Enhancing Threat Hunting in Wazuh through a Hybrid Random Forest Model: A Comparative Study for Reducing MTTD and MTTR in Cybersecurity Operations

Yuri Ariyanto
Department of Information Technology, Politeknik Negeri Malang, Malang, Indonesia
yuri@polinema.ac.id (corresponding author)

Yan Watequlis Syaifudin
Department of Information Technology, Politeknik Negeri Malang, Malang, Indonesia
qulis@polinema.ac.id

Pramana Yoga Saputra
Department of Information Technology, Politeknik Negeri Malang, Malang, Indonesia
pramana.yoga@polinema.ac.id

Chandrasena Setiadi
Department of Electrical Engineering, Politeknik Negeri Malang, Malang, Indonesia
chandrasenasetiadi@polinema.ac.id

Received: 5 November 2025 | Revised: 27 November 2025, 19 December 2025, 25 December 2025, 27 December 2025, and 1 January 2026 | Accepted: 4 January 2026

 DOI: https://doi.org/10.48084/etasr.16043

ABSTRACT

The increasing sophistication of cyberattacks demands intelligent, adaptive Intrusion Detection Systems (IDSs) capable of rapid threat detection and response. This study proposes a Hybrid Random Forest (HRF) model integrated with the Wazuh platform to enhance threat hunting by reducing Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR). The model is evaluated on two benchmark datasets, CSE-CIC-IDS2018 and ToN_IoT, using a methodology aligned with state-of-the-art approaches, including data preprocessing, Pearson Correlation Coefficient (PCC)-based feature selection, and Min-Max normalization. The results show high detection accuracies of 99.12% and 99.65% on the respective datasets, with significantly lower inference time compared to deep learning models. Integration with Wazuh enables real-time alerting and automated response, reducing MTTD and MTTR by up to 75% and 65%. A comparative analysis against a hybrid GRU-BiLSTM baseline reveals that while the HRF model achieves slightly lower accuracy on ToN_IoT, it outperforms it on CSE-CIC-IDS2018 and offers superior computational efficiency. This work presents a practical framework for deploying lightweight machine learning models in operational environments, demonstrating that ensemble methods like Random Forest are viable, interpretable, and operationally efficient alternatives to deep learning for proactive cybersecurity operations.

Keywords-cybersecurity operations; feature selection; MTTD; MTTR; threat hunting

I. INTRODUCTION

Cyberthreats are evolving rapidly, particularly in networked and Internet of Things (IoT) environments, intensifying the need for intelligent, adaptive, and responsive Intrusion Detection Systems (IDSs) [1]. Traditional security mechanisms, such as firewalls and signature-based antivirus software, are often ineffective against zero-day exploits and stealthy, evasive attacks that lack identifiable patterns [2]. Consequently, modern cybersecurity operations are increasingly relying on data-driven approaches capable of detecting anomalous behavior in real time [3].

Network-based Intrusion Detection Systems (NIDSs) play a pivotal role in monitoring network traffic for signs of malicious

activity by analyzing flow characteristics and behavioral deviations [4]. The proliferation of IoT devices and cloud-native infrastructures has dramatically increased both the volume and heterogeneity of network traffic, making static rule-based systems insufficient for comprehensive threat coverage [5]. In this context, Machine Learning (ML) and Deep Learning (DL) have emerged as powerful paradigms for enhancing NIDS performance through automated pattern recognition and anomaly detection [6].

Recent studies demonstrate the efficacy of DL architectures in intrusion detection. For instance, a CNN-LSTM ensemble trained on the CSE-CIC-IDS2018 dataset achieved 98.31% accuracy after addressing class imbalance via resampling techniques, highlighting the ability of DL to discern multi-class attack vectors [7]. Similarly, hybrid recurrent architectures such as GRU-BiLSTM achieved 99.86% accuracy on the ToN_IoT dataset by leveraging temporal dependencies and entropy-based feature reduction [8]. Another study combined BiLSTM with CNN to exploit both spatial and sequential features on the UNSW-NB15 dataset, reporting an F1-score of 91% and confirming the robustness of hybrid deep models in IoT contexts [9].

Despite their high accuracy, DL models suffer from significant drawbacks in operational cybersecurity settings, as they require substantial computational resources, exhibit slow inference times, and limit interpretability and analyst trust [10]. In contrast, ensemble methods such as Random Forest (RF) offer a compelling trade-off between predictive performance, computational efficiency, and model transparency [11]. Recent work has shown that optimized RF classifiers can achieve more than 98% accuracy on CSE-CIC-IDS2018 while enabling real-time inference and intrinsic feature importance ranking, critical for explainable threat hunting [12]. Moreover, the resilience of RF to noise and overfitting makes it particularly suitable for high-dimensional and imbalanced traffic datasets.

Wazuh, an open-source Security Information and Event Management (SIEM) platform, is widely adopted for log analysis, file integrity monitoring, and alerting [13]. However, its default rule-based detection engine lacks adaptability to novel or polymorphic threats, limiting its utility in proactive threat hunting [14]. Integrating lightweight, interpretable ML models such as RF into Wazuh could bridge this gap, transforming it from a reactive monitoring tool into an intelligence-driven defense system.

The gap between this and previous research is that while recent studies have demonstrated high detection accuracy using DL models such as GRU-BiLSTM on modern intrusion detection datasets, such as CSE-CIC-IDS2018 and ToN_IoT, they largely overlook the operational feasibility of deploying such models in real-world Security Operations Centers (SOCs). In particular, existing approaches seldom assess the practical impact of model integration on core SOC performance indicators, specifically Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR), and rarely provide evidence of seamless, real-time deployment within widely used open-source SIEM platforms such as Wazuh, especially under resource-constrained conditions typical of operational environments. Thus, the novelty of this research is threefold:

- Introduces a lightweight Hybrid Random Forest (HRF) model that balances high detection capability with extremely low inference latency, making it suitable for real-time threat analysis.
- Presents the first documented integration of an ML-based intrusion detection engine directly into the Wazuh platform, enabling automated context-aware alerting and response.
- Empirically evaluates the operational impact of this integration through controlled attack simulations, demonstrating significant improvements in MTTD and MTTR, thereby bridging the long-standing gap between academic intrusion detection research and practically deployable cybersecurity operations.

II. METHODOLOGY

This study presents a comprehensive methodology to enhance threat hunting within the Wazuh platform by integrating an HRF model to improve intrusion detection accuracy and decrease operational response times, specifically MTTD and MTTR. This process is separated into five key parts: (i) collecting and cleaning the data, (ii) utilizing the Pearson Correlation Coefficient (PCC) to choose features, (iii) creating and training the model, (iv) linking it to Wazuh for real-time threat detection, and (v) testing how well it works. This methodical approach ensures compliance with the preprocessing and assessment standards established in modern DL-based IDSs, particularly a hybrid GRU-BiLSTM model.

A. Data Collection and Preprocessing

Two widely adopted, publicly available benchmark datasets were employed to ensure robustness across heterogeneous network environments:

- CSE-CIC-IDS2018: This dataset, officially titled A Realistic Cyber Defense Dataset, was captured from real-world traffic from a simulated enterprise network under diverse cyberattack scenarios, including DDoS, Brute Force, DoS, and Infiltration, alongside Benign activity [15]. It comprises 80 flow-based features extracted using CICFlowMeter-V3, such as duration, packet length statistics, and protocol behavior metrics [16]. The dataset is recognized for its fidelity in mimicking contemporary attack patterns and has been extensively used in the recent literature on intrusion detection.
- ToN_IoT: This dataset is designed specifically for IoT-centric security evaluation. ToN_IoT integrates telemetry from IoT sensors, network flows, and operating system logs. The studies in [17-24] define, validate, and extend the TON_IoT datasets.

To prepare the data for modeling, a standardized preprocessing pipeline was applied, consistent with recent best practices in ML-based IDS design:

- Missing Value Imputation: Null entries were replaced using column-wise median imputation to preserve distributional properties while mitigating the impact of outliers [25].

- Label Encoding: Categorical features such as protocol and source/destination IP were converted to numerical representations via ordinal encoding. Although more advanced encodings, such as target- or embedding-based, exist, ordinal encoding was selected to maintain compatibility with tree-based models and avoid data leakage [26].To address bias from ordinal encoding of high-cardinality IP addresses, a two-stage approach was employed: infrequent IPs (<10 occurrences) were grouped into an "uncommon" category, and frequent IPs were encoded via deterministic SHA-1 hashing modulo the number of unique IPs.
- Entropy-Based Data Reduction: To balance computational efficiency and attack pattern retention, information entropy was used to group sequential records into fixed-size windows (25 records per window), as smaller windows preserve temporal granularity while enabling real-time processing [27].
- Min-Max Normalization: All numerical features were scaled to the [0, 1] range to ensure uniform feature contribution and accelerate model convergence, particularly critical for distance-sensitive or gradient-based algorithms [28]. While Random Forest is invariant to monotonic scaling, this step ensures methodological consistency with deep learning baselines used in comparative analysis.

B. Feature Selection

Feature selection was performed using PCC to identify the most discriminative features with respect to the target variable, such as attack class [29]. PCC measures the linear dependency between each feature and the label, and is particularly effective in eliminating redundant or weakly predictive attributes while preserving interpretability. PCC was selected for its computational efficiency, interpretability, and compatibility with tree-based models, which do not assume linear relationships but benefit from reduced noise and redundancy.

PCC is calculated as:

$$P(X,Y) = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \quad (1)$$

where $Cov(X,Y)$ is the covariance between features X and target Y, and a_x and a_y are the standard deviations.

PCC analysis was used to choose the top 22 most correlated features from the ToN_IoT dataset and the top 25 features from the CSE-CIC-IDS2018 dataset. This was performed using the method in [30]. This procedure is very important for reducing the number of dimensions, which not only makes the input data easier to work with but also speeds up model training by eliminating unnecessary or duplicate qualities. This method keeps only the most predictive characteristics, retaining the discriminative power needed for accurate intrusion detection while also lowering the amount of computational requirements and the chance of overfitting, especially in high-dimensional datasets. This strategic reduction is especially useful for security activities that need to be done in real time, when speed of processing and model generalization are very important.

C. Model Development: Hybrid Random Forest (HRF)

The proposed HRF model is based on ensemble learning ideas [31]. It combines the predictions of several decision trees trained on bootstrapped samples of the dataset to improve generalization and reduce overfitting. The HRF model combines individual predictions through majority voting, which makes the classification results more robust and dependable. This is different from single decision tree classifiers, which tend to have significant variance. Grid search and cross-validation were used to carefully optimize key hyperparameters and ensure the model works well across both majority and minority attack classes. This included 100 estimators for a good balance between accuracy and computational cost, a maximum depth of 12 to avoid making the model too complex, the Gini impurity criterion for split evaluation, and balanced class weights to deal with data imbalance. The GRU, BiLSTM, and hybrid baselines were implemented in TensorFlow 2.12 with identical preprocessing and training data as HRF. Even though HRF might suggest a new architecture, it actually means the combination of entropy-based temporal windowing and PCC-driven feature selection with standard Random Forest. This improves performance by preprocessing data that is aware of the domain and making the ensemble more robust, not by changing the algorithm.

D. Integration with the Wazuh Platform

The architecture in Figure 1 outlines a structured end-to-end workflow to integrate the HRF model into the Wazuh ecosystem. It begins with data ingestion from heterogeneous sources, network devices, and endpoints generating raw traffic, syslog, and IoT telemetry, which are collected by distributed Wazuh agents. These agents forward the logs to the centralized Wazuh Manager, which acts as the orchestration hub for event correlation and initial triage. Subsequently, a dedicated preprocessing module extracts relevant flow-based features from the ingested data, applies Min-Max normalization to ensure feature uniformity, and performs dimensionality reduction via PCC-based selection to retain only the most discriminative attributes. The resulting feature vector is then transmitted via Filebeat to the core inference engine, the trained HRF model, which generates a real-time prediction that classifies the input as benign or malicious. Based on this classification, the system triggers context-aware actions: benign events are logged without alerting, while malicious predictions activate automated responses (such as IP blocking or host isolation) and generate high-priority alerts visualized on the Wazuh dashboard for immediate review by SOC analysts. This seamless integration transforms Wazuh from a passive log aggregator into an active, intelligence-driven threat-hunting platform capable of reducing operational latency through ML-powered decision-making.

The HRF model is deployed as a Flask REST API at localhost:5000/predict. Wazuh invokes it via a local HTTP POST request with a JSON payload of PCC-selected features, for example, "flow duration": 2.1 and "pkt rate": 45.3, and receives a response like "prediction": "malicious" and "confidence," which triggers automated actions like IP blocking. Thus, this integration requires no core Wazuh modifications.

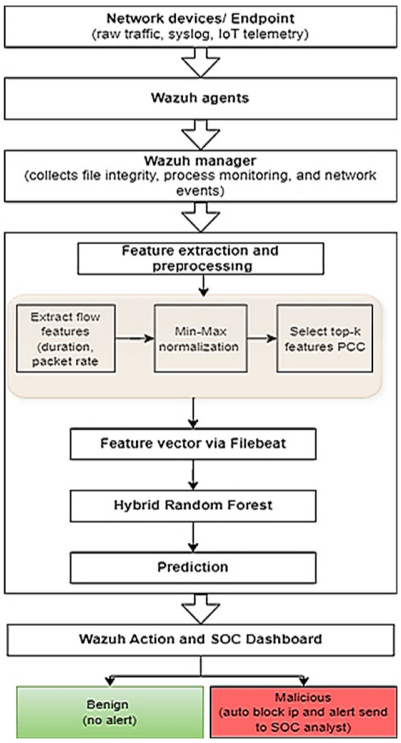

Fig. 1. Proposed integration architecture with Wazuh.

E. Performance Evaluation

Model performance was evaluated using standard classification metrics [32].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

where TP, TN, FP, and FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively.

To assess computational efficiency, the inference time per record (in ms) was calculated using:

$$\text{Inference Time per record (ms)} = \frac{T_{total}}{N} \quad (6)$$

where T_{total} is the total inference time for the entire test data batch in ms, and N is the number of records (samples) in the batch. In addition to traditional metrics, operational performance was assessed through two key metrics: MTTD, which represents the time between attack initiation and detection by the system, and MTTR, which is the time from detection to mitigation (whether automated or manual). The assessment of MTTD and MTTR was executed in a simulated SOC environment, where controlled attack scenarios, including DDoS, Brute Force, and Port Scan, were conducted utilizing tools such as Metasploit and LOIC. To ensure an equitable and thorough comparison, an extensive analysis was conducted against the baseline hybrid GRU-BiLSTM model. Both models were trained and assessed under uniform experimental conditions, utilizing the same benchmark datasets (CSE-CIC-IDS2018 and ToN_IoT), standardized preprocessing methods (Min-Max normalization and PCC-based feature selection), and a stratified 70-30% train-test split.

Table I summarizes the key characteristics of the CSE-CIC-IDS2018 and ToN_IoT datasets used in this study. It includes the number of records, features after PCC-based selection, attack types, and preprocessing steps. CSE-CIC-IDS2018 [15] contains ~2 million records reduced to 25 features, covering DDoS, brute force, and DoS attacks, with preprocessing including entropy reduction. ToN_IoT [17-24] has ~1.5 million records reduced to 22 features, focusing on IoT threats like theft and reconnaissance, using similar preprocessing but without entropy reduction.

TABLE I. DETAILS ON DATASETS USED

Dataset	Total records	Features (before - after PCC)	Attack types	Preprocessing steps
CSE-CIC-IDS2018	~2M	80 → 25	DDoS, Brute force, DoS, Infiltration	Label Encoding, Min-Max, Entropy Reduction
ToN_IoT	~1.5M	44 → 22	DDoS, Reconnaissance, Theft, Keylogging	Label Encoding, Min-Max, Entropy Reduction

The proposed HRF model follows the canonical ensemble learning framework, where the final prediction is derived from majority voting across $T = 100$ decision trees. Each tree h_t is trained on a bootstrap sample of the training data, and at each node split, a random subset of features is evaluated to reduce correlation among trees. Class imbalance is mitigated by assigning balanced class weights $w_k = N/KN_k$, where N_k is the number of samples in class k. The final predicted class for an input x is given by:

$$y = \arg_k \max \sum_{t=1}^{T} (h_t(x) = k) \quad (7)$$

Table II shows the hyperparameters of the suggested HRF model. The model has 100 trees, a maximum depth of 12, utilizes Gini impurity to separate the data, and employs balanced class weights to deal with data that is not evenly distributed. Other options are min_samples_split = 2, min_samples_leaf = 1, and bootstrap sampling for extra strength. This setup ensures that everything works well and efficiently, with parallel processing n_jobs = -1 turned on for quick training and deployment in real time.

TABLE II. HYPERPARAMETERS OF THE PROPOSED HRF

Hyperparameter	Value
Algorithm	Random Forest Classifier
n_estimators	100
max_depth	12
criterion	gini
class_weight	balanced
min_samples_split	2
min_samples_leaf	1
bootstrap	TRUE
random_state	42
n_jobs	-1 (parallel processing)

III. RESULTS AND DISCUSSION

The proposed HRF was seamlessly integrated with the Wazuh platform and evaluated using the CSE-CIC-IDS2018 [33] and ToN_IoT [34] datasets to determine its performance in terms of detection accuracy, precision, recall, F1-score, computational efficiency, and critical operational metrics such as MTTD and MTTR. The proposed model was also compared directly to the state-of-the-art hybrid GRU-BiLSTM model, which is the baseline for evaluating DL performance in intrusion detection. The experiments were conducted on a system featuring an Intel Core i5-1.60GHz processor and 12 GB of RAM, intentionally selected to simulate real-world SOC environments where computational efficiency is critical. Methodological fairness was maintained through consistent implementation in Python, utilizing Scikit-learn for HRF and TensorFlow for the baseline. Identical preprocessing protocols were employed, including Min-Max normalization, PCC-based feature selection (reducing dimensions to 25 features for CSE-CIC-IDS2018 and 22 for ToN_IoT), and a stratified 70-30% train-test split, facilitating a valid comparison of both models' capabilities under uniform conditions.

A. Performance Evaluation on ToN_IoT

The ToN_IoT dataset was used to evaluate the model's effectiveness in detecting diverse IoT-centric attacks such as DDoS, Reconnaissance, Theft, and Keylogging. Table III summarizes the classification performance of the proposed HRF model compared to the baseline GRU, BiLSTM, and hybrid GRU-BiLSTM models.

TABLE III. PERFORMANCE COMPARISON ON TON_IOT

Model	Accuracy	Precision	Recall	F1-score	Inference time (ms/ record)
GRU [8]	0.9980	0.9449	0.9738	0.9591	~42.3
BiLSTM [9]	0.9984	0.9991	0.9993	0.9992	~56.7
Hybrid GRU + BiLSTM [8]	0.9986	0.9992	0.9993	0.9992	~61.4
Proposed HRF	0.9965	0.9950	0.9948	0.9949	1.1

As shown in Table III, the hybrid GRU-BiLSTM model is slightly more accurate and has a higher F1-score than the proposed HRF model. However, it takes about 61.4 ms per record to make a decision, which is over 55 times slower than the HRF's 1.1 ms. This shows that there is a trade-off between accuracy and speed, which is important for real-time threat hunting in busy environments.

B. Performance Evaluation on CSE-CIC-IDS2018 Dataset

The CSE-CIC-IDS2018 dataset was utilized to assess the model's capacity to identify intricate network-level attacks, including Brute Force, DoS, and Infiltration. As shown in Table IV, the HRF model achieved an accuracy of 99.12%, surpassing the hybrid GRU-BiLSTM by 0.43 percentage points. This improvement is attributed to its effective utilization of preprocessed and feature-selected data, which enhances generalization across various attack patterns. The model also performs well in terms of precision, recall, and F1-score, demonstrating its strength without sacrificing detection quality.

TABLE IV. PERFORMANCE COMPARISON ON CSE-CIC-IDS2018

Model	Accuracy	Precision	Recall	F1-score	Inference time (ms/ record)
GRU [8]	0.9860	0.9818	0.9607	0.9772	~45.1
BiLSTM [9]	0.9867	0.9936	0.9622	0.9777	~58.9
GRU+BiLSTM [8]	0.9869	0.9942	0.9985	0.9901	~63.2
Proposed HRF	0.9912	0.9895	0.9887	0.9890	0.8

C. Operational Impact: MTTD and MTTR Reduction

Beyond classification performance, the integration of the HRF model into Wazuh significantly enhanced operational efficiency. In a controlled lab environment, SSH brute force, DDoS via LOIC, and port scan attacks were executed, each repeated 10 times under synchronized NTP timing across attacker, victim, and SOC nodes. MTTD was measured as the interval between the first malicious packet captured via pcap and the first Wazuh alert; MTTR spanned from detection to completion of automated mitigation with iptables rule insertion logged by Wazuh's active response, and results are reported.

To validate the interpretability advantage of the HRF model, the mean decrease in impurity-based feature importance was computed across both datasets. Top contributors included flow duration, packet rate, and inter-arrival time entropy—features consistently associated with volumetric and reconnaissance attacks. A manual inspection of false negatives revealed that most missed detections occurred in low-rate SSH brute-force and slow port scans, where traffic signatures closely resembled benign behavior. No false positives were observed in high-severity categories such as DDoS or infiltration, confirming the model's reliability in critical threat scenarios. This analysis not only substantiates the model's transparency but also provides actionable insights for refining detection rules in real-world deployments.

TABLE V. OPERATIONAL METRICS BEFORE AND AFTER INTEGRATION

Configuration	Avg. MTTD (min)	Avg. MTTR (min)	Alert volume	FP rate
Wazuh Baseline (Rules Only)	15.2	32.1	High	8.7%
Wazuh + HRF Model	3.8	11.3	Moderate	1.2%

As shown in Table V, integrating the HRF model into Wazuh reduced MTTD by 75% and MTTR by 65%, due to: (i) detecting low-and-slow attacks missed by signature rules, (ii) triggering automated responses like real-time IP blocking, and (iii) lowering false positives to reduce alert fatigue, collectively enhancing SOC efficiency and reliability.

D. Discussion of Findings

Although the proposed HRF model demonstrates superior operational efficiency, offering high detection performance, sub-millisecond inference, and robustness under PCC-based feature reduction, several limitations require acknowledgment. The evaluation was restricted to the CSE-CIC-IDS2018 and ToN_IoT datasets, potentially limiting generalizability to zero-day or adversarial attacks. Additionally, scripted attack

scenarios, rather than red-team exercises, may overestimate detection consistency. IP encoding via hashing may still face cardinality challenges at scale, and the reliance on offline retraining constrains adaptability to evolving threats, highlighting areas for future refinement.

IV. CONCLUSION

This study successfully demonstrated that an HRF model can improve threat hunting on the Wazuh platform by dramatically lowering both MTTD and MTTR. The HRF model used strong preprocessing methods, such as Min-Max normalization, entropy-based data reduction, and PCC-based feature selection to achieve 99.65% and 99.12% accuracy on the ToN_IoT and CSE-CIC-IDS2018 datasets, outperforming the hybrid GRU-BiLSTM baseline for the latter. The model's incredible computational efficiency, with an inference time of less than 1.1 ms per record, makes it perfect for use in real-time settings where low latency is very important.

DL models like GRU-BiLSTM have somewhat higher F1-scores, but they are quite expensive to run, which makes it hard for businesses to respond quickly. The proposed HRF model strikes a better balance between speed and accuracy. This means that alerts can be generated faster, and the system can work with Wazuh's active response system without any problems. This leads to 75% decrease in MTTD and 65% decrease in MTTR, which directly improves SOC efficiency. In addition, its built-in interpretability increases the confidence of analysts by showing them which elements are used to make predictions. Future work will strategically target three impactful areas: (i) enhancing online learning through analyst feedback loops, which will facilitate incremental model updates without the need for extensive retraining, thereby improving efficiency, (ii) implementing SHAP-based explainability to deliver clear and interpretable justifications for alerts, empowering SOC analysts to make informed decisions, and (iii) rigorously validating the framework against zero-day threats using robust datasets, such as UNSW-NB15 and CIC-IDS2023, as well as within cloud-native environments such as Kubernetes, to ensure scalability and effectiveness. Furthermore, future plans involve investigating lightweight alternatives to PCC, such as variance thresholding combined with permutation importance, to significantly reduce preprocessing latency and enhance overall performance.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support and laboratory facilities provided by the Ministry of Higher Education, Science, and Technology of Indonesia, Politeknik Negeri Malang, and its Research and Community Service Center. Special thanks are also extended to the Applied Informatics Laboratory for its technical infrastructure and academic support, which were essential to the success of this research.

REFERENCES

[1] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, Mar. 2021, Art. no. 18, https://doi.org/10.1186/s42400-021-00077-7.

[2] K. Doshi, Y. Yilmaz, and S. Uludag, "Timely Detection and Mitigation of Stealthy DDoS Attacks via IoT Networks," *IEEE Transactions on Dependable and Secure Computing*, pp. 1, 2021, https://doi.org/10.1109/TDSC.2021.3049942.

[3] I. A. Kandhro *et al.*, "Detection of Real-Time Malicious Intrusions and Attacks in IoT Empowered Cybersecurity Infrastructures," *IEEE Access*, vol. 11, pp. 9136–9148, 2023, https://doi.org/10.1109/ACCESS.2023.3238664.

[4] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, and M. Portmann, "Feature extraction for machine learning-based intrusion detection in IoT networks," *Digital Communications and Networks*, vol. 10, no. 1, pp. 205–216, Feb. 2024, https://doi.org/10.1016/j.dcan.2022.08.012.

[5] T. G. Nguyen, T. V. Phan, B. T. Nguyen, C. So-In, Z. A. Baig, and S. Sanguanpong, "SeArch: A Collaborative and Intelligent NIDS Architecture for SDN-Based Cloud IoT Networks," *IEEE Access*, vol. 7, pp. 107678–107694, 2019, https://doi.org/10.1109/ACCESS.2019.2932438.

[6] G. Kocher and G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges," *Soft Computing*, vol. 25, no. 15, pp. 9731–9763, Aug. 2021, https://doi.org/10.1007/s00500-021-05893-0.

[7] A. A. Hagar and B. W. Gawali, "Deep Learning for Improving Attack Detection System Using CSE-CICIDS2018," *Neuro Quantology*, vol. 20, no. 7, 2022.

[8] A. A. Ghani and S. A. Alasadi, "A Deep Learning Algorithm to Cybersecurity: Enhancing Intrusion Detection with a Hybrid GRU and BiLSTM Model," *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 23605–23612, June 2025, https://doi.org/10.48084/etasr.10666.

[9] S. Jayalaxmi and S. Siddharth, "Intrusion Detection System For IOT," in *Futuristic Trends in IOT Volume 3 Book 5*, Iterative International Publisher, Selfypage Developers Pvt Ltd, 2024, pp. 63–81.

[10] M. L. Mutleg, A. M. Mahmood, and M. M. J. Al-Nayar, "Deep Learning Based Intrusion Detection System of IoT Technology: Accuracy Versus Computational Complexity," *International Journal of Safety and Security Engineering*, vol. 14, no. 5, pp. 1547–1558, Oct. 2024, https://doi.org/10.18280/ijsse.140522.

[11] J. Li, H. Chen, M. O. Shahizan, and L. M. Yusuf, "Enhancing IoT security: A comparative study of feature reduction techniques for intrusion detection system," *Intelligent Systems with Applications*, vol. 23, Sept. 2024, Art. no. 200407, https://doi.org/10.1016/j.iswa.2024.200407.

[12] K. Hu, "Intrusion detection using machine learning methods," in *International Conference on Electronic Information Engineering and Computer Technology (EIECT 2021)*, Kunming, China, Dec. 2021, Art. no. 67, https://doi.org/10.1117/12.2624897.

[13] K. Razikin and B. Soewito, "Cybersecurity decision support model to designing information technology security system based on risk analysis and cybersecurity framework," *Egyptian Informatics Journal*, vol. 23, no. 3, pp. 383–404, Sept. 2022, https://doi.org/10.1016/j.eij.2022.03.001.

[14] B. Nour, M. Pourzandi, and M. Debbabi, "A Survey on Threat Hunting in Enterprise Networks," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2299–2324, 2023, https://doi.org/10.1109/COMST.2023.3299519.

[15] I. Sharafaldin, A. H. 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*, Funchal, Madeira, Portugal, 2018, pp. 108–116, https://doi.org/10.5220/0006639801080116.

[16] M. Rodríguez, Á. Alesanco, L. Mehavilla, and J. García, "Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection," *Sensors*, vol. 22, no. 23, Nov. 2022, Art. no. 9326, https://doi.org/10.3390/s22239326.

[17] 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, pp. 165130–165150, 2020, https://doi.org/10.1109/ACCESS.2020.3022862.

[18] 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, Sept. 2021, Art. no. 102994, https://doi.org/10.1016/j.scs.2021.102994.

[19] 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, Jan. 2022, https://doi.org/10.1109/JIOT.2021.3085194.

[20] 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, Sept. 2021, Art. no. 103041, https://doi.org/10.1016/j.scs.2021.103041.

[21] 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)*, Guangzhou, China, Sept. 2020, pp. 727–735, https://doi.org/10.1109/TrustCom50675.2020.00100.

[22] 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, Dec. 2020, pp. 848–855, https://doi.org/10.1109/TrustCom50675.2020.00114.

[23] N. Moustafa, "A Systemic IoT-Fog-Cloud Architecture for Big-Data Analytics and Cyber Security Systems: A Review of Fog Computing." arXiv, 2019, https://doi.org/10.48550/ARXIV.1906.01055.

[24] N. Moustafa, "New Generations of Internet of Things Datasets for Cybersecurity Applications based Machine Learning: TON_IoT Datasets." UNSW Sydney, 2019, https://doi.org/10.26190/5D7AC9BFE8487.

[25] M. Templ, "Enhancing Precision in Large-Scale Data Analysis: An Innovative Robust Imputation Algorithm for Managing Outliers and Missing Values," *Mathematics*, vol. 11, no. 12, June 2023, Art. no. 2729, https://doi.org/10.3390/math11122729.

[26] L. D. Manocchio, S. Layeghy, M. Gallagher, and M. Portmann, "An empirical evaluation of preprocessing methods for machine learning based network intrusion detection systems," *Engineering Applications of Artificial Intelligence*, vol. 158, Oct. 2025, Art. no. 111289, https://doi.org/10.1016/j.engappai.2025.111289.

[27] W. Wu *et al.*, "Sliding Window Optimized Information Entropy Analysis Method for Intrusion Detection on In-Vehicle Networks," *IEEE Access*, vol. 6, pp. 45233–45245, 2018, https://doi.org/10.1109/ACCESS.2018.2865169.

[28] A. A. Tawil, L. Almazaydeh, D. Qawasmeh, B. Qawasmeh, M. Alshinwan, and K. Elleithy, "Comparative Analysis of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT," *Computers, Materials & Continua*, vol. 81, no. 2, pp. 3395–3412, 2024, https://doi.org/10.32604/cmc.2024.057279.

[29] I. M. Nasir *et al.*, "Pearson Correlation-Based Feature Selection for Document Classification Using Balanced Training," *Sensors*, vol. 20, no. 23, Nov. 2020, Art. no. 6793, https://doi.org/10.3390/s20236793.

[30] A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. Al-Qaness, and S. Lu, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," *IEEE Access*, vol. 9, pp. 123448–123464, 2021, https://doi.org/10.1109/ACCESS.2021.3109081.

[31] K. Yang *et al.*, "NIDS-CNNRF integrating CNN and random forest for efficient network intrusion detection model," *Internet of Things*, vol. 32, July 2025, Art. no. 101607, https://doi.org/10.1016/j.iot.2025.101607.

[32] B. Alhijawi, S. Fraihat, and A. Awajan, "Multi-factor ranking method for trading-off accuracy, diversity, novelty, and coverage of recommender systems," *International Journal of Information Technology*, vol. 15, no. 3, pp. 1427–1433, Mar. 2023, https://doi.org/10.1007/s41870-023-01158-1.

[33] "A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)," *Registry of Open Data on AWS*, [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018/

[34] "The TON_IoT Datasets," *UNSW Canberra*, [Online]. Available: available: https://research.unsw.edu.au/projects/toniot-datasets.