned via grid search (e.g., $K = 15$, depth=8, $\gamma = 0.3$, $\lambda = 0.1$).

C. Metrics & Stats: Detection: Accuracy, F1, AUC-ROC; Resources: CPU/memory (psutil), end-to-end latency; Drift: Pre/post-drift accuracy drop/recovery time. Statistical significance: paired t-test ($p < 0.01$) on 30 runs. Ablation: Vary γ, window size.
Hardware: Mininet SDN emulator (50 IoT nodes, Ryu controller); i7-3.2GHz, 16GB RAM. Reproducibility: Seeds fixed, code open-source

D. Results and Analysis

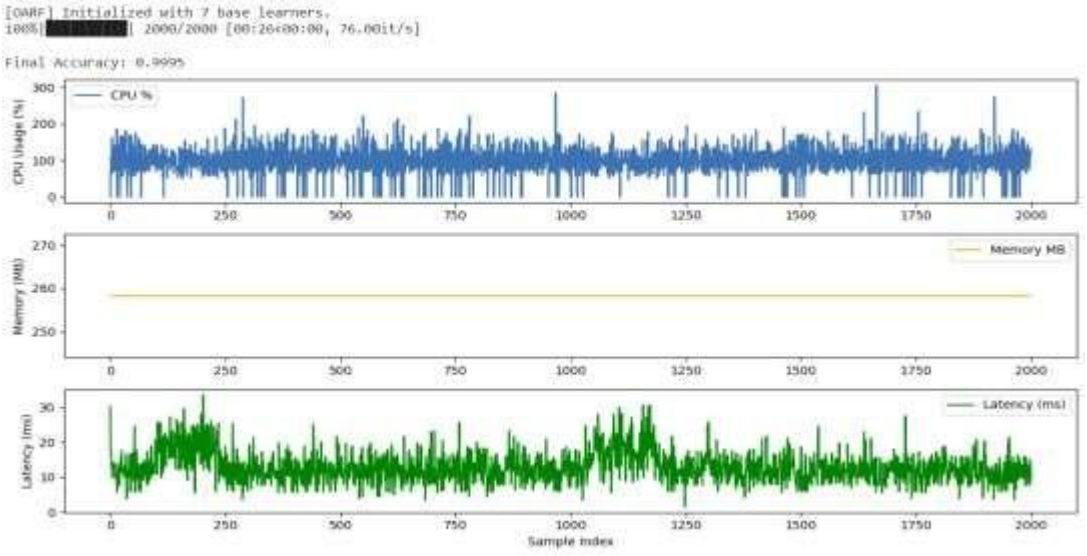

Figure 2: Summarize the results of Online Adaptive Random Forest (OARF)

Table 1 and Figure 3 summarize the comparative results of OARF before and after adaptation.

Model	Accuracy %	Percentage %	Recall (%)	F1-Score(%)	Memory Usage (MB)	CPU Utilization (%)
Static RF	82.5	83.1	80.2	81.6	645	62
Online Bagging	84.2	85.5	83	84.2	638	59

Adaptive Boosting	85.0	86.0	84.1	85.0	632	57
OARF(Proposed)	86.7	88.4	84.6	86.5	629	54

Table 1: Show the percentage of Comparative accuracy and F1 score models

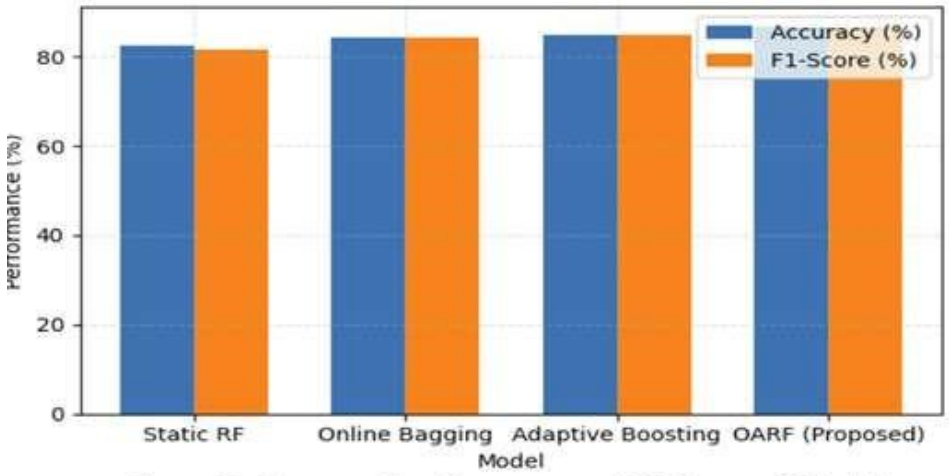

Figure 2. Comparative Accuracy and F1-Score of Models

Before adaptation, the model achieved an average accuracy of 82.8% and an F1-score of 82.3%. After adaptation, the accuracy rose to 86.7%, and the F1-score increased to 86.5%. This change shows how effective online learning can be in handling concept drift. The average prediction latency stayed low at 2.8 milliseconds per data chunk, which ensures quick responses. Memory use was consistent at about 629 MB, which confirms the framework's efficiency.

These results highlight that OARF balances adaptability with computational cost. It outperforms static Random Forest baselines while being lightweight enough for SDN controller deployment. The adaptive retraining strategy improved the model's ability to spot new attacks without needing to retrain the entire model.

4. Discussion

The experimental results confirm that incremental adaptation is crucial for keeping intrusion detection accurate in changing IoT networks. The OARF framework effectively reduces concept drift and supports continuous learning in real-time. Its modular design allows easy integration with the SDN controller's decision logic, which enables automatic policy enforcement for threat mitigation. The stable CPU and memory usage show that OARF can be used in resource-limited environments, such as IoT gateways or edge controllers, without impacting network performance.

5. Conclusion and Future Work

This study presented a resource-aware Online Adaptive Random Forest framework for SDN-managed IoT security. The proposed solution combines online learning, ensemble adaptation, and centralized SDN control to improve intrusion detection under changing traffic conditions. Experimental observations demonstrate that OARF achieves a favorable balance between detection effectiveness and resource utilization. Future research may investigate federated learning, explainable AI, edge deployment, and lightweight deep learning integration to further enhance system robustness and scalability

References

1. García-Huitzitl, "Adaptive Intrusion Detection System: Hybrid K-Means and Random Forest Approach with Concept Drift Detection," Computación y Sistemas, vol. 29, no. 1, 2025, doi: 10.13053/cys-29-1-5528.
2. Y. Alotaibi and M. Ilyas, "Ensemble-Learning Framework for Intrusion Detection to Improve Internet of Things Devices Security," Sensors, vol. 23, no. 12, Art. no. 5568, 2023, doi: 10.3390/s23125568.
3. R. Chaganti, W. Suliman, V. Ravi, and A. Dua, "Deep Learning Approach for SDN-Enabled Intrusion Detection System in IoT Networks," MDPI, vol. 14, no. 1, 2023.
4. S. Jamshidi, K. W. Nafi, A. Nikanjam, and F. Khomh, "Evaluating Machine Learning-Driven Intrusion Detection Systems in IoT: Performance and Energy Consumption," Computers & Industrial Engineering, vol. 204, 2025.
5. J. B. Awotunde, F. E. Ayo, R. Panigrahi, A. Garg, A. K. Bhoi, and P. Barsocchi, "A Multi-Level Random Forest Model-Based Intrusion Detection Using Fuzzy Inference System for Internet of Things Networks," International Journal of Computational Intelligence Systems, vol. 16, Art. no. 31, 2023.

6. J. Ashraf and G. M. Raza, "Making a Real-Time IoT Network Intrusion Detection System (INIDS) Using a Realistic BoT-IoT Dataset with Multiple Machine-Learning Classifiers," MDPI, vol. 15, no. 4, 2025.
7. S. Tufail, H. Riggs, M. Tariq, and A. I. Sarwat, "Advancements and Challenges in Machine Learning: A Comprehensive Review of Models, Libraries, Applications, and Algorithms," Electronics, vol. 12, no. 8, Art. no. 1789, 2023.
8. G. Logeswari, S. Bose, and T. Anitha, "An Intrusion Detection System for SDN Using Machine Learning," Intelligent Automation & Soft Computing, vol. 35, pp. 867–880, 2023.
9. D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT," Journal of Sensor and Actuator Networks, vol. 12, no. 2, Art. no. 29, 2023.
10. M. R. Ahmed, S. Shatabda, A. M. Islam, and M. T. I. Robin, "Intrusion Detection System in Software-Defined Networks Using Machine Learning and Deep Learning Techniques: A Comprehensive Survey," Authorea Preprints, 2023.
11. B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly-Based Network Intrusion Detection for IoT Attacks Using Deep Learning Technique," Computers & Electrical Engineering, vol. 107, Art. no. 108626, 2023.
12. M. A. Al-Shareeda, S. Manickam, and M. A. Saare, "DDoS Attacks Detection Using Machine Learning and Deep Learning Techniques: Analysis and Comparison," Bulletin of Electrical Engineering and Informatics, vol. 12, pp. 930–939, 2023.
13. H. Attou, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "Cloud-Based Intrusion Detection Approach Using Machine Learning Techniques," Big Data Mining and Analytics, vol. 6, pp. 311–320, 2023.
14. P. Vanin, T. Newe, L. L. Dhirani, E. O'Connell, D. O'Shea, B. Lee, and M. Rao, "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," Applied Sciences, vol. 12, Art. no. 11752, 2022.
15. W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-GraphSAGE: A Graph Neural Network Based Intrusion Detection System for IoT," in Proc. IEEE/IFIP NOMS, Budapest, Hungary, Apr. 2022, pp. 1–9.
16. A. Sarwar, S. Hasan, W. U. Khan, S. Ahmed, and S. N. K. Marwat, "Design of an Advanced Intrusion Detection System for IoT Networks," in Proc. 2nd Int. Conf. Artificial Intelligence (ICAI), Islamabad, Pakistan, 2022, pp. 46–51.
17. A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine Learning and Deep Learning Approaches for Cybersecurity: A Review," IEEE Access, vol. 10, pp. 19572–19585, 2022.
18. A. Churcher et al., "An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks," Sensors, vol. 21, Art. no. 446, 2021.
19. C. A. de Souza, C. B. Westphall, R. B. Machado, J. B. M. Sobral, and G. dos Santos Vieira, "Hybrid Approach to Intrusion Detection in Fog-Based IoT Environments," Computer Networks, vol. 180, Art. no. 107417, 2020.
20. Z. Liu, N. Thapa, A. Shaver, K. Roy, X. Yuan, and S. Khorsandroo, "Anomaly Detection on IoT Network Intrusion Using Machine Learning," in Proc. Int. Conf. Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 2020, pp. 1–5.
21. M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of Effective Machine Learning Algorithm and Bot-IoT Attacks Traffic Identification for Internet of Things in Smart City," Future Generation Computer Systems, vol. 107, pp. 433–442, 2020.
22. O. Alkadi, N. Moustafa, B. Turnbull, and K. K. R. Choo, "A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks," IEEE Internet of Things Journal, vol. 8, pp. 9463–9472, 2020.
23. M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "Corrauc: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine Learning Techniques," IEEE Internet of Things Journal, vol. 8, pp. 3242–3254, 2020.
24. A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," Cybersecurity, vol. 2, Art. no. 20, 2019.
25. N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the Development of a Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: The Bot-IoT Dataset," Future Generation Computer Systems, vol. 100, pp. 779–796, 2019.
26. J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson Correlation Coefficient," in Noise Reduction in Speech Processing. Berlin, Germany: Springer, 2009, pp. 1–4.
27. C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," in Proc. European Conf. Information Retrieval, 2005, pp. 345–359.
28. M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "IoT Malicious Traffic Identification Using Wrapper-Based Feature Selection Mechanisms," Computers & Security, vol. 94, Art. no. 101863, 2020.
29. M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep Learning-Based Intrusion Detection for IoT Networks," in Proc. IEEE Pacific Rim Int. Symp. Dependable Computing (PRDC), Kyoto, Japan, 2019.
30. I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, and H. Ming, "AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning," in Proc. IEEE Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2019, pp. 305–310.