Experimental Analysis of Random Forest-Based Classification for Adaptive Network Attack Detection

*Hillman Akhyar Damanik[1] Merry Anggraeni[2]

Information Technology, Budi Luhur University
Jl. Ciledug Raya, RT.10/RW.2, Petukangan Utara, Kec. Pesanggrahan, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta, Indonesia
hilladamanik@gmail.com

Abstract. The evolution of modern cyberattacks has introduced increasingly adaptive and dynamic behaviors that challenge conventional security mechanisms. Adaptive attacks employ strategies such as randomized timing intervals, unconventional source ports, and multi-vector techniques, allowing them to bypass traditional signature-based Intrusion Detection Systems (IDS). As a result, static rule-based detection approaches often struggle to recognize evolving attack patterns, highlighting the need for machine learning-based solutions capable of analyzing behavioral characteristics in network traffic. This study presents an experimental evaluation of the Random Forest for adaptive cyberattack classification using network traffic data. The dataset was generated through an enterprise network simulation consisting of four target systems, including two routers and two Ubuntu servers, alongside ten distributed attacker IP addresses. A total of 5,005 network log entries were collected over a seven-day observation period and categorized into six classes: adaptive brute force, unconventional denial-of-service, network scanning, anomalous login behavior, multi-vector attacks, and normal traffic. Following data preprocessing, twelve relevant features were selected to represent traffic behavior and flow characteristics. The dataset was divided using an 80:20 split for training and testing purposes. Experimental results indicate that the Random Forest model achieved an overall accuracy of 94.1%, with precision, recall, and F1-score values of 94%. Class-level analysis demonstrated perfect detection performance for several attack categories, while behavior-driven attacks such as adaptive brute force and anomalous login patterns exhibited lower classification accuracy due to their subtle characteristics. Feature importance analysis further revealed that anomaly-related metrics and traffic volume attributes played a significant role in attack detection.

Keywords: Random Forest, Adaptive Cyber Attacks, Intrusion Detection System, Suricata, CHR

1 Introduction

The advancement of information and communication technology has increased the complexity of modern network infrastructures, while simultaneously enabling more adaptive and evasive cyberattack behaviors (Damanik & Anggraeni, 2025) (Damanik & Anggraeni, 2024). Contemporary attacks are no longer static instead, they are designed to mimic legitimate traffic patterns in order bypass traditional rule-based and signature-based Intrusion Detection Systems (IDS) (Badár et al., 2024) (Larriva-Novo

N. A. Ishak et al. (eds.), *Proceedings of the International Conference on Cross- Disciplinary Academic Research 2025 - Track 1 Advances in Computing, Electronics, Engineering, and Mathematics (ICAR-T1 2025)*, Advances in Engineering Research 296,
https://doi.org/10.2991/978-94-6239-636-4_11

et al., 2020) (Chinnasamy et al., 2025). This evolution significantly reduces the effectiveness of conventional detection mechanisms. In enterprise network environments, adaptive attacks may appear as randomized brute force attempts, the use of uncommon ports, irregular scanning activities, or coordinated multi-vector attack strategies (Larriva-Novo et al., 2020) (Villalba et al., 2022) (Vedavyass et al., 2025) (Ahmad et al., 2021). Such characteristics pose serious challenges for static IDS approaches, which rely heavily on predefined rules and fixed signatures, leading to delayed detection and increased security risks.

The study conducted by Pai et al. (2021) focuses on a comparative analysis of machine learning algorithms for network attack classification using the NSL-KDD dataset. The research emphasizes the selection of suitable algorithms; how-ever, it remains limited to static benchmark data and does not fully represent real operational network conditions (Pai et al., 2021). The work by Wali et al. (2025) proposes a Random Forest–based intrusion detection system enhanced with Ex-plainable AI and adversarial defense mechanisms. Although the proposed architecture is sophisticated and resilient, the study primarily addresses model robust-ness against adversarial attacks and does not empirically evaluate the application of Random Forest using operational IDS log data (Wali et al., 2025). The research by Kumar et al. (2025) evaluates the performance of supervised and unsupervised learning algorithms for anomaly detection in critical infrastructure environments dataset. This study mainly focuses on time-series data analysis within industrial systems rather than enterprise network traffic derived from IDS logs (Kumar & Gutierrez, 2025). The study conducted by Tesfahun et al. (2013) applies Random Forest classification to the NSL-KDD dataset by ad-dressing class imbalance through SMOTE and performing feature selection using information gain. While the results demonstrate improved classification performance, the study still relies on benchmark datasets and does not incorporate adaptive attack data from real network environments (Tesfahun & Bhaskari, 2013). The research by Joseph et al. (2025) reviews various machine learning and deep learning approaches for intrusion detection in Internet of Things (IoT) environments. This work is conceptual in nature and focuses on literature analysis, emphasizing IoT-related challenges without presenting experiments based on IDS log data or empirical evaluations in enterprise networks (Joseph et al., 2025). Meanwhile, Mahmood et al. (2024) examines the integration of machine learning techniques with multi-factor authentication (MFA) to enhance network security. The study places greater emphasis on security architecture and system policy considerations rather than on experimental evaluation of intrusion detection model performance using IDS-based data (Mahmood et al., 2024).

Based on the reviewed studies, it can be concluded that most prior research primarily focuses on benchmark datasets, simulated environments, or conceptual analyses, and has not empirically evaluated the performance of classification algorithms using operational IDS log data in enterprise networks. In addition, adaptive attacks that exploit dynamic patterns and varying network traffic behaviors remain relatively underexplored when assessed using real-world data. There-fore, this study introduces novelty by conducting an experimental analysis of the Random Forest algorithm using Suricata IDS log data collected directly from an enterprise-like network topology. This research

emphasizes quantitative evaluation of model performance in detecting adaptive network attacks, thereby providing more empirically grounded contributions to the practical deployment of intrusion detection systems in real operational environments.

2 Method

This study applies the Random Forest algorithm to support adaptive cyberattack detection within an Intrusion Detection System (IDS) framework. The research methodology is structured to assess the capability of Random Forest in classify-ing multiple types of network attacks based on traffic patterns obtained from a simulated network infrastructure using virtual machines and the GNS3 emulator. The dataset is constructed in a controlled manner to represent cyberattack conditions that closely resemble real-world operational scenarios. The evaluation focuses on the model's ability to recognize and categorize adaptive threats that exhibit behavioral characteristics distinct from conventional attack patterns. The dataset consists of 5,005 network log entries collected over a seven-day period, covering six attack categories with a distribution that reflects realistic network environments. The proposed methodology follows a standard machine learning workflow commonly adopted in cybersecurity analysis, including data collection, preprocessing, labeling, model training, and performance evaluation. Each stage is systematically designed to ensure the reliability of the results and to support reproducibility in future research.

2.1 System Architecture and Network Topology

This study employs a network topology designed to simulate an infrastructure with multiple entry points and heterogeneous target systems. The topology is implemented using the GNS3 emulator and the hypervisor proxmox to represent realistic operational network conditions, where attackers may exploit different access paths and targets, as illustrated (see Fig. 1).

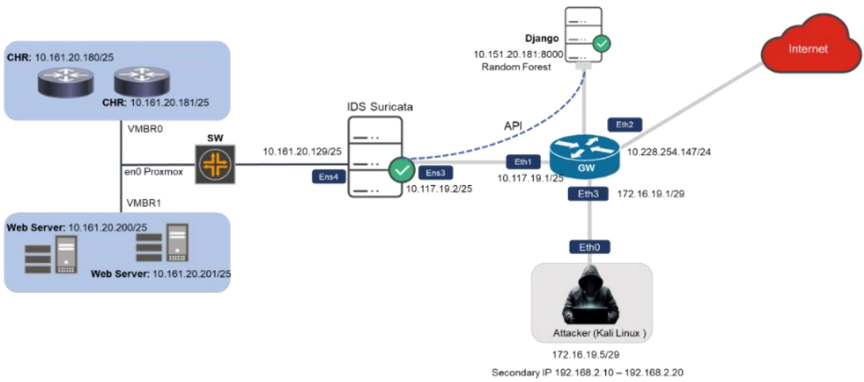

Fig. 1. System Architecture and Network Topology

The system architecture in this research consists of three integrated components. First, the attack network utilizes the IP address range from 192.168.2.10 to 192.168.2.20, comprising ten attack sources running on Kali Linux platforms. This configuration is intended to simulate multi-source attack scenarios, reflecting the characteristics of distributed attacks commonly observed in real-world environments. Second, the target infrastructure includes two CHR routers with IP addresses 10.161.20.180 and 10.161.20.181, along with two Ubuntu servers located at 10.161.20.200 and 10.161.20.201. These devices are selected to represent diverse network roles and service types typically found in enterprise networks. Third, the IDS monitoring component employs Suricata, deployed at IP address 10.161.20.129/25, to capture and detect network traffic activities, while a Django-based analytics platform operates at 10.151.20.181:8000 to process, analyze, and present the detection results. Data storage is managed using a SQLite database that records IDS logs and machine learning analysis outputs. This architecture is intentionally designed to support the generation of diverse traffic patterns, enable the evaluation of multiple adaptive attack scenarios, and provide an effective environment for monitoring and data-driven intrusion detection analysis.

2.2 Implementation of Multi-Source Attack Simulation

To represent adaptive attack characteristics that closely resemble real-world conditions, this study implements a multi-source attack simulation using a script named attack_scenarios.sh executed on a Kali Linux platform. The script is de-signed to launch coordinated attacks from multiple source IP addresses, effectively replicating the behavior of distributed attack scenarios commonly observed in operational networks. Each attack instance applies a random source IP selection technique to generate unpredictable traffic patterns that are difficult to trace. This approach enables the creation of adaptive and non-deterministic at-tack behaviors, providing a suitable foundation for evaluating the capability of machine learning–based IDS models to detect dynamic and evolving cyber threats.

```
Source IP Array:
SOURCE_IP=("192.168.2.10" "192.168.2.11" "192.168.2.12"
"192.168.2.13" "192.168.2.14" "192.168.2.15" "192.168.2.16"
"192.168.2.17" "192.168.2.18" "192.168.2.19" "192.168.2.20")
Target:
TARGET_UBUNTU=("10.161.20.200" "10.161.20.201")
TARGET_CHR=("10.161.20.180" "10.161.20.181")
ALL_TARGETS=("10.161.20.200" "10.161.20.201" "10.161.20.180"
"10.161.20.181")
# Multi-source IP selection SOURCE_IP=${SOURCE_IPS[$RANDOM %
${#SOURCE_IPS[@]}]} TARGET=${ALL_TARGETS[$RANDOM %
${#ALL_TARGETS[@]}]}
```

2.3 Adaptive Attack Scenario

This study implements six types of cyberattacks designed to reflect the modern threat landscape, with particular emphasis on the adaptive characteristics of each scenario.

One attack type is the adaptive brute force attack, accounting for ap-proximately 15% of the dataset. This attack is characterized by random time de-lays ranging from 1 to 30 seconds to evade rate-limiting mechanisms, simultaneous targeting of multiple services such as SSH on Ubuntu servers and HTTP on CHR routers, and dynamic username selection from commonly used credential lists. The next category, unconventional DoS attacks, represents approximately 10% of the dataset. These attacks employ uncommon source ports, including 1234, 8888, 9999, 12345, 31337, and 54321, combined with UDP and TCP SYN flooding techniques and periodic target shifting across distributed sources. Net-work scanning scenarios using Nmap contribute around 20% of the dataset and are generated through comprehensive port scans with realistic delays targeting multiple systems simultaneously. Lastly, anomalous login patterns account for approximately 7.5% of the dataset and are simulated using time-based usernames and extended login intervals, particularly during nighttime hours and weekends, to emulate stealthy, behavior-based attack patterns.

2.4 System Implementation Architecture

This study develops an integrated system using the Django framework to support data processing, analysis, and visualization. The system architecture is designed to enable real-time attack detection and machine learning model evaluation. At the data layer (models.py), key tables include SuricataLog for storing 5,005 net-work logs, MLModel for recording performance metrics, and AttackPattern and AnomalyDetection for storing analytical results. The machine learning layer (ml_services.py) implements the Random Forest algorithm through the Random-ForestAnomalyDetector class, which handles feature engineering, model training with hyperparameter optimization, and real-time prediction. The API layer (views.py) provides endpoints for accessing dataset summaries, model infor-mation, and training processes. Finally, the presentation layer (dashboard.html) delivers interactive visualizations of model performance and attack distributions, supporting operational monitoring of the IDS.

2.5 Preprocessing

The preprocessing stage serves as a crucial foundation in implementing Random Forest for IDS. This process involves transforming raw log data from Suricata EVE JSON format into a feature vector suitable for training a machine learning algorithm.

2.5.1 Data Collection and Parsing

Data collection was carried out using EVE logs in JSON format generated by the Suricata IDS, total 5,005 entries gathered over a seven-day period. These logs are hierarchically structured, containing nested objects that include flow information, alerts, and metadata. The raw data underwent a parsing process, beginning with the conversion of raw JSON into a structured data frame, followed by transformation into a feature matrix ready for subsequent analysis. Table 1 illustrates the parsing process, which involved extracting critical fields from the complex structure of the JSON logs.

Table 1 The parsing process involves extracting critical fields from a complex JSON structure:

Network Features Layer		
	Source IP (src_ip)	Attacker IP address
	Destination IP (dest_ip)	Target IP address
	Source Port (src_port)	Source port of the connection
	Destination Port (dest_port)	Destination service port
	Protocol (proto)	(TCP/UDP/ICMP)
Temporal Features	Timestamp	Event occurrence time with second-level resolution
	Time-based patterns	Extracted hour, day, weekend indicators
Flow Metrics	Bytes to server (flow_bytes_toserver)	Volume of data sent to the target
	Bytes to client (flow_bytes_toclient)	Volume of data sent in response
	Packets to server (flow_pkts_toserver)	Number of packets sent to the target
	Packets to client (flow_pkts_toclient)	Number of response packets
Alert Information	Alert signature	IDS signature that was triggered
	Alert category	Attack category
	Alert severity	Priority level

2.5.2 Feature Engineering

During the feature engineering stage, derived features were created from network traffic logs to enhance the model ability to distinguish between normal activity and suspicious behavior. These features were grouped based on traffic volume characteristics, behavioral patterns, and port usage. For traffic volume, features such as total_bytes and total_packets were calculated by combining data from both server and client directions. Ratio-based features like bytes_ratio and packets_ratio were included to assess the balance of bidirectional traffic. Behavioral features, including avg_packet_size and traffic_asymmetry, were designed to capture average packet size and the asymmetry of traffic flow. Meanwhile, port-based features were used to identify whether the source employed uncommon ports (is_unusual_port) and whether the destination corresponded to standard services (is_common_service).

2.5.3 Data Cleaning and Validation

The data cleaning and validation process involved several critical steps. First, columns containing null or incomplete values were identified and handled using context-aware approaches tailored to the nature of the data and the type of attack. Records deemed irreparable accounting for less than 1% of the dataset were removed. Next, outliers were detected through statistical analysis and re-evaluated based on domain knowledge in network security. Extreme values that were consistent with known attack patterns were retained. Finally, data type consistency was ensured by standardizing IP address formats, converting timestamps to UTC, and casting numerical values to appropriate data types.

2.5.4 Feature Scaling and Normalization

Normalization was applied to numerical features using StandardScaler to ensure consistent value scaling, particularly for features with varying distributions. This process is demonstrated in the following script.

```
from sklearn.preprocessing import StandardScaler
numerical_features = ['total_bytes', 'total_packets', 'bytes_ratio',
'packets_ratio', 'avg_packet_size']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X[numerical_features])
```

In addition, categorical features were processed using appropriate encoding techniques. Label encoding was applied to ordinal features such as alert_severity, one-hot encoding was used for nominal features like protocol, and binary encoding was implemented for boolean features such as is_unusual_port. This approach ensures that all features are optimally formatted for machine learning model processing.

2.5.5 Feature Selection

Correlation analysis was conducted to identify features with strong interdependencies. Redundant features were removed to reduce data complexity, while those deemed critical from a cybersecurity perspective were retained. The selection of key features also considered their relevance in detecting attacks, particularly behavioral patterns indicative of adaptive threats, while maintaining a balance between network-level and application-level attributes. Upon completing the preprocessing phase, a set of 12 primary features was finalized, including anomaly_score, total_bytes, bytes_ratio, and avg_packet_size. The final dataset comprised 5,005 entries, each with 12 features, labels representing six attack categories, and a normalized, balanced data distribution.

2.6 Labelling

Labeling is a crucial step in supervised learning, as it directly influences the quality and performance of the Random Forest model. In this study, a structured labeling approach was employed to classify the 5,005 preprocessed log entries into six distinct attack categories.

2.6.1 Ground Truth Labeling Strategy

During the data labeling stage, a ground truth strategy was applied to ensure that each log entry was accurately categorized based on the identified attack patterns. Following an in-depth analysis of the processed data, six primary labels were defined to represent various types of network activity, including both adaptive attacks and normal traffic. Each label was assigned based on a combination of IDS signatures, technical characteristics such as ports used, data volume, packet count, and behavioral traffic patterns. These categories form the basis for classification model evaluation, making accurate labeling essential to ensure the validity of the training results.

2.6.2 Multi-Class Classification

This phase defines the multi-class classification structure used to detect and distinguish between different types of cyberattacks. Label implementation was carried out using a label mapping scheme that assigned each attack type and normal traffic category to a corresponding numerical value for model training purposes. To streamline the classification process, the predefined attack categories were encoded into numeric labels. Table 2 below presents the label mapping scheme used in this study.

Table 2 Label Mapping Scheme

Label Category	Label Code
Adaptive Brute Force	0
Unconventional DoS	1
Nmap Network Scanning	2
Anomalous Login Patterns	3
Multi-Vector Coordinated Attack	4
Normal Traffic	5

2.7 Random Forest Training

The training phase constitutes the core implementation of the Random Forest algorithm for adaptive cyberattacks detection. This process involves dataset splitting, hyperparameter optimization, model training, and validation to produce a robust and accurate classifier.

2.7.1 Dataset Splitting

The dataset splitting strategy for the training and testing phases is designed to represent real-world network conditions. Stratified Random Sampling is employed to maintain balanced class proportions across both training and testing sets. The initial dataset D consists of 5,005 samples defined as feature-label pairs.

$$D = \{(x1, y1), (x2, y2), \ldots , (x5005, y5005)\} \quad (1)$$

- D is the dataset consisting of 5,005 data sample pairs
- $X_i \in R^{12}$ denotes the 12-dimensional feature vector of *i* data sample, representing the characteristics of network traffic.
- $Y_i \in \{0,1,2,3,4,5\}$ is the class label of the *i* data sample, which has been numerically encoded (e.g., 0 for normal traffic, 1 for Nmap scanning).
- $i \in \{1, 2, ...,5005\}$ indicates the index of each data pair in dataset D.

The dataset, consisting of 5,005 samples, is divided into two subsets: training data and testing data, with a ratio of $|D_{train}| : |D_{test}| = 0{,}8 : 0{,}2$. Based on this ratio, 4,004 samples are used to train the model, while the remaining 1,001 samples are reserved for evaluating the model performance. This data split is performed using stratified sampling, which ensures that the class distribution remains balanced across both the training and testing sets. The purpose of this partitioning is to ensure that the model learns from a representative dataset while being fairly evaluated on unseen data. Stratified sampling ensures that each class label is proportionally represented in both subsets. For example, if there are six distinct class labels {0, 1, 2, 3, 4, 5}, the data splitting process is carried out in such a way that each class maintains an equivalent distribution in both the training and testing subsets. After the above process is completed for each class $k \in$, the complete training and testing datasets are obtained by concatenating the respective subsets from each class.

$$D_{train} \bigcup_{k=0}^{5} D_{train,k} \quad D_{test} = \bigcup_{k=0}^{5} D_{test,k} \tag{2}$$

The data splitting process was performed using the stratified sampling technique, which ensures that the proportion of each class is preserved in both the training and testing datasets. In this context, all data samples were categorized based on attack or traffic type into six classes: *normal_traffic*, *nmap_scanning*, *adaptive_brute_force*, *unconventional_dos*, *anomalous_login*, and *multi_vector*. Each class was then divided into 80% for training and 20% for testing, ensuring the class distribution remains consistent across both subsets. Table 3 presents the number of samples used for training and testing for each respective class.

Table 3: Class Proportions for Training and Testing Data

Attack Type	Total Sample	Training (80%)	Testing (20%)
normal_traffic	2.128	1.702	426
nmap_scanning	1.001	801	200
adaptive_brute_force	749	599	150
unconventional_dos	497	398	99
anomalous_login	378	302	76
multi_vector	252	202	50
Total	5.005	4.004	1.001

2.8 Performance Evaluation Metrics

In multi-class classification, model performance is assessed based on the number of correct and incorrect predictions for each class. For each class $k \in \{0,1,2,3,4,5$, the following components are calculated:

- True Positive (TP_k): The number of instances that truly belong to class k and are also predicted as class k. $TP_k = |\{i: y_i = k \wedge y\hat{}i = k\}|$

- False Positive (FP_k): The number of instances that do not actually belong to class k, but are incorrectly predicted as class k. $FP_k = |\{i: y_i \neq k \wedge y\hat{}i = k\}|$
- False Negative (FN_k): The number of instances that should belong to class k, but are incorrectly predicted as another class. $FN_k = |\{i: y_i = k \wedge y^i \neq k\}|$
- True Negative (TN_k): The number of instances that do not belong to class k and are also not predicted as class k.

3 Result and Discussion

This chapter reports the experimental results obtained from the implementation of a Random Forest algorithm for detecting adaptive cyberattacks within an Intrusion Detection System. The evaluation is based on network traffic data collected over a seven-day observation period, consisting of 5,005 log records representing six attack-related categories. The discussion covers dataset characteristics, attack distribution patterns, temporal behavior, and classification performance across different adaptive threat types.

3.1 Statistic Dataset

The dataset was collected through an integrated system utilizing a web-based application developed with the Django framework. This system was designed to record network logs in real-time and store them within an internal database for subsequent classification analysis. During the 7-day observation period, the system operated reliably and produced a representative dataset suitable for training and testing the attack detection model. The dashboard interface (see Fig. 2) provides a summary of the collected data. The displayed information reflects processed Suricata EVE JSON logs that were classified using the Random Forest algorithm.

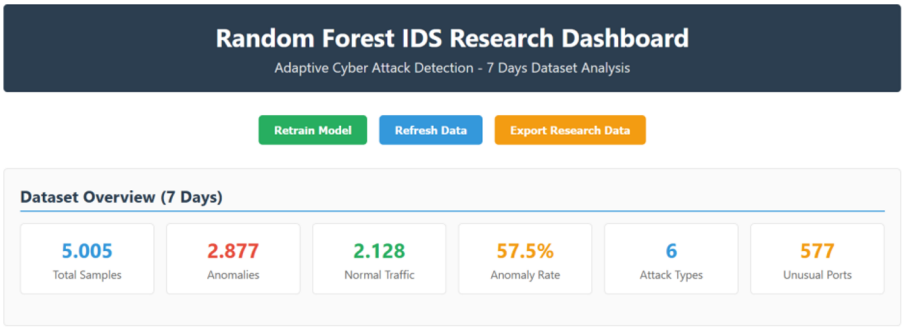

Fig. 2. Dataset Statistic

A total of 5,005 network log entries were collected, consisting of 2,877 records (57.5%) classified as anomalous traffic and 2,128 records (42.5%) identified as normal activity. Classification was performed automatically by the Django backend, which was directly integrated with the trained model and executed inference in near real-time. Additionally, the analysis identified ten distinct source IP addresses, ranging from 192.168.2.10 to 192.168.2.20, indicating a distributed attack pattern. These attacks targeted 44 hosts, including two CHR routers and two Ubuntu servers, demonstrating variability in attack targets within the simulated network environment.

3.2 Attack Distribution Categories

The attack dataset utilized in this study reflects real-world conditions in modern cybersecurity landscapes. The types of attacks included were selected based on those most frequently encountered within network infrastructure environments. The composition of the data was designed to align with the actual frequency of attacks typically observed in real scenarios. The distribution of normal traffic logs, which account for 42.5% (2,128 entries) of the total dataset (see Fig.3). This proportion of legitimate traffic reflects regular and authorized communication within an enterprise network. Such a distribution is considered reasonable, as in most operational environments, benign traffic usually surpasses malicious activity. Normal traffic often utilizes common service ports such as HTTP (80), HTTPS (443), SSH (22), DNS (53), SMTP (25), and POP3 (110). Its communication pattern tends to be orderly, bidirectional, and lacks any anomalous behavior. The average of 304 entries per day indicates stable and consistent business operations. This traffic encompasses routine activities such as web access, email transactions, system administration, and inter-service communication. Simulated attack traffic involving Nmap scanning constitutes approximately 20.0% (1,001 entries) of the dataset, making it the second most prevalent category. This outcome suggests that attackers typically initiate their intrusions with reconnaissance efforts. An average of 143 scans per day reveals persistent attempts at network mapping, a behaviour commonly observed in targeted and continuous attack scenarios.

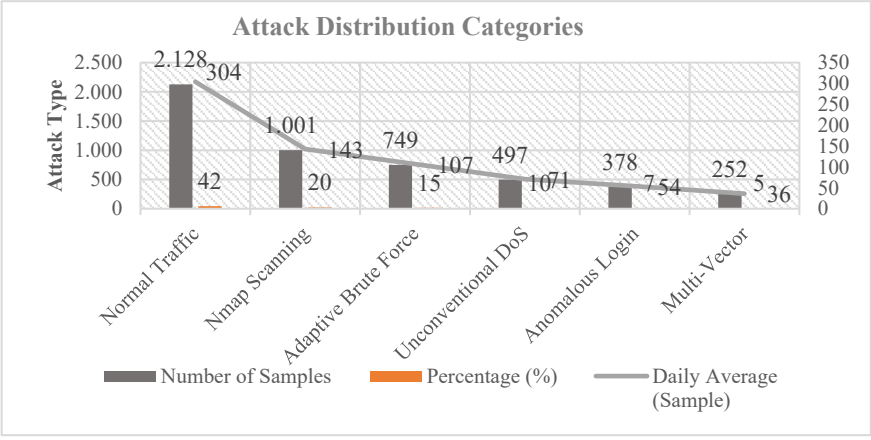

Fig. 3. Attack Distribution Categories

Adaptive Brute Force (15.0% – 749 samples) refers to brute force attacks averaging 107 attempts per day, primarily targeting authentication services such as SSH (port 22), RDP (port 3389), and FTP (port 21). Its adaptive nature is characterized use of randomized time intervals between attempts to evade automated access restriction mechanisms. This frequency is considered reasonable, as brute force attacks remain among the most prevalent threats. Attackers often prefer credential theft as a means of gaining initial access, rather than exploiting system vulnerabilities.

Unconventional DoS (10.0% – 497 samples) represents Denial of Service attacks originating from uncommon source ports such as 1234, 8888, 9999, 12345, 31337, and 54321, with an average of 71 attempts per day. The use of non-standard ports is intended to bypass typical security filters that detect attacks based on commonly used ports, while simultaneously generating traffic patterns that differ from legitimate high-volume applications.

Anomalous Login (7.5% – 378 samples) encompasses suspicious login activities occurring approximately 54 times per day. These attacks indicate infiltration attempts that are difficult to detect using traditional signature-based methods. The attacks typically target authentication ports such as SSH (port 22), with a temporal pattern that tends to occur predominantly during nighttime hours.

Multi-Vector Attacks (5.0% – 252 samples) represent a class of coordinated threats averaging 36 incidents per day and are considered the most sophisticated. These attacks simultaneously combine multiple vectors such as SSH, HTTP, and port scanning, reflecting the attacker's advanced capabilities and the allocation of significant resources for complex operations. Although they occur with the lowest frequency, this is justifiable given that multi-vector attacks require extensive coordination, resources, and technical expertise. The targeting of diverse ports (22, 80, 443, 3389) illustrates a comprehensive strategy aimed at maximizing the probability of success by leveraging multiple attack surfaces.

3.3 Temporal Distribution Analysis

Figure 4 presents the daily log distribution over a seven-day period, illustrating the balance between total log volume, normal traffic, and detected anomalous activity each day. The number of logs recorded per day ranges from 710 to 720, with daily anomaly counts remaining relatively stable, resulting in an anomaly rate between 56.3% and 58.3%. The consistency of these anomaly percentages indicates that the simulation process, executed via a traffic generator script, successfully maintained a controlled and systematic proportion between anomalous and normal traffic (see Fig. 4).

Throughout the seven-day simulation period, a total of 5,005 log entries were collected, comprising 2,877 classified as anomalies and 2,128 as normal traffic. This yields an average anomaly rate of 57.5%. Such proportions provide a strong foundation for training and evaluating the attack detection model, as the dataset offers a relatively balanced distribution that reflects realistic network traffic dynamics under controlled testing conditions.

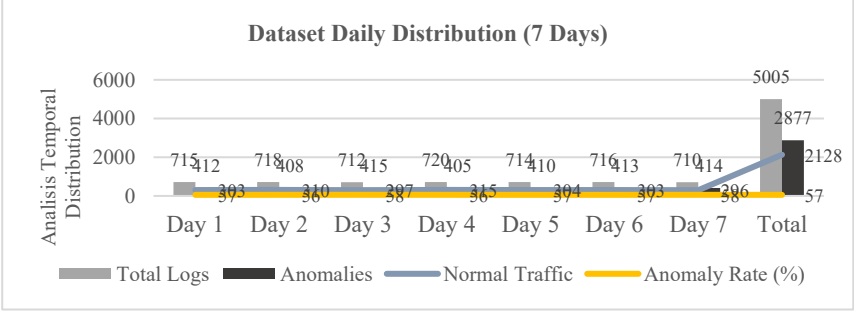

Fig. 4. Dataset Daily Distribution

3.4 Overall Performance Metric

The Random Forest model demonstrated excellent performance in detecting various types of attacks, particularly in scenarios involving adaptive threats. The evaluation was conducted on a test set comprising 1,001 samples, which had been previously separated from the training data to ensure the objectivity of the testing process. Illustrates the performance evaluation results of the Random Forest model (see Fig.5).

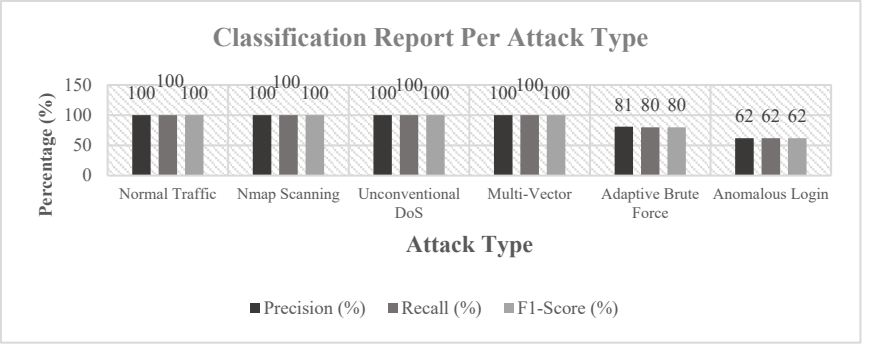

Fig. 5. Performance Metric Random Forest

The Random Forest model exhibited strong performance in identifying cyberattacks, based on evaluation results from a test dataset comprising 1,001 samples. Achieving an accuracy of 94.1%, the model successfully classified 942 instances correctly an impressive outcome within the domain of cybersecurity. The weighted average precision of 94% indicates that the majority of attack predictions made by the model were accurate, effectively reducing the likelihood of false positives. Similarly, the recall value of 94% reflects the model capability to detect most actual threats, which is critical in preventing missed detections. Furthermore, the F1-Score of 94% highlights a well-balanced trade-off between precision and recall, underscoring the model overall robustness in threat classification.

3.5 Classification Report Per Attack Type

The evaluation across individual attack categories demonstrates the strong capability of the Random Forest model in accurately recognizing certain types of intrusions. As depicted in Figure 6, categories like Normal Traffic, Nmap Scanning, Unconventional DoS, and Multi-Vector attacks achieved flawless classification results, with each reporting 100% precision, recall, and F1-score. These results suggest that the traffic patterns linked to these categories are highly distinctive, enabling the model to detect them without error. Particular, unconventional DoS attacks stood out due to their use of rare port numbers such as 1234, 8888, 9999, 12345, 31337, and 54321—which made them easier for the model to identify as abnormal behaviour.

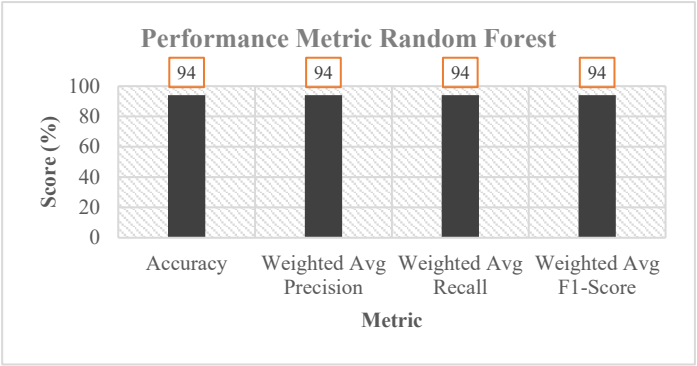

Fig. 6. Classification report per attack type

For adaptive brute force attacks, the model still demonstrated good performance despite the randomized timing scheme from the simulation results (ranging from 1 to 30 seconds), with an F1-score of 80%. This proves that even though the timing pattern is inconsistent, the model can still recognize the characteristics of the attack. Meanwhile, anomalous login became the category with the lowest performance, with an F1-score of 62%.

3.6 Adaptive Attack Detection Capability Analysis

3.6.1 Unusual Ports Detection (DoS)

The classification model demonstrated excellent capability in detecting DoS attacks that utilized unusual ports. All 497 samples originating from port variations such as 8888, 9999, 3133, 12345, 54321, and 1234 were successfully identified with a 100% detection rate. The attack traffic volume varied between 50KB and 200KB, depending on the port used. Table 4 presents the variation in traffic and the differing number of samples the model consistently exhibited perfect performance in all cases, indicating the system effectiveness in recognizing adaptive, signature-based attack techniques.

Table 4. Unusual Ports Detection (DoS)

Unusual Port	Count in Dataset	Detection Rate	Traffic Volume
8888	87 samples	100%	70KB–170KB
9999	164 samples	100%	50KB–200KB
3133	70 samples	100%	80KB–150KB
12345	116 samples	100%	60KB–180KB
54321	54 samples	100%	90KB–140KB
1234	6 samples	100%	75KB–160KB
Total	497 samples	100%	Variable

3.6.2 Multi-Vector Detection Results

The system capability to identify multi-vector attacks was evaluated through comprehensive analysis of 252 attack samples involving combined attack vectors such as simultaneous SSH brute force attempts, HTTP flooding, and port scanning activities. The results demonstrated flawless detection performance (100%) across all testing parameters. Table 5 presents the outcomes where the system successfully identified coordination patterns between different attack vectors, recognized diverse traffic patterns ranging from 1KB to 28KB in volume, and detected distributed source IP addresses spanning between 5 to 10 IP addresses. These findings indicate that the random forest model possesses superior abilities in recognizing coordinated attack patterns despite their high complexity levels.

Table 5. Multi-vector Detection Results

Aspect	Performance	Sample Size	Characteristics
Coordination Detection	100%	252 samples	Perfect identification
Multiple Attack Types	100%	SSH + HTTP + Scan	Successfully detected
Traffic Volume Range	100%	1KB–28KB	Variable pattern recognition
Source IP Diversity	100%	5–11 sources	Distributed attack detection

3.6.3 Brute Force Adaptive Timing

The Random Forest model underwent testing against brute force attack patterns that employed adaptive approaches through timing variation strategies (timing randomization). Table 6 displays the testing results which encompassed random intervals between 1 to 30 seconds, utilization of various authentication ports (including 22, 3389, and 21), along with low data volumes and target distribution across multiple systems. The outcomes revealed detection rates ranging from 80% to 85%, depending on the attack characteristics. The model successfully identified majority of brute force patterns even when attacks were executed using disguised timing and traffic volume methods. This demonstrates that Random Forest possesses resilience against attacks employing adaptive timing strategies, although the challenge level remains moderately significant.

Table 6 Brute Force Adaptive Timing

Timing Pattern	Detection Rate	Interval Range
Random Intervals (1–30s)	80–81%	Variable
Authentication Ports	85%	22, 3389, 21
Low Volume Pattern	83%	200–1,500 bytes
Multi-Target Approach	82%	4 target systems

4 Conclusion

The research findings on implementing Random Forest algorithms for adaptive cyberattack detection in Intrusion Detection Systems demonstrate exceptional effectiveness in

identifying and categorizing adaptive cyber threats with outstanding performance metrics. The developed model achieved a 94.1% accuracy rate when tested on a dataset containing 1,001 samples, while maintaining weighted average precision, recall, and F1-score values of 94% each. These results illustrate Random Forest's capability to address limitations inherent in traditional IDS frameworks that rely heavily on signature-based detection methodologies. Complete detection accuracy (100%) was obtained for legitimate network traffic, nmap reconnaissance activities, non-conventional denial-of-service attacks, and coordinated multi-vector assault patterns. Feature importance analysis revealed that anomaly_score (18.26%), total_bytes (18.11%), and flow_bytes_to-client (13.01%) constituted the three most influential variables in the classification process. The combination of the top five features contributed 67.70% to the decision-making mechanism, indicating that a hybrid approach combining behavioral analysis (anomaly_score) with quantitative traffic measurements (volume-based features) produces optimal results for adaptive attack identification. The dataset compilation through seven-day simulation procedures generated 5,005 network log samples with realistic distribution patterns: legitimate traffic (42.5%), nmap reconnaissance (20%), adaptive brute force attempts (15%), non-conventional denial-of-service (10%), suspicious authentication activities (7.5%), and coordinated multi-vector attacks (5%). This distribution pattern reflects contemporary threat environments where reconnaissance operations and credential-based intrusions predominate, while sophisticated coordinated attacks occur with lower frequency rates.

5 Acknowledgements

The author would like to express sincere gratitude to the Directorate of Research and Community Service (DRPM) of Universitas Budi Luhur for the support and funding that made this research possible.

References

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Alta Frequenza*, *32*(1). https://doi.org/10.1002/ett.4150

Badár, J., Papaj, J., Chovanec, M., & Cavojský, M. (2024). Comparative Analysis of Intrusion Detection Systems and Neural Networks for Anomaly Detection in Network Security. *2024 International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 1–6. https://doi.org/10.1109/ICETA63795.2024.10850850

Chinnasamy, P., SivaKrishnaiah, C., Anjali, T., Kambalapelly, A., Rao, P. V., & Degala, D. P. (2025). Managing Network Security in IT Sector using the Suricata. *2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, 1721–1728. https://doi.org/10.1109/ICSSAS66150.2025.11080990

Damanik, H. A., & Anggraeni, M. (2024). Sistem Deteksi Intrusi Hybrid dan Mitigasi Kerentanan Infrastruktur Jaringan Menggunakan Teknik Active Response (XDR) Wazuh dan Suricata. *Jurnal Pekommas*, *9*(2), Article 2. https://doi.org/10.56873/jpkm.v9i2.5829

Damanik, H. A., & Anggraeni, M. (2025). Analisis dan Mitigasi Kerentanan DDoS pada Infrastuktur Jaringan dengan Teknik Hierarchical Clustering dan Firewall IPTables. *Jurnal Pekommas*, *10*(1), Article 1. https://doi.org/10.56873/jpkm.v9i1.5551

Joseph, J. E., Aleke, N. T., & Onyeanisi, O. P. (2025). Deep Learning Based Intrusion Detection System for Network Security in IoT System. *International Journal of Education, Management, and Technology*, *3*(1), 119–138. https://doi.org/10.58578/ijemt.v3i1.4539

Kumar, A., & Gutierrez, J. A. (2025). Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure. *Information*, *16*(7), 515. https://doi.org/10.3390/info16070515

Larriva-Novo, X. A., Vega-Barbas, M., Villagrá, V. A., & Sanz Rodrigo, M. (2020). Evaluation of Cybersecurity Data Set Characteristics for Their Applicability to Neural Networks Algorithms Detecting Cybersecurity Anomalies. *IEEE Access*, *8*, 9005–9014. https://doi.org/10.1109/ACCESS.2019.2963407

Mahmood, R. K., Mahameed, A. I., Lateef, N. Q., Jasim, H. M., Radhi, A. D., Ahmed, S. R., & Tupe-Waghmare, P. (2024). Optimizing Network Security with Machine Learning and Multi-Factor Authentication for Enhanced Intrusion Detection. *Journal of Robotics and Control (JRC)*, *5*(5), 1502–1524. https://doi.org/10.18196/jrc.v5i5.22508

Pai, V., Devidas, & Adesh, N. D. (2021). Comparative analysis of Machine Learning algorithms for Intrusion Detection. *IOP Conference Series: Materials Science and Engineering*, *1013*(1), 012038. https://doi.org/10.1088/1757-899X/1013/1/012038

Tesfahun, A., & Bhaskari, D. L. (2013). Intrusion Detection Using Random Forests Classifier with SMOTE and Feature Reduction. *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, 127–132. https://doi.org/10.1109/CUBE.2013.31

Vedavyass, N., Hari, N. S., Sasikala, T., & Nagarajan, G. (2025). Machine Learning-Driven Intrusion Detection System for Enhanced Accuracy and Scalability. *2025 International Conference on Intelligent and Cloud Computing (ICoICC)*, 1–5. https://doi.org/10.1109/ICoICC64033.2025.11052209

Villalba, D. A. M., Varon, D. F. M., Pórtela, F. G., & Triana, O. A. D. (2022). Intrusion Detection System (IDS) with anomaly-based detection and deep learning application. *2022 V Congreso Internacional En Inteligencia Ambiental, Ingeniería de Software y Salud Electrónica y Móvil (AmITIC)*, 1–4. https://doi.org/10.1109/AmITIC55733.2022.9941277

Wali, S., Farrukh, Y. A., & Khan, I. (2025). Explainable AI and Random Forest based reliable intrusion detection system. *Computers & Security*, *157*, 104542. https://doi.org/10.1016/j.cose.2025.104542