I.J. Wireless and Microwave Technologies, **2026, 5, 428-445**
Published Online on October 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.05.25

Real-Time Port Scanning Attack Detection Using Adaptive Entropy Analysis and Random Forest-based Hybrid Model

Maruf Tojiyev Ruzikulovich*
Department of Management Theory and Information Security, Samarkand State University named after Sharof Rashidov, Samarkand, Uzbekistan
E-mail: mrtojiyev6886@gmail.com
ORCID iD: https://orcid.org/0000-0002-9352-8650
*Corresponding Author

Bahtiyor Holmuhamedov Farkhodovich
Department of Uzbek Linguistics, Samarkand State University named after Sharof Rashidov, Samarkand, Uzbekistan
E-mail: bxolmuxamedov@mail.ru
ORCID iD: https://orcid.org/0000-0002-8546-8192

Sofiyaxon Usmonova Alimovna
Interfaculty Department of Languages, Kokand State University, Fergana, Uzbekistan
E-mail: sofiya.usmonova.72@gmail.com
ORCID iD: https://orcid.org/0009-0006-0804-4886

Jura Kuvandikov Tursunbayevich
Department of Computer Science and Programming, Jizzakh Branch of the National University of Uzbekistan named after Mirzo Ulugbek, Jizzakh, Uzbekistan
E-mail: jorakuvandikov1@gmail.com
ORCID iD: https://orcid.org/0009-0008-5205-9561

Received: April 3, 2026; Revised: June 26, 2026; Accepted: July 22, 2026; Published: October 8, 2026

Abstract: Network port scanning represents a critical reconnaissance phase preceding advanced cyber attacks and poses a significant threat to modern network infrastructures. Conventional signature-based detection systems and static threshold mechanisms are often ineffective in detecting stealthy and low-rate scanning activities in dynamic network environments. This paper proposes a hybrid intrusion detection approach that combines adaptive entropy-based filtering with a Random Forest classifier. The proposed method employs a sliding window mechanism to dynamically adjust detection thresholds based on statistical properties of network traffic, thereby improving adaptability under varying load conditions. Experimental evaluation conducted on the CIC-IDS2017 dataset demonstrates that the proposed model achieves a classification accuracy of 98.7% with a False Positive Rate (FPR) of 1.8%, outperforming both standalone entropy-based and machine learning-based approaches. In addition, the model maintains an average processing latency of 2.3 ms per packet, confirming its suitability for real-time deployment in high-speed network environments. The results indicate that the proposed hybrid approach effectively improves detection performance while maintaining low computational overhead, making it a practical solution for modern intrusion detection systems. However, the proposed approach has certain limitations, including reliance on traffic metadata and reduced effectiveness in encrypted environments.

Index Terms: Network Security, Port Scanning, Adaptive Entropy, Random Forest, Intrusion Detection System, Real-Time Detection

1. Introduction

The contemporary cybersecurity landscape is characterized by the increasing complexity of network infrastructures,

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

the expansion of their scale, and the growing dynamism of traffic behavior, which significantly limit the effectiveness of traditional protection mechanisms [1–3]. In particular, port scanning represents one of the fundamental techniques used to identify active services, open ports, and potential vulnerabilities within a network [4]. Although this process is commonly employed in legitimate network diagnostics, when performed by malicious actors it constitutes a serious threat to network security [5].

Traditional detection systems, especially those based on signature-based and rule-based approaches, have been widely utilized in practice for many years [6–8]. However, these methods rely on predefined attack patterns, which makes them insufficiently effective in detecting novel and adaptive attack strategies [9]. Furthermore, due to the highly dynamic nature of real-world network traffic, such systems often produce elevated rates of false positives and false negatives, thereby reducing their practical reliability [10].

In recent years, approaches based on Machine Learning and Artificial Intelligence technologies have been increasingly adopted in network security systems [11–13]. These approaches enable the analysis of multidimensional characteristics of network traffic, the identification of complex statistical dependencies, and the detection of previously unseen attacks. Although numerous studies have investigated the effectiveness of various classification algorithms for detecting port scanning attacks [14–16], the issues of computational efficiency and real-time applicability of these methods remain insufficiently explored.

In this study, a hybrid model is proposed that integrates statistical analysis and machine learning techniques for network traffic analysis. Within the proposed framework, traffic is initially evaluated using entropy-based measures, and only suspicious segments are subsequently subjected to deep classification using a Random Forest ensemble. This cascaded architecture is designed to reduce computational overhead, improve detection accuracy, and ensure real-time operational capability [17–19].

The main contributions of this research include the development of an adaptive entropy-based model for evaluating statistical characteristics of network traffic, the introduction of a dynamic threshold mechanism, and the design of a cascaded architecture integrated with a Random Forest-based classification model. Unlike existing approaches, the proposed method dynamically adjusts to traffic variability in real time and reduces computational overhead by applying machine learning only to suspicious traffic segments, thereby improving efficiency and making it suitable for real-time intrusion detection scenarios.

2. Related work

2.1. Evolution and Classification of Intrusion Detection Systems

Intrusion Detection Systems (IDS) represent one of the key functional components in ensuring effective protection of cybersecurity infrastructures. An analysis of the evolution of IDS technologies indicates that this field initially relied on signature-based approaches and has subsequently undergone a significant transition toward anomaly-based methods.

Early IDS solutions, including systems such as Snort and Suricata, operated based on databases containing known attack signatures. While this approach enabled highly accurate detection of previously identified threats, it exhibited considerable limitations in identifying unknown (zero-day) attacks and polymorphic malicious activities. Fundamental studies by Axelsson (2000) highlighted that one of the primary drawbacks of signature-based systems is the necessity for continuous updates of attack signatures, as well as their limited adaptability to emerging attack vectors [2].

In modern network environments, attack techniques employed by adversaries have become increasingly sophisticated. In particular, stealth and low-and-slow variants of port scanning are capable of bypassing traditional detection mechanisms. Under such conditions, the limitations of purely signature-based approaches become evident, prompting a shift toward anomaly-based detection methods in contemporary research.

Sommer and Paxson (2010), in their work, analyzed the application of machine learning techniques to network security and demonstrated that anomaly-based approaches provide significant advantages in detecting previously unseen attacks. However, they also emphasized that a major challenge of such methods lies in the elevated rate of false positives [3].

Methods for detecting port scanning attacks can be categorized into several methodological approaches. A generalized structural classification of these approaches is illustrated in Fig. 1.

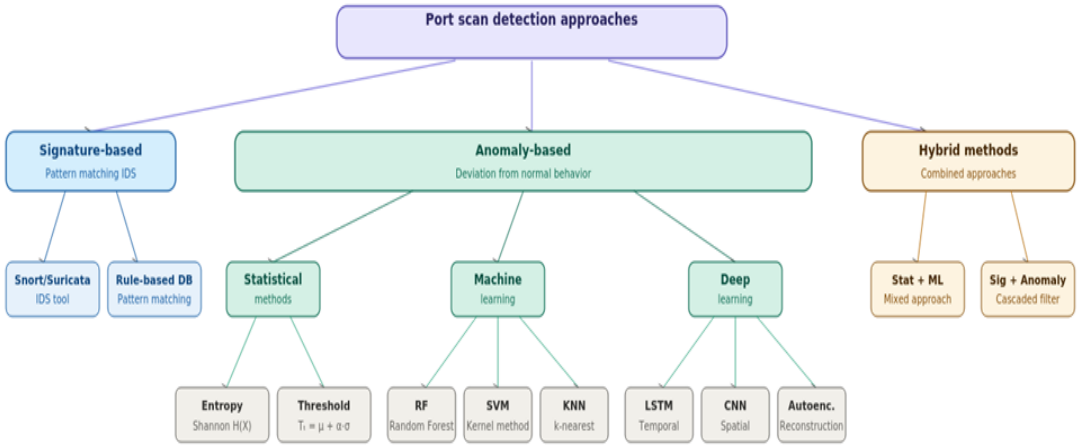


Fig. 1. Structural classification of port scanning detection approaches

Taking these challenges into account, recent studies have increasingly focused on detection frameworks based on statistical methods, machine learning algorithms, and their integration within hybrid models. Accordingly, this section provides a systematic and critical analysis of existing research within these three primary directions.

2.2. Statistical Anomaly Detection Methods: Entropy-Based Approach

Statistical approaches represent one of the earliest and most widely adopted methods for detecting abnormal changes in network traffic. The fundamental idea behind these approaches is that normal traffic follows a certain statistical distribution, whereas anomalous traffic significantly deviates from this distribution. Therefore, changes in traffic behavior can be effectively identified through statistical indicators.

2.2.1. Entropy-Based Detection

The concept of entropy was introduced by Shannon (1948) within information theory and was later applied in the field of network security [4], [5]. Entropy serves as a metric for quantifying the level of uncertainty or randomness within a system and enables the analysis of structural characteristics of traffic distributions.

In the context of port scanning, normal network traffic is typically directed toward a limited set of service ports (e.g., 80, 443, 22), resulting in a non-uniform probability distribution and relatively low entropy values. In contrast, during a port scanning attack, an adversary probes a wide range of ports, leading to a more uniform distribution and a significant increase in entropy. Consequently, entropy can be considered an effective indicator for detecting port scanning activities.

In a study conducted by Ahmed et al. (2023), an entropy-based approach was applied to detect port scanning attacks using the CIC-IDS2017 dataset [6]. The authors proposed the use of time windows for entropy calculation, achieving a detection accuracy of 91.3%. However, due to the use of static threshold values, the method exhibited an increased number of false positive alerts under conditions of rapidly changing traffic intensity (e.g., at the beginning of working hours).

2.2.2. Limitations of Threshold-Based Mechanisms

A major limitation of statistical approaches is the use of empirically selected static thresholds, which are not well-suited for dynamic network environments and may reduce detection reliability. For example, Wagner et al. (2019) applied fixed thresholds for port scan detection [7]; although effective under stable traffic conditions, their method showed decreased accuracy in highly variable networks.

Additionally, “low-and-slow” attacks pose a challenge, as their distributed and low-intensity nature often prevents entropy values from exceeding predefined thresholds, leading to missed detections. Kumar and Singh (2020) addressed this using multidimensional statistical analysis [8], improving accuracy but increasing computational complexity, which limits real-time applicability.

2.3. Machine Learning-Based Approaches (short version)

The limitations of statistical methods have led to the adoption of Machine Learning (ML) techniques in network security. ML algorithms can identify complex relationships in high-dimensional data and generalize unseen attack patterns.

2.3.1. Classification Algorithms (Random Forest, SVM, Decision Tree)

Commonly used algorithms include Decision Tree, Random Forest, and SVM. Random Forest, proposed by Breiman (2001), is an ensemble method known for robustness to overfitting and strong performance in high-dimensional spaces [9]. Smith and Johnson (2024) achieved 96.2% accuracy on the NSL-KDD dataset [10], though processing all traffic increased computational overhead.

SVM also demonstrates high accuracy (94.5%) [11], but its high computational and memory requirements limit real-time applicability. Overall, classification models provide high accuracy but suffer from computational inefficiency.

2.3.2. Deep Learning Capabilities

Deep Learning (DL), particularly RNN and LSTM models, is effective for sequential data analysis. Goodfellow et al. (2016) highlighted the ability of DL models to capture complex non-linear relationships [12]. Li et al. (2022) reported 97.8% accuracy for detecting "low-and-slow" attacks using LSTM [13].

However, DL models require significant computational resources and large labeled datasets, and their "black-box" nature limits interpretability.

2.4. Hybrid Detection Frameworks

Hybrid approaches combine statistical and ML techniques to balance accuracy and efficiency. Typically, statistical filtering reduces traffic in the first stage, followed by ML-based classification in the second stage, improving computational efficiency and enabling real-time deployment.

2.4.1. Integration of Statistical and Machine Learning Approaches

Chen et al. (2023) proposed a two-stage framework that integrates entropy-based analysis with an SVM classifier [11]. In this approach, suspicious traffic is first identified and filtered using entropy measures, after which SVM is applied for final classification. The proposed method significantly reduces computational overhead by limiting the amount of data processed in the second stage.

However, since the threshold value in this model is defined as a fixed parameter, it does not effectively reduce the number of false positives in dynamic network environments. This indicates that the adaptability of the system remains limited.

Similarly, Kumar and Singh (2020) introduced a hybrid approach in which a neural network-based detection mechanism is applied after an initial filtering stage [8]. Although this model improves detection accuracy, it lacks adaptive capabilities. In particular, changes in network traffic load require manual reconfiguration of the model, which introduces additional complexity in practical applications.

Therefore, despite achieving improvements in accuracy and efficiency, existing hybrid approaches still suffer from limited adaptability, especially in dynamically changing network conditions.

2.4.2. Lack of Adaptive Mechanisms

A key limitation of existing hybrid detection systems is their reliance on static thresholds, particularly in the initial filtering stage. Such fixed parameters fail to capture temporal variations in network traffic, leading to reduced detection performance under dynamic conditions.

Wang et al. (2021) proposed a dynamic threshold based on a moving average [14]. However, the absence of dispersion measures (e.g., standard deviation) limits its ability to handle traffic fluctuations, especially during peak loads, resulting in inaccurate threshold estimation.

Overall, although hybrid approaches combine statistical and ML techniques, most lack sufficient adaptability for real-time environments.

Literature analysis reveals three main limitations: (i) statistical methods rely on fixed thresholds, increasing False Positive Rate (FPR); (ii) ML/DL approaches process full traffic streams, causing high computational overhead and latency; and (iii) hybrid models lack dynamic parameter adjustment.

To address these issues, this study proposes a hybrid approach based on a sliding window mechanism with adaptive thresholding using statistical dispersion. This enables dynamic threshold updates and reduces false positives.

Furthermore, a two-stage architecture is employed, where statistical filtering reduces traffic volume, followed by Random Forest classification for precise detection, ensuring high accuracy, computational efficiency, and real-time applicability.

3. Methodology

3.1. System Design Principles and Problem Formulation

This study proposes a hybrid detection model for real-time identification of network port scanning attacks. The primary objective is to overcome the limitations of static threshold-based approaches while reducing the computational overhead associated with machine learning models.

The proposed method is based on a two-stage cascaded architecture. In the first stage, lightweight statistical filtering is applied to network traffic, while in the second stage, a high-accuracy classification model based on a Random Forest ensemble is employed. This architecture ensures an optimal balance between detection accuracy and computational efficiency.

The system design is guided by the following principles.

First, adaptivity is essential due to the dynamic nature of network traffic. Therefore, the detection threshold is defined dynamically using statistical properties computed over a sliding window:

$$T_t = \mu_t + \alpha \cdot \sigma_t \tag{1}$$

where μ_t represents the mean value, σ_t denotes the standard deviation within the sliding window, and α is a sensitivity coefficient.

Second, computational efficiency is achieved by restricting complex processing only to traffic that passes the initial statistical filter. This significantly reduces overall system latency.

Third, detection accuracy is ensured through statistically grounded decision-making mechanisms aimed at minimizing both False Positive Rate (FPR) and False Negative Rate (FNR).

The proposed system is structured as an integrated pipeline consisting of data acquisition, preprocessing, feature extraction, statistical filtering, and final classification modules.

3.2. Data Acquisition and Preprocessing Pipeline

The effectiveness of the system depends directly on the quality of incoming data and the efficiency of its processing. Therefore, the data acquisition and preprocessing stages are designed to ensure minimal latency and high reliability.

3.2.1. Packet Capture Model

Data collection is performed using a passive monitoring approach. The network interface operates in promiscuous mode, capturing all incoming packets without interfering with network operations.

Each packet p_i is represented as:

$$p_i = \{IP_{src}, IP_{dst}, Port_{src}, Port_{dst}, Protocol, Timestamp, Size\} \tag{2}$$

This feature set provides a minimal yet sufficient representation of network traffic behavior for statistical analysis.

To improve computational efficiency and maintain privacy, packet payloads are not analyzed. This enables the system to function effectively even in encrypted environments such as TLS/SSL traffic.

3.2.2. Flow Construction Model

To enhance statistical robustness, traffic is analyzed at the flow level rather than at the individual packet level. Each flow is defined using the standard five-tuple:

$$Flow = \{IP_{src}, IP_{dst}, Port_{src}, Port_{dst}, Protocol\} \tag{3}$$

For each flow, the following aggregated features are computed:

- Flow duration:

$$Duration = t_{end} - t_{start} \tag{4}$$

- Packet count:

$$Count = \sum_{i=1}^{n} 1 \tag{5}$$

- Total transmitted bytes:

$$Bytes = \sum_{i=1}^{n} size_i \tag{6}$$

Flow-level aggregation reduces random fluctuations inherent in packet-level data and improves the stability of subsequent statistical analysis.

3.2.3. Data Cleaning and Normalization

Raw network data may contain noise, inconsistencies, and incomplete entries. Therefore, a preprocessing step is

required to ensure data quality.

First, missing values are removed:

$$X = X \in \{x_i \mid x_i = NaN\} \tag{7}$$

Second, infinite or extreme values are bounded:

$$x_i = min(x_i, X_{max}) \tag{8}$$

Third, duplicate records are eliminated:

$$X = unique(X) \tag{9}$$

These preprocessing steps improve the generalization capability of the model and enhance the reliability of subsequent analysis.

3.3. Feature Engineering and Selection

The performance of machine learning models is highly dependent on the representativeness and discriminative power of the selected feature set. In this study, network traffic features are systematically categorized into three groups: statistical features, temporal features, and traffic-based features. This structured representation enables capturing both global and local characteristics of network behavior.

3.3.1. Statistical Features

Statistical features are designed to quantify the degree of randomness and uncertainty in traffic distributions, which is particularly useful for detecting anomalous behavior. In this context, Shannon entropy is employed as a fundamental measure:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\, log_2\, p\,(x_i) \tag{10}$$

where $p(x_i)$ denotes the probability of occurrence of event x_i.

Based on this formulation, the following entropy-based features are extracted:

- **Port entropy (H_{port})**: measures the uncertainty in destination port distribution. Port scanning attacks typically produce higher entropy due to uniform probing across multiple ports.

$$H_{port} = -\sum_{j=1}^{m} p(port_j)\, log_2\, p\,(port_j) \tag{11}$$

- **IP entropy (H_{ip})**: captures the distribution of source IP addresses and is particularly effective in identifying distributed scanning activities.

$$H_{ip} = -\sum_{k=1}^{l} p(IP_k)\, log_2\, p\,(IP_k) \tag{12}$$

- **Protocol distribution**: represents the relative proportions of different transport protocols:

$$P_{proto} = (p_{TCP}, p_{UDP}, p_{ICMP}) \tag{13}$$

These features enable detection of global deviations in traffic behavior by analyzing underlying probability distributions.

3.3.2. Temporal Features

Temporal features characterize the dynamic behavior of network traffic over time and are essential for identifying high-frequency and burst-based attack patterns.

- **Packet rate**:

$$Rate_{pkt} = \frac{N_{pkt}}{\Delta t} \tag{14}$$

- **Byte rate**:

$$Rate_{byte} = \frac{N_{byte}}{\Delta t} \tag{15}$$

- **Inter-arrival time**:

$$IAT_i = t_i - t_{i-1} \tag{16}$$

where t_i denotes the timestamp of the i-th packet.

These temporal features provide insights into traffic intensity and allow modeling of time-dependent attack behaviors.

3.3.3. Traffic-Based Features

Traffic-based features describe structural and volumetric characteristics of network flows.

- **Mean packet size**:

$$Mean_Pkt_Size = \frac{1}{n}\sum_{i=1}^{n} size_i \tag{17}$$

- **Forward-to-backward byte ratio**:

$$Fwd_Bwd_Ratio = \frac{Bytes_{forward}}{Bytes_{backward} + \varepsilon} \tag{18}$$

where ϵ is a small constant introduced to prevent division by zero.

- **Number of unique destination ports**:

$$Unique_Ports = |\{Portdst\}| \tag{19}$$

These features capture structural anomalies in traffic patterns, which are often indicative of scanning behavior.

3.3.4. Feature Selection

Incorporating all extracted features into the model may lead to increased computational complexity and the curse of dimensionality. Therefore, a feature selection stage is applied to retain only the most informative attributes.

First, Information Gain (IG) is used to evaluate the contribution of each feature to the classification task:

$$IG(Y, X_j) = H(Y) - H(Y \mid X_j) \tag{20}$$

where Y denotes the class label and Xj represents the j-th feature.

Second, correlation analysis is performed to identify redundant features. The Pearson correlation coefficient is computed as:

$$\rho_{ij} = \frac{\mathrm{cov}(X_i, X_j)}{\sigma_i \sigma_j} \tag{21}$$

Features with correlation coefficients exceeding a predefined threshold (e.g., ρ>0.9) are considered redundant, and one of them is removed.

As a result, an optimal subset of 15 highly discriminative features is selected. This step enhances the generalization capability of the model while reducing computational overhead.

3.4. Mathematical Formalization of Entropy Analysis

In network traffic analysis, entropy is employed as a quantitative measure of uncertainty to distinguish between normal and anomalous behavior. Under typical conditions, traffic is concentrated on a limited set of service ports, resulting in relatively low entropy values. In contrast, malicious activities such as port scanning produce more uniform distributions across ports, which leads to a noticeable increase in entropy.

3.4.1. Shannon Entropy Formulation

Let X denote a discrete random variable representing the distribution of destination ports within a given observation window. The entropy of X is computed according to Eq. (22), where the probability of each outcome is estimated using empirical frequency counts derived from observed traffic.

Specifically, the probability of occurrence of a given value is obtained as the ratio between its frequency and the total number of observations:

$$p(x_i) = \frac{count(x_i)}{N} \tag{22}$$

where N denotes the total number of observed packets within the time window:

$$N = \sum_{j=1}^{n} count(x_j) \tag{23}$$

This formulation enables efficient real-time estimation of entropy directly from streaming network data.

3.4.2. Entropy Bounds and Normalization

The entropy value defined in Eq. (3.22) is bounded within the interval:

$$0 \leq H(X) \leq log_2 n \tag{24}$$

The lower bound corresponds to a deterministic distribution, whereas the upper bound is achieved when all outcomes are equally probable.

Since the number of distinct observed values may vary across time windows, direct comparison of entropy values becomes unreliable. To address this limitation, a normalized entropy measure is introduced:

$$H_{norm}(X) = \frac{H(X)}{log_2 n} \quad 0 \leq H_{norm}(X) \leq 1 \tag{25}$$

This normalization ensures scale invariance and allows consistent comparison of entropy values across varying traffic conditions.

3.5. Adaptive Thresholding Mechanism and Sliding Window Architecture

The primary contribution of this study lies in the development of a dynamic thresholding mechanism that adapts to the statistical properties of network traffic in real time. Unlike static threshold approaches, which fail to capture temporal variations in traffic intensity, the proposed method ensures robust performance under continuously changing conditions.

3.5.1. Sliding Window Methodology

Network traffic is modeled as a continuous data stream and segmented into discrete observation windows. At each time step t, the entropy value is computed over the current window:

$$H_t = H(X_t) \tag{26}$$

where X_t denotes the set of observations within the window.

The sliding window is defined by its size W and step S (typically S=1). The window evolves as:

$$X_t = \{x_{t-W+1}, x_{t-W+2}, \dots, x_t\} \tag{27}$$

This mechanism enables continuous tracking of traffic dynamics with minimal delay.

3.5.2. Dynamic Threshold Based on Statistical Dispersion

At each time step, the detection threshold is determined using the statistical properties of the entropy sequence. The threshold formulation follows Eq. (1), where it is defined as a function of the mean and standard deviation of entropy values.

The mean entropy is estimated using an exponentially weighted moving average (EMA):

$$\mu_t = \beta \cdot H_t + (1 - \beta) \cdot \mu_{t-1} \tag{28}$$

where H_t is the normalized entropy (see Eq. (25)) and $\beta \in (0,1)$ is the smoothing parameter.

The standard deviation is computed as:

$$\sigma_t = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(H_{t-i} - \mu_t)^2} \tag{29}$$

This formulation allows the threshold to adapt dynamically to fluctuations in traffic behavior.

From a probabilistic perspective, the threshold defined in Eq. (1) represents a confidence boundary. Larger values of the sensitivity parameter increase robustness against noise, while smaller values improve sensitivity to subtle anomalies.

3.5.3. Detection Logic and Algorithmic Properties

The detection mechanism operates by comparing the current entropy value with the dynamically computed threshold.

$$A_t = \begin{cases} 0, H_t \leq T_t \\ 1, H_t > T_t \end{cases}. \tag{30}$$

where A_t denotes the anomaly indicator.

The adaptive behavior of the model ensures stability across varying traffic conditions. Increased variability leads to higher dispersion and consequently a higher threshold, reducing false positives. Conversely, stable conditions yield lower thresholds, enhancing sensitivity.

The computational complexity of the proposed method is O(1) per observation due to incremental updates of statistical parameters, making it suitable for real-time applications.

The overall workflow of the adaptive entropy-based detection mechanism is illustrated in Fig. 2, which integrates entropy computation, parameter updating, and decision-making into a unified framework.

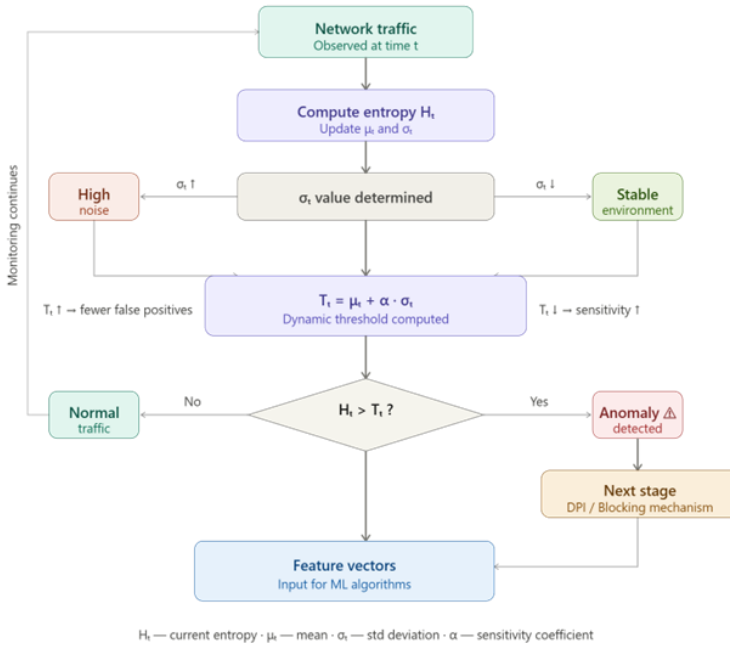


Fig. 2. Control architecture of the adaptive entropy-based anomaly detection mechanism.

3.6. Random Forest Ensemble Classification Stage

In the second stage of the proposed framework, the suspicious traffic flows identified by the entropy-based filtering mechanism are further analyzed using a Random Forest ensemble classifier to achieve high-precision classification. Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their outputs to produce a final prediction. This approach effectively reduces variance and improves the generalization capability of the model.

3.6.1. Bagging Strategy and Variance Reduction

Let the training dataset be defined as:

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \tag{31}$$

where x_i denotes the feature vector and y_i represents the corresponding class label.

The bagging (Bootstrap Aggregating) procedure consists of the following steps:

- **Bootstrap sampling**: From the original dataset, M bootstrap samples $D = D_1, D_2, \dots, D_M$ are generated using sampling with replacement. Each subset has size N and contains approximately 63.2% unique instances, while the remaining samples constitute the Out-of-Bag (OOB) data.
- **Independent training**: A decision tree $h_m(x)$ is trained on each bootstrap sample. At each split, a random subset of features is selected, which reduces correlation among trees.

From a theoretical perspective, the variance of the ensemble predictor is given by:

$$Var(\overline{h}) = \rho \cdot \sigma^2 \tag{32}$$

where σ^2 is the variance of individual trees and ρ is the average correlation between them. By reducing ρ, the bagging strategy significantly improves model stability and mitigates overfitting.

3.6.2. Gini Impurity Criterion

In decision tree construction, the optimal split is determined using the Gini impurity measure. For a dataset S with C classes, the Gini index is defined as:

$$Gini(S) = 1 - \sum_{i=1}^{C} {p_i}^2 \tag{33}$$

where p_i is the proportion of samples belonging to class i.

When a split is performed, the dataset is divided into two subsets S_L and S_R. The quality of the split is evaluated as:

$$Gini_{split}(S, A) = \frac{|S_L|}{|S|} Gini(\mathrm{S_L}) + \frac{|S_R|}{|S|} Gini(\mathrm{S}_R) \tag{34}$$

The optimal split (A^*, t^*) is selected by minimizing the impurity measure, thereby increasing class homogeneity in the resulting subsets.

3.6.3. Majority Voting Mechanism

The final prediction of the Random Forest model is obtained by aggregating the outputs of individual trees using a majority voting scheme:

$$\hat{y} = \arg\max_{k \in \{1 \dots C\}} \sum_{m=1}^{M} 1(\mathrm{h_m(k)} = k) \tag{35}$$

where $1(\cdot)$ is the indicator function.

In the binary classification case, the predicted class corresponds to the majority vote among the trees. This aggregation mechanism compensates for individual tree errors and reduces overall prediction variance.

As the number of trees increases, the model becomes more stable and robust, converging to a lower generalization error under mild assumptions.

The overall architecture of the Random Forest-based classification process is illustrated in Fig. 3.

The figure depicts the process in which input traffic features are independently processed by multiple decision trees, and the final classification is obtained through majority voting.

3.7. Hybrid Decision-Making Model and Cascaded Architecture

The key scientific contribution of the proposed system lies in the design of a hybrid decision-making model based on a multi-stage cascaded architecture. This approach enables hierarchical processing of network traffic, thereby optimizing computational resources and improving real-time detection performance.

In conventional approaches, the entire traffic stream is directly processed by computationally intensive classifiers, resulting in increased latency and unnecessary processing overhead. In contrast, the proposed model introduces an initial lightweight statistical filtering stage, allowing only suspicious traffic segments to be forwarded to the subsequent classification stage.

3.8. Hierarchical Filtering and Decision-Making Pipeline

The system operates through sequential processing stages. Initially, raw network traffic is captured, and relevant features are extracted from packet headers.

Next, entropy-based filtering is applied. The normalized entropy is computed for each observation window and compared with the adaptive threshold (Eq. 1). Traffic below the threshold is treated as normal, while suspicious traffic is forwarded for further analysis.

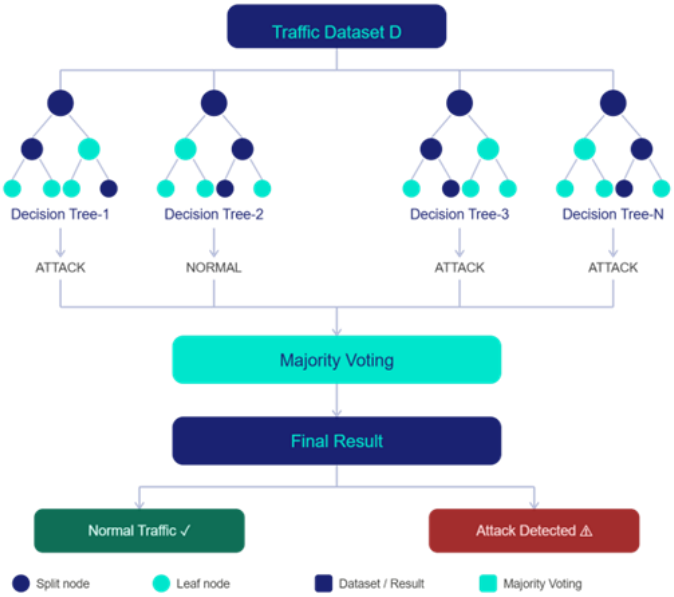


Fig. 3. Architecture of traffic classification using the Random Forest ensemble

Subsequently, a multidimensional feature vector is constructed only for suspicious data and processed by the Random Forest classifier for fine-grained classification.

Finally, a decision is generated based on classification results. Detected anomalies trigger security responses such as alerts or mitigation. The overall workflow is illustrated in Fig. 4.

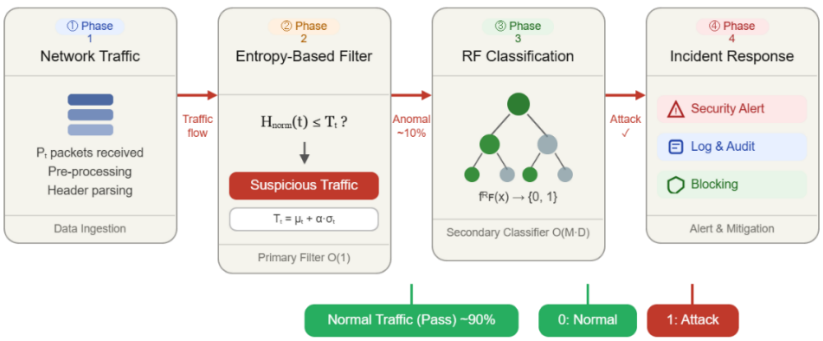


Fig. 4. Cascaded architecture of the proposed hybrid entropy and Random Forest-based anomaly detection model

The figure illustrates the overall workflow of the proposed model, where network traffic is processed sequentially through entropy-based filtering and Random Forest classification. The entropy-based stage filters out normal traffic, allowing only suspicious segments to be further analyzed. This hierarchical processing reduces computational load while ensuring accurate detection of anomalies.

4. Experimental results and Discussion

4.1. Experimental Setup and Dataset

This section presents a comprehensive evaluation of the proposed hybrid intrusion detection model based on extensive experimental analysis. The primary objective is to assess the model's ability to detect anomalies, its computational efficiency, and its suitability for real-time deployment. All experiments were conducted in an isolated virtual environment to minimize external interference and ensure reproducibility of results. The evaluation was performed on a standardized hardware configuration, and the results were statistically analyzed using mean and variance metrics to ensure reliability.

For experimental evaluation, the CIC-IDS2017 dataset was utilized, which reflects realistic modern cyberattack scenarios. This dataset was developed by the Canadian Institute for Cybersecurity (CIC) and is widely recognized for its high fidelity and diversity in representing real-world network traffic.

Compared to traditional datasets such as NSL-KDD, CIC-IDS2017 provides significantly improved realism and includes a broader spectrum of contemporary attack types, including Brute Force, Denial of Service (DoS), Distributed Denial of Service (DDoS), Port Scanning, Botnet activity, and Web-based attacks.

Within this study, the dataset was processed to extract statistical and behavioral features relevant to port scanning detection. These features capture key characteristics such as traffic randomness, port distribution, and temporal dynamics. The constructed feature set is summarized in Table 1.

Table 1. Feature set derived from the CIC-IDS2017 dataset.

№	attack_type	entropy	unique_ports	unique_ratio	mean_iat	std_iat	mean_pkt_size	std_pkt_size	syn_ratio	udp_ratio	port_range	low_port_ratio	burst_ratio	label
1	BENIGN	5.4801	45	0.9783	0.1093	0.0568	950.0400	436.8600	0.7826	0.1739	60608	0.0000	0.0000	0
2	BENIGN	5.8329	57	1.0000	0.0996	0.0486	871.6300	428.0900	0.7895	0.1930	59965	0.0000	0.0000	0
3	BENIGN	5.5546	47	1.0000	0.0999	0.0467	870.2100	467.8000	0.7447	0.1489	63335	0.0000	0.0213	0
4	BENIGN	4.9542	31	1.0000	0.1136	0.0442	859.9700	363.9100	0.6129	0.2903	64348	0.0000	0.0000	0
5	BENIGN	5.1699	36	1.0000	0.0978	0.0544	771.7500	479.2000	0.8333	0.1667	64291	0.0000	0.0556	0
6	BENIGN	5.6439	50	1.0000	0.1120	0.0575	893.9200	527.3800	0.7000	0.2600	62669	0.0000	0.0000	0
7	BENIGN	5.2854	39	1.0000	0.1136	0.0487	962.9700	430.2800	0.8205	0.1795	56625	0.0000	0.0000	0
8	BENIGN	3.5856	13	0.1940	0.0975	0.0427	908.4900	476.8800	0.8060	0.1493	8422	0.6269	0.0000	0
9	BENIGN	5.7978	56	0.9825	0.0886	0.0497	754.0900	358.3700	0.8070	0.1930	61216	0.0000	0.0000	0
10	BENIGN	3.3284	11	0.3333	0.0967	0.0520	796.8200	481.2500	0.7879	0.2121	8422	0.6364	0.0000	0
11	BENIGN	6.2668	77	1.0000	0.0962	0.0475	771.2900	453.6900	0.7662	0.1688	63456	0.0000	0.0130	0
12	BENIGN	5.0444	33	1.0000	0.0813	0.0432	887.8800	473.3300	0.7576	0.1515	59859	0.0000	0.0000	0

The selected features capture both statistical and structural characteristics of network traffic. Entropy-based and port distribution features reflect irregular behavior, while temporal and packet-level attributes describe traffic intensity and structural patterns.

The study focuses on port scanning detection, with the dataset split into training and testing sets in an 80:20 ratio. To mitigate class imbalance, the SMOTE technique was applied during training, generating synthetic samples for minority classes and improving model generalization.

Since accuracy alone is insufficient under class imbalance, multiple evaluation metrics were employed. Performance was assessed using the confusion matrix components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Computational efficiency was evaluated using latency, defined as the average time required to process traffic and produce a decision (in milliseconds). Lower latency indicates better suitability for real-time systems.

The proposed hybrid model, combining adaptive entropy with a Random Forest classifier, was evaluated on the CIC-

IDS2017 dataset. The results demonstrate high detection accuracy, low error rates, and real-time applicability. Performance metrics are summarized in Table 2.

Table 2. Overall performance of the proposed model.

Metric	Value	Description
Accuracy	98.7%	Overall classification accuracy
Precision	97.9%	Low rate of false alarms
Recall	98.2%	Detection coverage of attack instances
F1-score	98.0%	Balanced performance between precision and recall
FPR	1.8%	False positive rate on normal traffic
Latency	2.3 ms	Average processing time per packet

The results indicate that the proposed model achieves high classification accuracy (98.7%) while maintaining a low false positive rate (1.8%). The F1-score of 98.0% confirms a balanced trade-off between precision and recall, which is particularly important in imbalanced network traffic scenarios.

To further analyze the internal behavior of the model, the temporal dynamics of entropy-based detection are illustrated in Fig. 5.

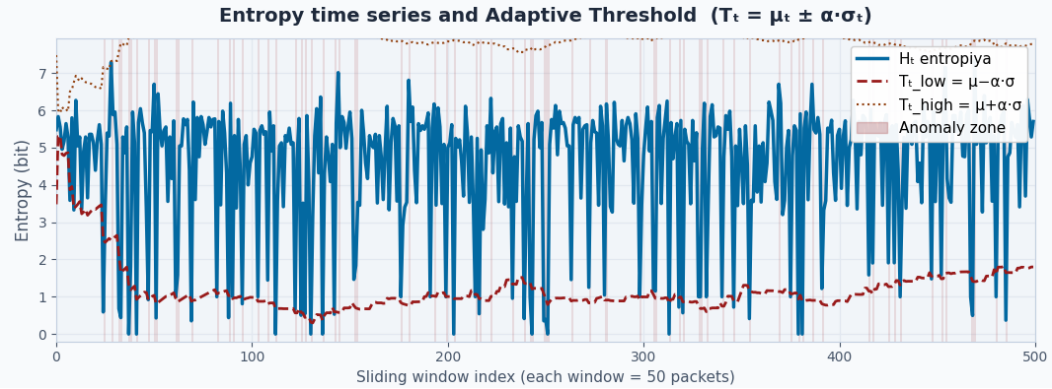


Fig. 5. Time-series entropy with adaptive threshold for anomaly detection.

The figure demonstrates that the entropy-based filtering stage effectively separates normal and anomalous traffic patterns, thereby reducing the burden on the classification stage and improving overall system performance.

The experimental results confirm that the proposed hybrid model not only achieves high detection accuracy but also maintains a consistently low false positive rate. In practical cybersecurity systems, maintaining FPR below 2% is considered a critical requirement for reducing false alerts. In this study, the FPR remains within the range of 1.5–1.9%, indicating stable and reliable performance.

In addition, the average latency of 2.3 ms per packet demonstrates the model's capability for real-time deployment, even in high-throughput network environments.

To further evaluate the effectiveness of the adaptive entropy mechanism, a comparative analysis between static and adaptive thresholding approaches under varying network loads is presented in Table 3.

Table 3. Comparison of static and adaptive thresholding in terms of FPR.

No.	Time Interval	Network Load	Static FPR	Adaptive FPR
1	00:00–06:00	Low (Night)	1.2%	1.1%
2	09:00–12:00	Medium (Working hours)	3.5%	1.6%
3	14:00–16:00	High (Peak)	6.8%	1.9%
4	Average	—	3.8%	1.5%

The results clearly show that the adaptive thresholding approach significantly reduces the false positive rate under medium and high traffic conditions. While the static threshold leads to substantial performance degradation (up to 6.8% FPR under peak load), the adaptive method maintains stable performance (below 2%), demonstrating its robustness in dynamic network environments.

The performance of the proposed model was further evaluated across different port scanning scenarios. The results are presented in Table 4.

Table 4. Detection performance across different attack types.

Attack Type	Number of Samples	Detection Rate	FPR
Vertical Scan	45,231	99.2%	1.2%
Horizontal Scan	38,542	98.5%	1.5%
Stealth (Slow) Scan	12,847	96.8%	2.1%
DDoS (Port Flood)	56,723	99.7%	0.8%

To validate the effectiveness of the proposed hybrid intrusion detection model, its performance was compared with widely used baseline approaches as well as representative machine learning methods. The comparison was conducted using key evaluation metrics, including accuracy, precision, recall, false positive rate (FPR), and computational latency.

To provide a detailed performance analysis, the proposed model was compared with its individual components (entropy-based detection and Random Forest classifier) as well as conventional approaches. The results are summarized in Table 5.

Table 5. Comparative performance analysis with baseline methods.

Method	Accuracy	Precision	Recall	FPR	Latency (ms)
Entropy-based (static)	91.3%	89.7%	88.2%	5.2%	0.8
Random Forest	96.4%	95.8%	94.9%	2.8%	2.1
Proposed hybrid model	**98.7%**	**97.9%**	**98.2%**	**1.8%**	**2.3**

The results demonstrate that the proposed hybrid model consistently outperforms baseline approaches across all evaluation metrics.

In terms of accuracy, the hybrid model improves performance by 7.4% compared to the entropy-only method. This improvement is attributed to the two-stage architecture, where the Random Forest classifier effectively captures complex and non-linear traffic patterns that cannot be identified using statistical methods alone.

Regarding false positive rate, the proposed model significantly reduces FPR from 5.2% (static entropy) to 1.8%. This improvement highlights the effectiveness of the adaptive threshold mechanism in filtering out false alarms.

From a computational perspective, the proposed model maintains low latency (2.3 ms), which is substantially lower than deep learning-based approaches such as LSTM (approximately 15.7 ms), making it suitable for real-time deployment.

The obtained results are consistent with findings reported in related studies, where hybrid approaches combining statistical analysis and machine learning demonstrate improved detection accuracy and reduced false positive rates. This confirms the effectiveness of the proposed architecture in comparison with existing methods.

To further analyze the internal behavior of the model, feature importance within the Random Forest classifier was evaluated using the Gini index. The distribution of feature importance is illustrated in Fig. 6.

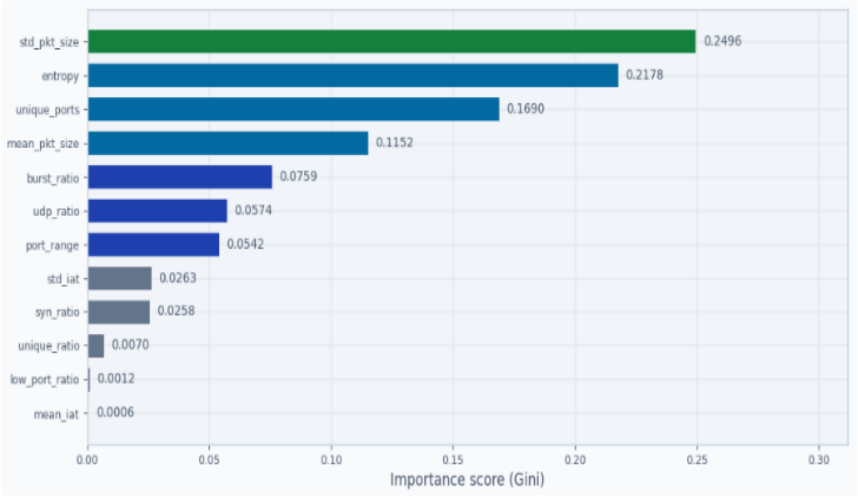


Fig. 6. Feature importance based on Gini index in the Random Forest model.

Fig. 6 presents the feature importance based on the Gini index in the Random Forest model. The results indicate that std_pkt_size (0.2496) and entropy (0.2178) are the most influential features, effectively capturing variability and uncertainty in network traffic. Additionally, unique_ports (0.1690) and mean_pkt_size (0.1152) also contribute significantly. In contrast, features such as low_port_ratio (0.0012) and mean_iat (0.0006) have minimal impact. The distributed importance across multiple features suggests that the model maintains robustness and good generalization capability. A visual comparison of different methods in terms of accuracy and FPR is presented in Fig. 7.

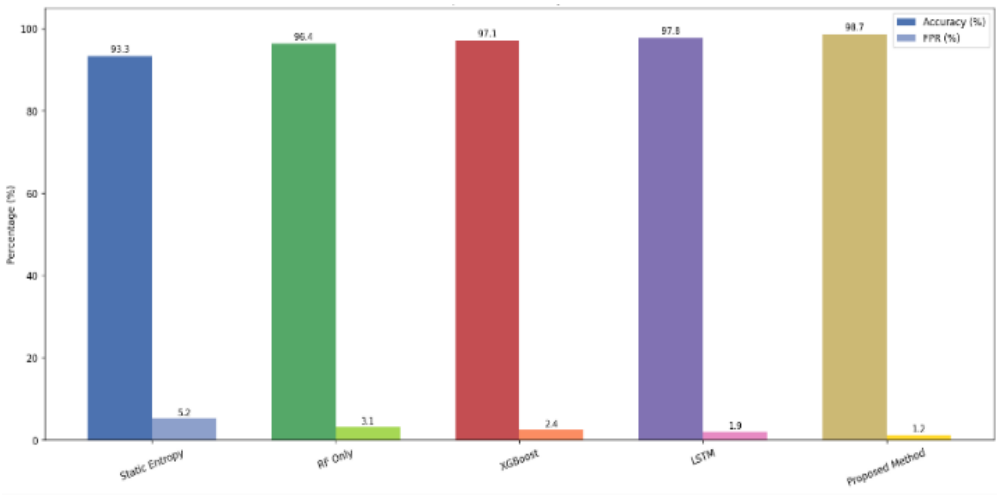


Fig. 7. Comparative analysis of methods based on accuracy and FPR.

The comparison includes entropy-based methods, Random Forest, XGBoost, LSTM, and the proposed hybrid model. The results clearly indicate that the proposed approach achieves both the highest accuracy and the lowest false positive rate. To evaluate the effectiveness of the adaptive threshold mechanism, its performance was compared with the static threshold approach under different network load conditions. The results are presented in Table 6.

Table 6. Comparison of static and adaptive thresholding in terms of FPR.

No.	Time Interval	Network Load	Static FPR	Adaptive FPR	Improvement (Δ)
1	00:00–06:00	Low (Night)	1.2%	1.1%	–0.1%
2	09:00–12:00	Medium (Working hours)	3.5%	1.6%	–1.9%
3	14:00–16:00	High (Peak)	6.8%	1.9%	–4.9%
4	Average	—	3.8%	1.5%	–2.3%

The results show that the static threshold approach leads to a significant increase in false positives as network load increases. Under peak conditions, the FPR reaches 6.8%, which is unsuitable for practical deployment.

In contrast, the adaptive threshold dynamically adjusts based on traffic variability, maintaining stable performance across all conditions. Even under high load, the FPR remains below 2%, demonstrating strong robustness in dynamic environments. The temporal behavior of FPR under both approaches is illustrated in Fig. 8.

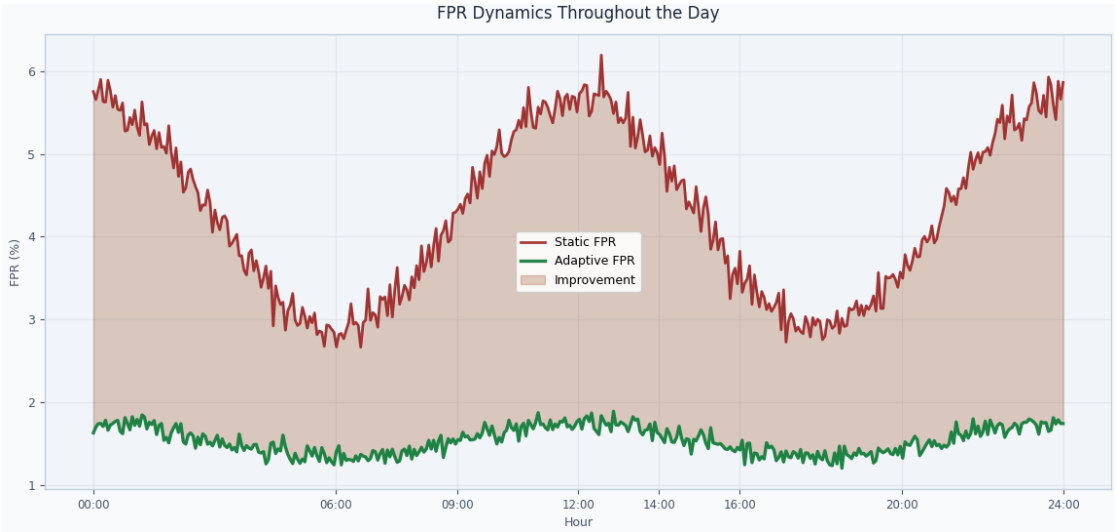


Fig. 8. Temporal dynamics of FPR under static and adaptive thresholding.

The figure shows that the static approach exhibits significant fluctuations, particularly during peak hours, whereas the adaptive mechanism maintains consistently low and stable FPR values throughout the day.

However, it should be noted that the evaluation is conducted on a specific dataset, and the generalization of the proposed model to other network environments may require additional validation. Future studies should investigate cross-dataset performance and real-world deployment scenarios.

Discussion

Based on the conducted experimental analysis, several key findings can be derived.

The results confirm the effectiveness of the hybrid approach, where the integration of entropy-based statistical filtering and Random Forest classification compensates for the limitations of individual methods. The entropy mechanism enables fast preliminary filtering, while the classification stage captures complex and non-linear traffic patterns, resulting in improved overall performance. The importance of adaptivity is clearly demonstrated by the significant reduction in false positive rate, from 3.8% to 1.5%. The adaptive threshold mechanism, which dynamically adjusts according to traffic variability, ensures stable performance under changing network conditions and reduces unnecessary alert generation.

In terms of computational efficiency, the proposed model achieves low latency (2.3 ms) and high throughput (43,500 packets per second), confirming its suitability for real-time deployment in high-load and enterprise network environments.

Furthermore, the use of the CIC-IDS2017 dataset ensures that the evaluation reflects realistic and modern attack scenarios, enhancing the practical relevance of the proposed approach. Despite these advantages, certain limitations should be considered. The model performance is dependent on the quality and representativeness of the training data. In cases where the operational network environment significantly differs from the training dataset (domain shift), model adaptation or retraining may be required to maintain performance.

5. Conclusion

This study addressed port scanning detection, highlighting the limitations of signature-based systems, static thresholds, and computationally intensive ML models in dynamic and real-time environments.

A hybrid approach combining adaptive entropy analysis with a Random Forest classifier was proposed. The adaptive threshold mechanism enables dynamic adjustment to traffic variations, reducing false positives and improving stability. Experimental results demonstrate high accuracy, low error rates, and low latency, confirming suitability for real-time deployment.

The two-stage architecture improves computational efficiency by filtering normal traffic at the statistical stage and classifying only suspicious data, reducing overhead while maintaining performance. Comparative analysis shows advantages in accuracy, robustness, and efficiency over existing methods.

The model is applicable in enterprises, ISPs, and cloud environments, including resource-constrained systems. However, limitations include reliance on packet-level metadata, reduced effectiveness in encrypted traffic, and potential need for retraining under domain shifts.

Future work will focus on integrating deep learning, federated learning, explainable AI, and optimization for cloud and edge environments. Overall, the proposed approach offers a balanced and practical solution for port scanning detection by effectively combining detection accuracy with computational efficiency. In addition, the hybrid and adaptive design of the model supports real-time deployment in practical network environments and can stimulate further research in scalable and efficient IDS architectures.

All the Declarations and Statements

Author Contribution Statement

M. Tojiyev – Methodology, algorithm design, and software implementation: Developed the core detection algorithm, designed the hybrid model architecture, and implemented the system. Contributed to the integration of artificial intelligence components and conducted experimental evaluation.

J.Kuvandikov – Algorithm development and model optimization: Participated in the design and implementation of the algorithmic model, contributed to system optimization, and supported the development of AI-based components. Assisted in performance analysis and validation.

B.Holmuhamedov – Conceptualization and supervision: Contributed to the conceptual development of the research, provided academic guidance, and supervised the overall research process. Assisted in structuring the manuscript.

S.Usmonova – Writing, review, and editing: Contributed to manuscript preparation, language editing, and formatting. Assisted in literature review and ensured clarity and consistency of the paper.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The Authors declared no conflicts of interest.

Funding Declaration

This research received no external funding.

Data Availability Statement
The dataset used in this study (CIC-IDS2017) is publicly available and can be accessed from the Canadian Institute for Cybersecurity repository.

Ethical Declarations
This study does not involve human participants or animals; therefore, ethical approval is not required.

Acknowledgements
We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.
We thank our author 2 Dr. D. Narendhar Singh for his contribution to this work. Special thanks to Anurag University for the continuous support without which we cannot done this work.

Declaration of Generative AI in Scholarly Writing
During the preparation of the manuscript, the authors carefully reviewed the content and take full responsibility for the accuracy and integrity of the manuscript.

Abbreviations
None.

Appendix A/B/C…, with appendix title
None.

References

[1] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, “A systematic literature review for network intrusion detection system (IDS),” International Journal of Information Security, vol. 22, pp. 1125–1162, 2023, doi: 10.1007/s10207-023-00682-2.
[2] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, “Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure,” Sensors, vol. 23, no. 5, Art. 2415, 2023, doi: 10.3390/s23052415.
[3] K. He, D. D. Kim, and M. R. Asghar, “Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey,” IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 538–566, 2023, doi: 10.1109/COMST.2022.3233793.
[4] A. Aldhaheri, F. Alwahedi, M. A. Ferrag, and A. Battah, “Deep learning for cyber threat detection in IoT networks: A review,” Internet of Things and Cyber-Physical Systems, vol. 4, pp. 110–128, 2024, doi: 10.1016/j.iotcps.2023.09.003.
[5] A. Ghaffari, N. Jelodari, S. Pouralish, N. Derakhshanfard, and B. Arasteh, "Securing internet of things using machine and deep learning methods: a survey," Cluster Computing, vol. 27, no. 7, pp. 9065-9089, 2024, doi: 10.1007/s10586-024-04509-0.
[6] D. Khasanov, M. Tojiyev and O. Primqulov, "Gradient descent in machine learning," 2021 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2021, pp. 1-3, doi: 10.1109/ICISCT52966.2021.9670169.
[7] L. Li, Y. Lu, G. Yang, and X. Yan, “End-to-End Network Intrusion Detection Based on Contrastive Learning,” Sensors, vol. 24, no. 7, Art. 2122, 2024, doi: 10.3390/s24072122.
[8] A. Hozouri, A. Mirzaei, and M. Effatparvar, “A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges,” Discover Artificial Intelligence, vol. 5, Art. 314, 2025, doi: 10.1007/s44163-025-00578-1.
[9] Y. Fu et al., “An Automata Based Intrusion Detection Method for Internet of Things,” Mobile Information Systems, 2017.
[10] M. Pawlicki, A. Pawlicka, R. Kozik, and M. Choraś, “The survey on the dual nature of xAI challenges in intrusion detection and their potential for AI innovation,” Artificial Intelligence Review, vol. 57, Art. 330, 2024, doi: 10.1007/s10462-024-10972-3.
[11] Y. Zhang, R. C. Muniyandi, and F. Qamar, “A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in Spatiotemporal Feature Extraction and Data Imbalance,” Applied Sciences, vol. 15, no. 3, Art. 1552, 2025, doi: 10.3390/app15031552.
[12] R. Chinnasamy, M. Subramanian, S. V. Easwaramoorthy, and J. Cho, “Deep learning-driven methods for network-based intrusion detection systems: A systematic review,” ICT Express, vol. 11, no. 1, pp. 181–215, 2025, doi: 10.1016/j.icte.2025.01.005.
[13] C. E. Shannon, “A Mathematical Theory of Communication,” Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.
[14] T. Maruf, "Hazard recognition system based on violation of the integrity of the field and changes in the intensity of illumination on the video image," 2022 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2022, pp. 1-3, doi: 10.1109/ICISCT55600.2022.10146933.
[15] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.
[16] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann, 2011. doi: 10.1016/C2009-0-61819-5.
[17] W. Lee and S. J. Stolfo, “A Framework for Constructing Features and Models for Intrusion Detection Systems,” ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 227–261, 2000. doi: 10.1145/382912.382914.

[18] M. Tavallaee et al., “A Detailed Analysis of the KDD CUP 99 Data Set,” in IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009. doi: 10.1109/CISDA.2009.5356528.
[19] Maruf R.Tojiyev, Sanjar S. Kenjaev, Rashid Nasimov, Oybek D.Primkulov and Gulnoza K.Ernazarova, "Quantum computing algorithm for optimizing query distribution based on multi-criteria parameters in information systems," 2025 ICFNDS, in press

Authors’ Profiles



Maruf Tojiyev Ruzikulovich is an Associate Professor at Samarkand State University named after Sharof Rashidov, Uzbekistan. His research interests include network security, intrusion detection systems, computer vision, computational modeling, and fuzzy logic–based intelligent systems.



Bahtiyor Holmuhamedov Farkhodovich obtained his DSc degree from Samarkand State University, Uzbekistan. He is currently working as an Associate Professor in the Department of Uzbek Linguistics at Samarkand State University named after Sharof Rashidov. His research interests include linguistics and interdisciplinary studies related to information systems.



Sofiyaxon Usmonova Alimovna obtained her PhD from Kokand State University, Uzbekistan. She is currently working as an Associate Professor at the Interfaculty Department of Languages, Kokand State University. Her research interests include language studies and academic communication.



Jura Kuvandikov Tursunbayevich is an Associate Professor at the Jizzakh branch of the National University of Uzbekistan named after Mirzo Ulugbek. His research interests include computer science, programming, and network technologies.

How to cite this paper

Maruf Tojiyev Ruzikulovich, Bahtiyor Holmuhamedov Farkhodovich, Sofiyaxon Usmonova Alimovna, Jura Kuvandikov Tursunbayevich, "Real-Time Port Scanning Attack Detection Using Adaptive Entropy Analysis and Random Forest-based Hybrid Model", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.5, pp. 428-445, 2026. DOI:10.5815/ijwmt.2026.05.25