Intrusion detection based on fractal analysis of network traffic anomalies*

Vira Titova[1,†], Yurii Klots[1,†], Nataliia Petliak[1,†], Dmytro Tymoshchuk[2,*,†], and Vitaliy Tymoshchuk[2,†]

[1] *Khmelnytskyi National University, 11 Instytutska str., 29016 Khmelnytskyi, Ukraine*
[2] *Ternopil Ivan Puluj National Technical University, 56 Ruska str., 46001 Ternopil, Ukraine*

Abstract

As a result of the study, a classification of anomaly detection methods in computer networks was conducted. Since network traffic exhibits self-similarity properties, the possibility of assessing traffic anomalies based on the Hurst exponent was established. As a result of the work, a method was implemented, the basis of which is a mechanism for detecting network traffic anomalies using the Hurst exponent. Two algorithms for calculating the Hurst exponent were implemented: the R/S method and a method based on the stationarity-aware analysis of network traffic. During the study, an experimental evaluation of the effectiveness of three anomaly detection methods was also conducted: Autoencoder, Isolation Forest, and online-SVM. For this purpose, the values of the Hurst exponent were used as the sole feature, enabling the assessment of the models' ability to detect different types of attacks on network traffic. Experimental results showed that the use of a stationarity-aware estimation of the Hurst exponent increases detection accuracy compared to the classical R/S approach, especially for attacks with less pronounced or short-term deviations. Thanks to the combination of fractal traffic assessment and modern machine learning methods, the feasibility of using the Hurst exponent as an informative feature for early-warning systems was demonstrated. The developed algorithms allow effective detection of both pronounced and weakly expressed anomalies, and their implementation can be integrated into existing network traffic monitoring systems to enhance security and reduce the number of false positives.

Keywords

network anomaly, normal traffic, abnormal traffic, Hurst exponent, fractal analysis, autoencoder, isolation forest method, online-SVM model, accuracy.

1. Introduction

One of the key attributes of the modern world is the comprehensive integration of information, which is based on the development of enterprise-scale computer networks and their subsequent interconnection through the Internet.

The complexity of the logical and physical organization of modern networks presents objective challenges in addressing issues of network management and protection. When solving problems related to the diagnostics and protection of network resources, the central task is the timely detection of network states that result in the network becoming unusable.

In addressing issues related to the diagnostics and protection of network resources, the primary focus is on the prompt identification of conditions that lead to the complete or partial loss of network functionality, destruction, distortion, or leakage of information. Such consequences may arise from accidental failures or malfunctions, as well as from unauthorized access to network resources by malicious actors, or the penetration of network worms, viruses, and other cybersecurity threats. Early detection of such states enables the timely implementation of countermeasures, thereby preventing potentially catastrophic outcomes.

* *CSDP'2026: Cyber Security and Data Protection, May 07, 2026, Lviv, Ukraine*
* Corresponding author.
† These authors contributed equally.
titovav@khmnu.edu.ua (V. Titova); klots@khmnu.edu.ua (Y. Klots); npetlyak@khmnu.edu.ua (N. Petliak); dmytro.tymoshchuk@gmail.com (D. Tymoshchuk); tymoshchuk@tntu.edu.ua (V. Tymoshchuk)
0000-0001-8668-4834 (V. Titova); 0000-0002-3914-0989 (Y. Klots); 0000-0001-5971-4428 (N. Petliak); 0000-0003-0246-2236 (D. Tymoshchuk); 0009-0007-2858-9434 (V. Tymoshchuk)

A wide range of specialized tools is used to detect such threats [1-3]. For network diagnostics, tools such as management systems, network protocol analyzers, load testing systems, and network monitoring tools are commonly employed. The protection of network information resources is ensured by firewalls, antivirus software, intrusion detection systems, integrity monitoring systems, and cryptographic protection mechanisms. In recent studies, intrusion detection systems have increasingly incorporated intelligent data analysis and explainable artificial intelligence techniques to improve detection accuracy and transparency, particularly in complex and heterogeneous network environments [4-6]. Typical characteristics of these systems include either their periodic and short-term use for solving a specific problem or their constant operation with static configurations. As a result, the analytical methods applied in modern systems are generally aimed at detecting known and well-described types of attacks; however, they often fail to identify modifications or new, previously unknown types, which significantly reduces their effectiveness [7-9].

Therefore, the objective of this research is to develop an intrusion detection system based on methods for detecting anomalies in network traffic. The primary tasks of the study include analyzing existing methods of anomaly detection, classifying threats in information systems, and developing a method capable of effectively identifying anomalies and intrusions in computer networks.

2. Review of Existing Solutions

In recent years, a wide range of approaches to anomaly detection in network traffic has been developed, reflecting the increasing complexity of computer networks and the sophistication of cyber threats. Existing methods can be grouped into behavioral, statistical, machine learning, computational intelligence, and knowledge-based techniques, each offering distinct advantages and limitations depending on the type of monitored system, data characteristics, and detection objectives.

Behavioral methods are based on the assumption that deviations from a model of normal operation indicate potential anomalies [10-12]. Such methods construct a reference profile of normal activity and compare current network behavior against it. Wavelet analysis is a typical example of this class, in which incoming network signals are decomposed into basic components to highlight irregularities of large amplitude while filtering out low-amplitude noise. The main advantage of behavioral methods lies in their conceptual simplicity and their ability to detect a wide range of known irregularities without prior labeling of data. However, they require extensive historical data for building accurate profiles, and they perform poorly in dynamic environments where normal behavior changes frequently [13-14].

Statistical methods extend behavioral analysis by incorporating formal probabilistic models that describe normal network traffic distributions [15-17]. These models can be parametric, assuming a known distribution such as Gaussian, or non-parametric, estimating probability densities directly from data using histogram or kernel functions. A further development of this approach is fractal analysis, which leverages the self-similarity of network traffic to quantify long-range dependencies using the Hausdorff dimension and the Hurst exponent. Statistical and fractal methods have the advantage of being mathematically interpretable and requiring minimal prior knowledge about anomaly characteristics. Nonetheless, their effectiveness strongly depends on the accuracy of the chosen probabilistic model and the quality of the data. Determining appropriate detection thresholds also remains a significant challenge, as excessively high thresholds lead to false negatives, while low thresholds generate numerous false positives.

Machine learning methods have become increasingly popular for anomaly detection due to their ability to adapt and self-improve [18-20]. Recent studies have demonstrated the effectiveness of supervised and ensemble-based machine learning models for detecting and classifying DDoS flooding attacks and other high-volume anomalies in network traffic [21]. Unlike traditional statistical models, machine learning systems are capable of learning complex, nonlinear

relationships from historical data. Depending on the presence or absence of labeled data, these methods may employ supervised, unsupervised, or semi-supervised learning paradigms. During the training phase, models such as support vector machines, Bayesian networks, and rule-based systems learn to distinguish between normal and abnormal patterns, which they later apply to classify new traffic samples. The key advantage of machine learning methods is their ability to generalize from data and detect previously unseen attacks. However, they require large volumes of high-quality training data, involve significant computational costs, and may produce inconsistent results when applied to data outside their training domain.

Computational intelligence methods integrate concepts such as neural networks, evolutionary algorithms, and hybrid models [22, 23]. Neural networks can capture nonlinear dependencies and adapt to complex patterns in traffic, while evolutionary algorithms, such as genetic algorithms or genetic programming, search for optimal solutions through iterative adaptation. The combination of these methods often produces robust and flexible detection systems capable of identifying previously unknown attacks. Graph-based neural architectures, such as graph attention networks, have recently been applied to model complex relationships between network entities and detect coordinated multi-vector attacks with high accuracy [24]. The primary strength of computational intelligence lies in its adaptability and tolerance to noise; however, these methods also demand substantial computational resources and training time. Additionally, their stochastic nature may lead to inconsistent performance across different network environments.

Knowledge-based methods rely on predefined sets of rules, patterns, or attack signatures compiled from expert knowledge or historical data [1, 25]. Signature-based systems compare network traffic with databases of known attack patterns, such as specific byte sequences or protocol behaviors. In contrast, rule-based systems employ logical conditions that describe abnormal activity. Such systems are efficient and highly accurate in detecting known threats, with minimal computational overhead. Their major limitation, however, is their inability to detect novel or modified attacks that do not match existing signatures. Frequent updates to the knowledge base are required, and systems of this type are prone to false negatives when facing zero-day exploits or polymorphic malware.

Across these diverse approaches, fractal analysis has emerged as a promising direction for detecting anomalies in network traffic. Empirical studies have shown that network flows often exhibit self-similarity, meaning their structural properties remain consistent across multiple time scales [26, 27]. The effectiveness of fractal and statistical traffic analysis strongly depends on the quality of extracted features, and comparative studies of feature extraction tools confirm that appropriate preprocessing significantly influences anomaly detection performance [28].

The Hurst exponent, which quantifies the degree of long-range dependence and self-similarity in time series data, serves as a reliable metric for identifying deviations from normal traffic patterns. Experiments conducted on benchmark datasets demonstrate that abnormal traffic significantly alters the self-similarity structure, producing detectable variations in the Hurst exponent [29, 30].

Thus, fractal analysis, and particularly the application of the Hurst exponent, represent a promising and theoretically grounded direction for future research.

3. Fractal Analysis–Based Anomaly Detection Method

To assess the fractal properties of various processes, the Hurst exponent (H) has become widely used, which is directly related to the Hausdorff dimension (D_f) [26].

$$D_f = 2 - H \tag{1}$$

The Hurst exponent is defined in terms of the asymptotic behavior of the scaled range as a function of the time segment of the time series as follows:

$$E[\frac{R(n)}{S(n)}]=Cn^H \quad (2)$$

where $R(n)$ is the range of accumulated deviations of the first n values from the mean of the series; $S(n)$ is standard deviation; E is mathematical expectation; n is the number of points in the time series segment; C is a constant.

Using the Hurst exponent, we can determine the dimensionality of the probability space. From this, we obtain the following variations.

If $0 \le H < 0.5$, the time series exhibits anti-persistent behavior, meaning that an increase is more likely to be followed by a decrease, and vice versa. We have an inverse relationship between the current state and time. The stability of the system depends on how close the Hurst exponent is to 0.

If $H = 0.5$, then we have a random series. Current values do not affect future values.

If $0.5 < H \le 1$, then the series is trend-stable. The series maintains a direct relationship.

The Hurst exponent is defined on the interval [0, 1].

The Hurst exponent can be estimated using several methods, including the R/S method, the periodogram method, and wavelet analysis [31-33]. However, in some cases, other methods are used.

The advantage of using the Hurst exponent is its speed of data processing, even with large series sizes. However, this also has a disadvantage: large sample sizes are required to obtain reliable results; small sample sizes can lead to errors. In the conducted experiments, this limitation was mitigated by computing the Hurst exponent on high-resolution traffic sequences aggregated within each time window, ensuring a sufficient number of observations for stable estimation even for short window durations.

Calculating the Hurst exponent using the R/S method consists of two stages: benchmark calculation and real-time anomaly detection. To assess anomalies, information from the packet headers of the OSI model's transport and network layers is used for one second, including the source/destination port numbers, source/destination IP addresses, the number of IPv4 packets, and the payload size.

The data calculated during the benchmark calculation stage is compared with the data calculated in real time during the detection stage. The method uses the Hurst exponent to assess anomalies. The result is two arrays with four values: the Hurst exponent values for different time intervals and their standard deviation. The R/S method is used to estimate the Hurst exponent. As is well known, the basic formula for RS analysis is the ratio:

$$R/S=(N/2)^H \quad (3)$$

where H — Hurst exponent; S — standard deviation of the observation series; R — cumulative deviation range; N — discrete time (sample size).

Finding the value of exponent H can be broken down into the following steps:

1. Divide the original time series into blocks of equal length.
2. Calculate the range R for each block.
3. Calculate the standard deviation for each block.
4. Take the sample size (discrete time) as N.
5. Logarithmize the resulting expression.
6. Obtain the desired exponent:

$$H=\frac{\log(R/S)}{\log(N/2)} \quad (4)$$

Knowing the duration of the standard calculation stage — 4 minutes — we can use the following interval splitting scheme for the calculation: every 5 seconds, every 15 seconds, every 60 seconds, and every 120 seconds.

The "every n" seconds scheme means that the entire time interval, equal to four minutes (240 seconds), is split into intervals of n seconds, and the Hurst exponent are calculated within each interval. Thus, for the "every 5 seconds" position, we obtain forty-eight intervals, and for each of them, we calculate the Hurst exponent and the average value of these exponent. Similarly, for the “every 15 seconds” position, we obtain 16 intervals; for the “every 60 seconds" position, 4 intervals; and for the "every 120 seconds" position, 2 intervals.

After calculating the benchmarks, a comparison is made during the actual search for anomalies with the benchmarks for traffic received in 5 seconds, 15 seconds, 60 seconds, and 120 seconds.

After receiving real-time values (5-second intervals), the data is stored in a temporary variable. Each of the four Hurst exponent values is recalculated. The deviation size is then checked. If the deviation is outside the three-sigma range, a network fault warning is issued.

The three-sigma threshold was adopted as a heuristic baseline, as the empirical distribution of the Hurst exponent within normal traffic windows was observed to be close to Gaussian, making this rule suitable for distinguishing significant deviations while maintaining a low false alarm rate.

The algorithm for calculating the Hurst exponent based on the stationarity properties of traffic is as follows [34]. Let X_i be the number of bits, bytes, or packets detected during the *i*-th time interval. A given sequence X_i is second-order stationary if its expectation $E(X_i)$ is independent of *i*, and its autocovariance function depends only on the difference $k=i-j$:

$$CovX = E[(X_i - E(X_i))(X_j - E(X_j))] \tag{5}$$

Next, the autocovariance itself can be replaced by $\gamma(k)$. We denote the variance as:

$$\sigma^2 = \gamma(0) = E[(X_i - E(X_i))^2] \tag{6}$$

Then the normalized correlation function of the sequence will have the following form:

$$p(k) = \frac{\gamma(k)}{\sigma^2} \tag{7}$$

A stationary process is self-similar of the second order with the Hurst exponent $0<H<1$ if its correlation coefficient has the form:

$$p(k) = \frac{1}{2}\left(|k+1|^{2H} - 2|k|^{2H} - |k-1|^{2H}\right) \tag{8}$$

Since X_i is a self-similar sequence of the second order, we obtain:

$$p(1) = 2^{2H-1} - 1 \tag{9}$$

By solving this equation, we can obtain H:

$$H = \frac{1}{2}(1 + \log(1 + p(1))) \tag{10}$$

Next, to calculate the Hurst exponent in real time, we will use traffic sequences over a 5-second interval. The packet count will be used to assess anomalies. Other metrics, such as those used in the method above, can also be used. As a result, we will obtain a sequence of values for the following time intervals: every 5 seconds, every 15 seconds, every 60 seconds, and every 120 seconds.

Next, we calculate the corresponding required parameters: the mean, the sample covariance, and the autocorrelation function. Based on these, we calculate the Hurst exponent and check that the value is within acceptable limits.

To calculate the Hurst exponent, it is advisable to use anomaly detection methods such as an Autoencoder, Isolation Forest, or online-SVM, since the presence of anomalous observations in a

time series significantly distorts the estimate of the Hurst index. This exponent reflects the degree of long-term dependence and the structure of variations in the data, therefore its accuracy depends on the stability of the statistical properties of the series. The use of anomaly detection methods allows us to pre-clean the data from outliers, noise, or structural breaks that violate the natural autocorrelation structure of the process. The Autoencoder effectively detects nonlinear deviations due to the ability to reconstruct normal patterns, the Isolation Forest quickly identifies single and atypical values, and the online SVM provides adaptive processing of streaming data. Thus, the combination of the Hurst exponent calculation with anomaly detection methods provides a more reliable, stable, and interpretable estimate of the long-term properties of time series.

Let us calculate the Hurst exponent for each time series window (X_i):

$$H(X_i) \in R \tag{11}$$

The Autoencoder takes this scalar as its only input:

$$\begin{aligned} z_i &= f_\theta(H(X^i)) = \sigma(W_e * H(X^i) + b_e) \\ \hat{H}^i &= g_\phi(z_i) = \sigma(W_d * z_i + b_d) \end{aligned} \tag{12}$$

where z_i — is the latent representation, f_θ, g_ϕ — are the encoder and decoder parameters, σ - is the activation function.

The reconstruction error is defined as the distance between the original and reconstructed Hurst values:

$$e_i = \left| (H(X^i) - \hat{H}^i) \right| \tag{13}$$

For all windows, we write the total loss function of the Autoencoder:

$$L_{AE} = \frac{1}{N} * \sum_{i=1}^{N} (H(X^i) - \hat{H}^i)^2 \tag{14}$$

Here $e_i$serves as an anomaly exponent for *H*. Exceeding three sigma range non-standard or anomalous Hurst exponents in the corresponding windows:

$$IF \left| (\hat{H}^i - H(X^i)) \right| > 3\sigma \rightarrow Anomaly\ warning \tag{15}$$

Thus, the Autoencoder is trained to reconstruct only the Hurst exponent, and the reconstruction error is used to detect anomalies in its values.

The mathematical implementation of Isolation Forest for the Hurst exponent is as follows. We choose windows of length L and calculate the Hurst exponent for each. So, we have a set of scalars.

$$H = h(X_1), h(X_2), \ldots, h(X_N) \tag{16}$$

The Isolation Forest method is based on the idea that anomalies are isolated faster than normal observations. For each tree $t = 1, \ldots, T$ a subset $H_t \subset H$ of size ψ is randomly selected. An isolation tree is constructed, where at each node a threshold p_t is randomly selected in the range $[min(H_t), max(H_t)]$. The data is split before each h_i is isolated.

For each h_i the path length (number of nodes to isolation) in each tree is determined: $h_t(h_i) = path\ length\ of\ h_i in\ tree\ t$.

Average insulation length:

$$\bar{h_t}(h_i) = \frac{1}{T} \sum_{t=1}^{T} h_t(h_i) \tag{17}$$

Anomaly score for each value of the Hurst exponent:

$$s(h_i)=2^{-\frac{\bar{h}(h_i)}{c(\psi)}} \tag{18}$$

Let us determine the mean and standard deviation of the scores:

$$\mu_s=\frac{1}{N}*\sum_{i=1}^{N} s(h_i),\ \sigma_s=\sqrt{\frac{1}{N}*\sum_{i=1}^{N}(s(h_i)-\mu_s)^2} \tag{19}$$

An object is considered anomalous if: $s(h_i)>\mu_s+3\sigma_s$

The Hurst exponent H_i is used as the sole input feature for the model. The Isolation Forest algorithm isolates these exponent values and determines the degree of anomaly they exhibit. If the resulting score values fall outside the three-sigma range from the mean, this signals a possible system failure, attacks, or structural change in the process under study.

The set of input observations of the Hurst exponent for Online-SVM is $H=\{h_1,h_2,\dots,h_N\}$

Online-SVM learns incrementally, updating the hyperplane as each new value h_i. In our case, the radial basis function (RBF) is used as the kernel for online-SVM, as it effectively maps the Hurst exponent values into the feature space, allowing for the detection of nonlinear dependencies between normal and abnormal states of the system.

When a new observation h_t arrives, the weight vector is updated. After each update, a decision is made on the new value of the Hurst exponent:

$$f(h_t)=\boldsymbol{w}_t^T\phi(h_t)+b_t \tag{20}$$

If $|f(h_t)-\mu_f|>3\sigma_f$ then the value $H(X^t)$is considered anomalous.

Hurst exponent $H(X^i)$ is the only feature that characterizes the long-term memory or chaotic nature of the system. Online-SVM gradually updates the boundary between normal and abnormal states. Output values of the function $f(h_t)$ that go beyond the three sigma limits signal an anomaly or potential process failure.

4. Experimental Verification of the Developed Method

For testing, three traffic flows were organized, each containing 1000 templates, of which 500 were normal, and 500 had signs of a particular attack class. In total, three attacks were tested: DDoS, port scanning, and C2-beaconing. The attack selection was as follows.

DDoS (distributed denial of service). During a DDoS, the traffic of the targeted resource suddenly and steadily increases – long-term, correlated load spikes occur. This strengthens the long-term dependence of the signal (traffic shows a trend/persistence), so the applied value of the Hurst exponent usually increases and a sharp jump in the time series serves as an indicator of an attack.

Port scanning (massive port scanning). Scanners generate numerous short pulses with a high frequency and a minimum duration for each request – this makes the signal more "noisy" and less correlated over long ranges. In such cases, a decrease in long-term dependence is observed, i.e. the Hurst exponent tends to fall, which allows us to distinguish intensive scanning from smoother normal traffic.

C2-beaconing. When an attacker uses regular, periodic, or very slow but repetitive transmissions to extract data or communicate, persistent cyclical components or new correlations emerge at specific scales over time. This leads to local changes in the correlation structure – affected windows may exhibit an increase or a characteristic pattern in the Hurst exponent, or the appearance of anomalous dependence between scales. The detection of such a shift, in combination with a periodic signal spectrum, indicates a possible stealthy exfiltration or botnet backdoor.

The general requirement for the satisfactoriness of solving the problem performed by the developed method can be expressed using well-known and common classification quality

indicators [35, 36]: TP (True Positive) is the number of cases in which the method correctly identified anomalous network traffic (attack) as an anomaly; FP (False Positive) is the number of cases in which normal network traffic was incorrectly classified as anomalous; TN (True Negative) is the number of cases in which normal network traffic was correctly recognized as normal; FN (False Negative) is the number of cases in which anomalous network traffic (attack) was not detected and was incorrectly classified as normal. The classic synonym of FP is errors of the first kind, and FN is errors of the second kind. In classical statistical terminology, false positive errors (FP) correspond to errors of the first kind (Type I errors), which lead to false alarms, whereas false negative errors (FN) correspond to errors of the second kind (Type II errors), resulting in missed detections of anomalous events.

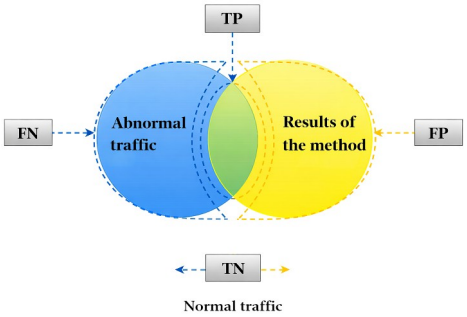

Figure 1: Graphical interpretation of performance metrics of the developed method.

Figure 1 illustrates a schematic representation of the sets: the set of real anomalies (attacks) is marked with a red circle. In contrast, the set of traffic detected by the method as anomalous is marked with a green circle. The white area surrounding the circles corresponds to normal traffic. The intersection of the red and green circles corresponds to correctly detected anomalies (True Positive, TP). The absence of circles in the corresponding area (white area) means the correct identification of the absence of attacks (True Negative, TN). The part of real anomalies that did not fall into the green circle are omissions of the method (False Negative, FN – errors of the second kind). The part of normal traffic that was mistakenly attributed by the method to anomalous forms is referred to as false positives (False Positive, FP – errors of the first kind, or "false alarm"). The test results are presented in Tables 1-3.

Table 1
Comparison of attack detection results for Autoencoder

Traffic type	Hurst exponent							
	R/S				CoV			
	TP	FN	FP	TN	TP	FN	FP	TN
Normal + DDoS	437	63	48	452	456	44	46	454
Normal + Port Scan	318	182	63	437	361	139	56	444
Normal + C2-beacon	279	221	39	461	332	168	44	456

Table 2
Comparison of attack detection results for Isolation Forest

Traffic type	Hurst exponent							
	R/S				CoV			
	TP	FN	FP	TN	TP	FN	FP	TN
Normal + DDoS	476	24	42	458	484	16	37	463
Normal + Port Scan	431	69	77	423	448	52	71	429
Normal + C2-beacon	198	302	31	469	243	257	36	464

Table 3
Comparison of attack detection results for online-SVM

Traffic type	Hurst exponent							
	R/S				CoV			
	TP	FN	FP	TN	TP	FN	FP	TN
Normal + DDoS	462	38	33	467	474	26	28	472
Normal + Port Scan	362	138	52	448	401	99	48	452
Normal + C2-beacon	348	152	51	449	419	81	53	447

After analyzing the results obtained, we obtain the following accuracy indicators for each of the implementations.

For the DDoS attack, the accuracy of the Autoencoder based on the R/S method is 88.90%, while with stationarity-aware, it is 91.00%. For port scanning and C2-beaconing, the results are as follows: 75.50% and 80.50%; 74.00% and 78.80%. The final average accuracy over three attacks for the Autoencoder is 81.45%.

The Isolation Forest method has the following accuracy indicators compared to the R/S method for DDoS – 93.40%, for Port Scan – 85.40%, and for C2-beacon – 66.70%. With stationarity-aware, the accuracy is 94.70%, 87.70%, and 70.70%, respectively. The final average accuracy over three attacks for the Isolation Forest is 83.10%.

The online-SVM has the following accuracy indicators with the R/S method for DDoS – 92.90%, for Port Scan – 81.00%, and for C2-beacon – 79.70%. With stationarity-aware, the accuracy is 94.60%, 85.30%, and 86.60%, respectively. The final average accuracy over three attacks for the online-SVM is 86.68%.

In this study, the Autoencoder used the Hurst exponent as the sole feature for reconstruction; therefore, its effectiveness largely depended on the model's ability to reproduce the temporal structure of changes in H and to respond to anomalous deviations in this indicator. For the Hurst exponent calculated using the classical R/S method, the average accuracy reached 79.47%, indicating the model's fundamental ability to distinguish typical deviations in the presence of pronounced anomalies, such as DDoS attacks. However, its efficiency is limited to less intensive variations, such as those typical of port scanning or stealth exfiltration. The highest result was observed for DDoS (88.9%), where the structure of H showed a stable upward trend, which was well captured during the reconstruction process.

When using the stationary-aware Hurst exponent, the average accuracy increased to 83.43%, reflecting an improvement in the Autoencoder's adaptation to minor structural changes in the correlation behavior of the signal. Performance improvements were observed across all attack types, particularly for port scanning (80.5%) and stealth exfiltration (78.8%), indicating higher sensitivity to short-term violations of statistical equilibrium. This enhancement demonstrates that eliminating trend effects and accounting for local stationarity reduces the number of false deviations during reconstruction, making the model more stable with respect to natural traffic fluctuations.

Overall, the Autoencoder demonstrated moderate effectiveness in the task of anomaly detection based on the Hurst exponent. Its average accuracy (81.45%) was lower than that of other methods due to the limited informativeness of a scalar input; however, the model remains suitable for detecting large deviations accompanied by significant changes in the long-term correlation structure of network traffic.

This is attributable to the fact that the architecture of the Autoencoder, which focuses on minimizing reconstruction error, is more responsive to distinctly expressed structural changes. In contrast, the detection of subtle and short-term deviations in the Hurst exponent requires a more contextual representation or supplementation with additional statistical features.

In the conducted experiments, the Isolation Forest algorithm was applied to the Hurst exponent values as an unsupervised method for anomaly detection. For the classical Hurst exponent

calculated using the R/S method, the average accuracy was 81.83%, which indicates the model's stable ability to identify significant deviations associated with large-scale changes in correlations within network traffic. The highest efficiency of the method was demonstrated during the detection of DDoS attacks (93.4%), when the H value increased significantly due to pronounced traffic persistence, which facilitated the isolation of anomalous windows in the feature space. For port scanning, the result remained sufficiently high (85.4%). In contrast, for stealth exfiltration, the accuracy decreased to 66.7%, indicating the method's limited sensitivity to weakly expressed or periodic changes that do not create distinct outlier points.

When using the stationarity-aware Hurst exponent, the average accuracy increased to 84.37%, which indicates improved consistency between the isolation estimates of points and their correlation behavior. The best results were again observed for DDoS (94.7%) and port scanning (87.7%), confirming the model's suitability for detecting clearly expressed disruptions in traffic structure. At the same time, even with the adjusted exponent, the efficiency in the case of stealth exfiltration remained lower (70.7%), since this type of attack is characterized by minimal changes in long-term dependencies that do not produce statistically significant outliers.

In general, the Isolation Forest demonstrated high stability and the ability to detect significant or abrupt changes in the Hurst exponent effectively, but its sensitivity to small and gradual anomalies was lower compared to other methods. The model's average accuracy was 83.10%, which reflects a balance between speed, implementation simplicity, and overall efficiency when working with single-feature inputs.

This is because the Isolation Forest focuses on detecting statistically isolated observations and is most effective in cases where anomalies have a pronounced deviation from the primary distribution. In contrast, weakly expressed structural changes of a correlational nature require models with higher sensitivity to contextual dependencies.

During the experiments, the Online-SVM model utilized the Hurst exponent as an input feature for sequential observations. For the Hurst exponent calculated using the classical R/S method, the average accuracy reached 84.53%, indicating the model's ability to maintain stable classification performance under sequential data analysis conditions. The best results were observed for DDoS attacks (92.90%), where anomalies manifest as sustained changes in long-term correlations, while lower accuracy values were obtained for port scanning (81.00%) and C2-beaconing (79.70%), which are characterized by weaker or short-term deviations in the correlation structure.

When using the stationarity-aware estimation of the Hurst exponent, the average accuracy increased to 88.83%, reflecting improved consistency between the evolving correlation profile of network traffic and the decision boundaries formed by the classifier. In this case, accuracy improved across all attack types, reaching 94.60% for DDoS, 85.30% for port scanning, and 86.60% for C2-beaconing, which confirms the higher sensitivity of the model to subtle structural changes when local stationarity is taken into account.

Overall, Online-SVM demonstrated the highest effectiveness among the considered methods, achieving an overall average accuracy of 86.68% across both classical and stationarity-aware Hurst exponent estimations. This indicates a high degree of stability and flexibility of the model in detecting both pronounced and weak anomalies in the time series of the Hurst exponent.

This is because the architecture of Online-SVM with the RBF kernel function allows for the adaptive formation of nonlinear boundaries between normal and anomalous states, providing gradual model updates according to changes in the statistical properties of traffic, which makes it the most suitable for dynamic network environments.

5. Conclusions

As a result of this study, the following results were obtained. Methods for anomaly detection were considered, and their classification, advantages, and disadvantages were presented. Several groups of anomalies can be distinguished: point, contextual, and collective. Detection methods can be divided

into major groups: behavioral methods, machine learning methods, computational intelligence methods, and knowledge-based methods.

More detailed consideration and implementation were given to anomaly detection methods, taking into account fractal analysis. The Hurst exponent is used as a measure for evaluating traffic anomalies. For the calculation of anomaly, three approaches were applied: Autoencoder, Isolation Forest, and online-SVM, which allowed assessing their ability to detect different types of attacks based on structural and correlation changes in network traffic. Experimental results showed that methods taking into account local stationarity when calculating the Hurst exponent provide increased detection accuracy compared to the classical R/S approach.

Among the considered methods, online-SVM demonstrated the highest accuracy, indicating the effectiveness of the adaptive approach with an RBF kernel for detecting both pronounced and weakly expressed anomalies. The Autoencoder demonstrated moderate effectiveness when working with a scalar representation of the Hurst exponent; however, its accuracy increased when using stationarity-aware estimates. The Isolation Forest demonstrated stable performance for sharply expressed anomalies, such as DDoS attacks, but was less sensitive to small-scale or periodic deviations.

Overall, the study confirmed that combining fractal traffic assessment through the Hurst exponent with modern anomaly detection methods allows the creation of effective early-warning systems in computer networks. The results can be utilized for constructing multi-level anomaly detectors by combining various methods to enhance accuracy and minimize false alarms, as well as for adaptive monitoring of changes in network traffic behavior.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to grammar and spell check, and improve the text readability. After using the tool, the authors reviewed and edited the content as needed to take full responsibility for the publication's content.

References

[1] Y. Kumar, V. Kumar, A Systematic Review on Intrusion Detection System in Wireless Networks: Variants, Attacks, Appl. Wirel. Pers. Commun., 133, 395-452, 2023. doi:10.1007/s11277-023-10773-x.

[2] I. Opirskyy, et al., Detection of network attacks based on the redhunt virtual machine, in: Advancements in cybersecurity, CRC Press, Boca Raton, 2025, pp. 310-330. doi:10.1201/9781003546153-14.

[3] M. Issa, M. Aljanabi, H. Muhialdeen, Systematic literature review on intrusion detection systems: Research trends, algorithms, methods, datasets, and limitations. J. Intell. Syst., 33, 20230248, 2024. doi:10.1515/jisys-2023-0248.

[4] D. Tymoshchuk, et al., An explainable artificial intelligence approach for detecting network attacks. CEUR Workshop Proc., 2025, pp. 38-51.

[5] O. Harasymchuk (ed.): Modern Methods of Ensuring Information Protection in Cybersecurity Systems Using Artificial Intelligence and Blockchain Technology. TECHNOLOGY CENTER PC, Kharkiv, 2025. doi:10.15587/978-617-8360-12-2.

[6] Y. Klots, V. Titova, N. Petliak, D. Tymoshchuk, N. Zagorodna, Intelligent data monitoring anomaly detection system based on statistical and machine learning approaches. CEUR Workshop Proc., (2025), pp. 80-89.

[7] D. Kus, et al., A false sense of security? Revisiting the state of machine learning-based industrial intrusion detection. Proc. 8th ACM CyberPhysical Syst. Secur. Workshop (CPSS '22), 73-84, 2022. doi:10.1145/3494107.3522773.

[8] R. Sikorskyi, O. Harasymchuk, A multi-stage model for detecting and preventing SQL injections in the XAMPP/MySQL environment. CEUR Workshop Proc., 2025, pp. 224-237.

[9] A. Khraisat, A. Alazab, A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecur., 4, 18, 2021. doi:10.1186/s42400-021-00077-7.

[10] D. Kim, G.-Y. Shin, M.-M. Han, Anomaly Detection Based on Discrete Wavelet Transformation for Insider Threat Classification. Comput. Syst. Sci. & Eng., 46(1), 153-164, 2023. doi:10.32604/csse.2023.034589.

[11] P. Hrynchenko, Detection of Unauthorized Actions in Networks Using Wavelet Analysis. Igor Sikorsky Kyiv Polytechnic Institute. pp. 1-12, 2023.

[12] D. O'Shea, et al., A Wavelet-Inspired Anomaly Detection Framework for Cloud Platforms. In: Proceedings of the 12th International Conference on Cloud Computing and Big Data, pp. 106-117, 2016.

[13] N. Petliak, Y. Klots, M. Karpinski, V. Titova, D. Tymoshchuk, Hybrid system for detecting abnormal traffic in IoT. CEUR Workshop Proc., 2025, 4057, pp. 21-36.

[14] N. Zagorodna, et al., Network attack detection using machine learning methods, Chall. Natl. Def. Contemp. Geopolit. Situat. 2022.1, 2022, 55-61. doi:10.47459/cndcgs.2022.7.

[15] C. Ding, Y. Chen, Z. Liu, A. Alshehri, T. Liu, T., Fractal characteristics of network traffic and its correlation with network security. Fractals, 30(2), 1-11, 2022, doi:10.1142/S0218348X22400679.

[16] J. Rosenberger, K. Müller, A. Selig, M. Bühren, D. Schramm, Extended kernel density estimation for anomaly detection in streaming data. Procedia CIRP, 112, 156-161, 2022. doi:10.1016/j.procir.2022.09.065.

[17] M. Ahsan, H. Khusna, Wibawati, M. Lee, Support vector data description with kernel density estimation (SVDD-KDE) control chart for network intrusion monitoring. Sci. Reports, 13(1), 2023. doi:10.1038/s41598-023-46719-3.

[18] V. Priya, S. Soumya, Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning. J. Sci. Res. Technol., 2(6), 38-48, (2024). doi:10.61808/jsrt114.

[19] V. Titova, Y. Klots, V. Cheshun, N. Petliak, A.-B. Salem, Detection of Network Attacks in Cyber-Physical Systems Using a Rule-Based Logical Neural Network. In ICyberPhyS-2024: 1st International Workshop on Intelligent and CyberPhysical Systems. CEUR Workshop Proc., vol. 3736, 2024.

[20] T. Assani, B. Aditya, B. Likhitha, G. Govardhan Reddy, Anomaly detection in network traffic using machine learning techniques. Glob. J. Eng. Innov. Interdiscip. Res., 5(4), 2025, doi:10.33425/3066-1226.1129.

[21] D. Tymoshchuk, et al., Detection and classification of DDoS flooding attacks by machine learning method. CEUR Workshop Proc., 2024, 3842, pp. 184-95.

[22] Y. Klots, V. Titova, N. Petliak, V. Cheshun, A.-B. M. Salem, Research of the neural network module for detecting anomalies in network traffic. In: IntelITSIS'2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 23-25, 2022, Khmelnytskyi, Ukraine. CEUR Workshop Proc., vol. 3156, paper 28.

[23] M. Khan, A. Reshi, S. Shafi, I. Aljubayri, An adaptive hybrid framework for IIoT intrusion detection using neural networks and feature optimization using genetic algorithms. Discov. Sustain., 6, Article 382, 2025. doi:10.1007/s43621-025-01141-9.

[24] M. Stetsiuk, Y. Klots, D. Tymoshchuk, M. Karpinski, N. Petliak, Detection of multi-vector attacks in IoT networks: a graph attention network-based approach. CEUR Workshop Proc., 2025, 4146, pp. 296-313.

[25] M. Almseidin, M. Alkasassbeh, S. Kovacs, Intrusion detection mechanism using fuzzy rule interpolation, 2019. doi:10.48550/arXiv.1904.08790.

[26] P. Dymora, M. Mazurek, Anomaly Detection in IoT Communication Network Based on Spectral Analysis and Hurst Exponent. Appl. Sci., 9(24), 5319, 2019. doi:10.3390/app9245319.

[27] P. Dymora, M. Mazurek, Influence of Model and Traffic Pattern on Determining the Self-Similarity in IP Networks. Appl. Sci., 11(1), 190, 2021. doi:10.3390/app11010190.

[28] B. Lypa, et al., Comparison of feature extraction tools for network traffic data, CEUR Workshop Proc., 3896, 2024, pp. 1-11.
[29] L. Kirichenko, Y. Koval, S. Yakovlev, D. Chumachenko, Anomaly Detection in Fractal Time Series with LSTM Autoencoders. Math., 12(19), 3079, 2024. doi:10.3390/math12193079.
[30] M. Fernández-Martínez, A survey on fractal dimension for fractal structures. (2016). Appl. Math. Nonlinear Sci., 1(2), 437-472. doi:10.21042/AMNS.2016.2.00037.
[31] M. Knight, G. Nason, M. Nunes, A wavelet lifting approach to long-memory estimation. Stat. Comput., 27(6), 1453-1471, 2017. doi:10.1007/s11222-016-9698-2.
[32] A. Hamza, M. Hmood, Comparison of Hurst exponent estimation methods. J. Econ. Adm. Sci., 27(128), 2021. doi:10.33095/jeasi.v27i128.2162.
[33] L. Wu, A Note on Wavelet-Based Estimator of the Hurst Parameter. Entropy, 22(3), 349, 2020. doi:10.3390/e22030349.
[34] G. Millán, Estimation of Hurst Exponent in Self-similar Traffic Flows, 2021. doi:10.48550/arXiv.2103.08592.
[35] Y. Klots, N. Petliak, S. Martsenko, V. Tymoshchuk, I. Bondarenko, Machine Learning system for detecting malicious traffic generated by IoT devices. In: 2nd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2024), Ternopil, Ukraine (pp. 97-110). CEUR Workshop Proc., vol. 3742.
[36] M. Moharam, O. Hany, A. Hany, et al., Anomaly detection using machine learning and adopted evaluation metrics. Sci. Reports, 15, 18352, 2025. doi:10.1038/s41598-025-02759-5.
[37] Y. Kostiuk, et al., Intelligent System for Simulation Modeling and Research of Information Objects, In: 1st Workshop Software Engineering and Semantic Technologies (SEST), vol. 4053, 2025, 237-251.
[38] P. Skladannyi, et al., Model and Methodology for the Formation of Adaptive Security Profiles for the Protection of Wireless Networks in the Face of Dynamic Cyber Threats, in: Workshop Cyber Security and Data Protection (CSDP), vol. 4042, 2025, 17-36.
[39] J. Hutchins, D. Ferrer, J. Fillers, C. Schuman, Black-Box Adversarial Attacks on Spiking Neural Network for Time Series Data, In: Int. Conf. on Neuromorphic Systems (ICONS), Arlington, VA, USA, 2024, 229-233. doi:10.1109/ICONS62911.2024.00040.
[40] O. Partyka, A. Partyka, A. Imoize, AI for Web Application Security. Handb. Cybersecur. Challenges Solutions Emerg. Technol, 2026. doi:10.1201/9781003640790-14.
[41] I. Opirskyy, O. Partyka, A. Partyka, A. Imoize, AI for Wireless Network Protection. Handb. Cybersecur. Challenges Solutions Emerg. Technol., 2026. doi:10.1201/9781003640790-15.