COMPUTER SCIENCE

Research article
УДК 004.021, 004.056
DOI: 10.17072/1993-0550-2026-1-72-91
https://elibrary.ru/pvxuen

Statistical Analysis of Time Series for Port Scan and DDoS Detection

Adeyemi Marc Aurele Emmanuel Djeguede
RUDN University, Moscow, Russia
djeguede.marc@gmail.com

Abstract. In this paper, statistical methodologies for time series analysis – specifically the Z-score and the modified Z-score – are examined in the context of detecting Port Scan and Distributed Denial of Service (DDoS) attacks. Six different time series were constructed using the following traffic characteristics: the average number of packets transmitted from sources to destinations, the data transfer rate, the response data transfer rate, the duration of the connection between the source and destination, the entropy computed based on destination ports associated with each IP source, and the number of unique destination ports available to each IP source. To evaluate the statistical methodologies under study, the indicators such as reliability, accuracy, response time, and F1-score were used. The obtained numerical results show that when detecting the network threats in question, the modified Z-score reduces the number of false positives compared to the Z-score standard, thereby influencing the evaluation of these performance metrics. The F1-scores achieved using the modified Z-score for DDoS detection ranged from 93% to 98%, depending on the specific traffic characteristics analyzed. Conversely, in the case of Port Scan detection, the F1-score did not exceed 58% even under optimal conditions. A comprehensive analysis showed that all the identified Port Scan instances refer to fast port scanning since this scanning method causes a sharp increase in network traffic. This phenomenon is manifested in a local violation of the stationarity of the time series. These findings were confirmed by Augmented Dickey-Fuller (ADF) and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) statistical tests conducted to evaluate various hypotheses regarding the stationarity of the time series.

Keywords: *time series analysis; anomaly detection; Port Scan; DDoS; Z-score.*

For citation: Djeguede, A. M. A. E. (2026), "Statistical Analysis of Time Series for Port Scan and DDoS Detection", *Bulletin of Perm University. Mathematics. Mechanics. Computer Science*, no 1(72), pp. 72−91, DOI: 10.17072/1993-0550-2026-1-72-91, https://elibrary.ru/pvxuen.

The article was submitted 11.08.2025; approved after reviewing 18.11.2025; accepted for publication 10.03.2026.

КОМПЬЮТЕРНЫЕ НАУКИ И ИНФОРМАТИКА

Научная статья

Статистический анализ временных рядов для обнаружения Port Scan и DDoS

Адейеми Марк Аурeле Эммануэль Джегюеде

Российский Университет Дружбы Народов, Москва, Россия
djeguede.marc@gmail.com

Аннотация. В ходе этой научной статьи статистические методологии анализа временных рядов, в частности Z-оценка и модифицированная Z-оценка, были рассмотрены в контексте обнаружения атак Port Scan и распределенного отказа в обслуживании (DDoS). Было построено шесть различных временных рядов с использованием следующих характеристик трафика: среднее количество пакетов, передаваемых от источников к получателям, скорость передачи данных от источника к получателю, скорость передачи ответных данных, продолжительность соединения между источником и адресатом, вычисленная энтропия на основе портов назначения, связанных с каждым IP-источником, и количество уникальных портов назначения, доступных каждому IP-источнику. Для оценки вышеупомянутых статистических методологий использовались показатели надежности, точности, времени отклика и показатели F1. Полученные численные результаты показывают, что модифицированная Z-оценка снижает количество ложных срабатываний (по сравнению с Z-оценкой) при выявлении исследованных сетевых угроз, что влияет на оценку этих показателей. Результаты измерения F1, полученные с использованием модифицированной Z-оценки при обнаружении DDoS-атак, варьировались от 93 до 98% в зависимости от конкретных проанализированных характеристик трафика. И наоборот, показатель F1 в контексте обнаружения Port Scan в самом оптимальном случае не превышает 58%. Комплексный анализ показал, что все экземпляры, выявленные Port Scan, относятся к категории быстрого сканирования портов, поскольку именно этот метод сканирования приводит к резкому увеличению сетевого трафика. Это явление проявляется в локальном нарушении стационарности временных рядов. Эти выводы были подтверждены статистическими тестами Дики–Фуллера (ADF) и Квятковского–Филлипса–Шмидта–Шина (KPSS), проведенными для оценки различных гипотез относительно стационарности временных рядов.

Ключевые слова: *анализ временных рядов; обнаружение аномалий; Port Scan; DDoS; Z-score.*

Для цитирования: Джегюеде А. М. А. Э. Статистический анализ временных рядов для обнаружения Port Scan и DDoS // Вестник Пермского университета. Математика. Механика. Информатика. 2026. № 1(72). С. 72−91, DOI: 10.17072/1993-0550-2026-1-72-91. https://elibrary.ru/pvxuen.

Статья поступила в редакцию 11.08.2025; одобрена после рецензирования 18.11.2025; принята к публикации 10.03.2026.

Introduction

Intrusion detection systems (IDS) play a vital role in safeguarding computer networks by continuously monitoring and analyzing network traffic to identify potential security threats. As network environments grow more complex and data volumes expand, traditional static detection methods often fail to recognize sophisticated or evolving cyberattacks. Time series analysis offers a dynamic alternative by focusing on how network behaviors change over time. By treating indicators such as connection rates, packet flows, and data volumes as temporal

data, IDS can detect subtle irregularities or evolving trends that may indicate malicious activity. This temporal perspective enhances the system's ability to identify both sudden and gradual attacks, paving the way for more adaptive and intelligent intrusion detection mechanisms.

Port scanning is a common reconnaissance technique used by attackers to discover open ports and available services on a target system. Detecting such activity is crucial for maintaining network security as scans often precede more severe intrusions. Traditional detection methods, which rely on predefined rules or threshold limits, may fail to capture slow or stealthy scans designed to evade detection. Time series analysis provides a stronger alternative by examining temporal patterns and anomalies in connection attempts. By modeling these attempts as time-dependent data, it becomes possible to identify deviations from normal behavior that signal scanning activity, even when it occurs gradually or covertly. This approach not only improves detection accuracy but also enables earlier and more proactive responses to potential threats.

Distributed Denial-of-Service (DDoS) attacks remain a persistent threat to online services, aiming to overwhelm targeted infrastructures – such as servers, networks, or applications – with massive amounts of malicious traffic, thereby denying access to legitimate users. Traditional signature-based detection systems often struggle to cope with new or rapidly changing attack patterns. Time series analysis (TSA) offers a data-driven alternative by analyzing the temporal dynamics of network traffic, allowing for the identification of subtle or emerging anomalies that may signal the onset of a DDoS attack.

1. Literature review

The centralized architecture of software-defined networks (SDN) makes them ideal targets for flood attacks such as DDoS and port scanning. Addressing this issue, G. F. Scaranti et al. [1] proposed an intrusion detection system (IDS) based on online clustering to detect attacks in evolving SDN networks by leveraging the entropy of the source and destination IP addresses and ports. The proposed system eliminates the need for data labeling, paving the way for comprehensive analysis by projecting cluster structures into the feature space and providing information on the intensity, seasonality, and type of various attacks. The system is built on the DenStream algorithm [1], utilizing multiple databases targeted by DDoS and port scanning attacks with varying intensity and duration. C. Birkinshaw et al. in [2] also designed an SDN-based Intrusion Detection and Prevention System (IDS/IPS). The proposed system is a software application that monitors the network for malicious activity or security policy violations and takes measures to mitigate such activity. Special emphasis is placed on protection against port scanning and Denial-of-Service (DoS) attacks. In their study, the authors described and tested the Port Bingo (PB) algorithm as a defense mechanism against port scanning and implemented two connection-based methods: Credit-Based Threshold Random Walk (CB-TRW) and Rate Limiting (RL). Botnets are responsible for some of the most significant malicious attacks on the internet: DDoS attacks, email spam, brute-force attacks, port scanning, and others. Their danger stems from the coordinated actions of infected hosts targeting a single objective. R. Abrantes et al. in their article [3] focused on identifying botnet traffic to prevent communication between the botmaster and infected hosts. For analyzing hosts in the Botnet2014 dataset, they employed the CICFlowMeter algorithm and machine learning methods – Random Forest (RF) and Decision Tree (CART). The results show that the analysis scenario using IP addresses and L4 ports achieves higher accuracy but a lower F1-score compared to equivalent scenarios without IP addresses or L4 ports. Detection of low-frequency attacks requires additional computational overhead compared to regular traffic. In [4], D. Ono et al. proposed a method for detecting port scanning by analyzing the characteristics of Packet-In messages sent from an OpenFlow (OF) switch to the controller. Port scanning typically generates a large volume of Packet-In messages. The proposed method monitors the flow rate of Packet-In messages sent by each host

to the switch, identified by their addresses. Upon detecting an abnormal increase in this rate, the controller requests statistics from the switch and implements an algorithm based on the collected data to identify port scanning. The employed algorithm significantly reduces computational costs compared to conventional methods. Port scanning, commonly used as a reconnaissance tool in attacks, can create significant performance and bandwidth challenges for applications. In [5], B. Hartpence et al. describe an architecture of recurrent neural networks (RNN) for packet classification, TCP datagram separation, TCP packet type identification, and port scanning detection. Recurrent neural networks enable detecting temporal dependencies in prolonged port scanning sequences that unfold over time. Testing the proposed model in this work with real NMAP application pcap files demonstrated successful detection of open ports and scanning attempts with high accuracy and a low false-positive rate. Q. Abu Al-Haija et al. in [6] proposed a novel inclusive port scanning detection scheme that evaluates five machine learning classifiers, including logistic regression, decision trees, linear/quadratic discriminant analysis, naive Bayes, and ensemble boosted trees. Studies conducted on the modern PSA-2017 dataset demonstrated the best performance for the logistic regression-based detection scheme, achieving 99.4% accuracy, 99.9% precision, 99.4% recall, 99.7% F-score, and a detection time of 0.454 μsec. SNORT is an intrusion detection and prevention system (IDS/IPS) and a popular tool for monitoring network traffic in real-time and performing rule-based packet analysis. These rules act as signatures for various types of attacks. Each packet passing through SNORT is thoroughly analyzed to identify matches with predefined rules. In their work [7], M. Almseidin et al. propose a novel approach for detecting slow port scanning using a Fuzzy Rule Interpolation (FRI) rule set, which also determines the maliciousness level of detected attacks. These rules are based on the following parameters:

- number of packets sent between the source and destination;
- average Time between Packets received by the victim, in milliseconds;
- number of Packets Received by the destination (victim) per second.

The majority of approaches proposed in the literature for detecting slow port scanning are focused on identifying slow port-scanning attacks within a static period. Mehr u Nisa et al. [8] proposed a technique to detect slow port-scanning attacks not only during static time intervals but all attacks conducted with a gradual increase or decrease in duration over time. The proposed system is divided into four main modules. In the first module, real-time data packets are captured from a live network for analysis. In the second module, the captured data is analyzed to detect signs of port scanning and labeled accordingly. The third module categorizes the labeled packets into parallel and single scans based on the scanner's IP address and other selected features. Finally, in the last module, a decision is made based on time duration analysis to determine whether the scan was a fast or slow attack. The generated reports can then be used to block the attacker' IP address or take other necessary measures. E. S. Sagatov et al. [9] presented methods for detecting and countering the initial stages of cyberattacks, including TCP and UDP port scanning. The proposed methods analyze outgoing traffic to identify response packets such as ICMP 3.3 and TCP RST, which indicate the onset of an attack. The authors also described two countermeasures based on developed modules for software-defined network controllers and Linux OS utilities. Testing of the developed methods was conducted on a cybersecurity testbed and demonstrated that the accuracy of detecting open TCP ports did not exceed 15%, while for other ports (closed TCP ports and UDP ports of any type), the accuracy remained below 2%. E. K. Baah [10] employed seven machine learning classifiers for port scan detection after successfully applying the Principal Component Analysis (PCA) algorithm for reducing dimensionality and selecting the most relevant features. A comparison of the results from various models and prior studies identified the XGBoost model as the best classifier, achieving the highest accuracy of 99.98%, with no false positives detected, a precision of 99.99%, a recall of

99.98%, and an area under the curve (AUC) of 99.99%. M. Ring et al. [11] propose an innovative approach to preprocessing streaming data, designed to detect slow port scanning. The preprocessing process generates new objects based on domain knowledge and information on the network structure collected over a specific period. The computed objects are used as input data for further analysis; based on these, two distinct approaches for detecting slow port scans were proposed. One approach employs sequential hypothesis testing, while the other utilizes classification algorithms. The proposed methods were tested on the CIDDS-001 dataset.

2. Materials

The **CIC-IDS-2017** dataset is one of the most widely used cybersecurity datasets for evaluating **IDS**. The Canadian Institute **for Cybersecurity (CIC)** created it at the **University of New Brunswick (UNB)**. The data was captured using **realistic user behavior** and **attack scenarios**, therefore it contains both benign and malicious activities such as botnets and web attacks.

The data capturing period started at 9 a.m., Monday, July 3, 2017 and ended at 5 p.m. on Friday July 7, 2017, for a total of 5 days. The daily traffic classification and attacks were presented in Table 1.

Table 1. *CIC-IDS-2017 daily traffic classification*

Day	Type of traffic	Attack types present
Monday	Normal traffic only	None
Tuesday	Brute Force attacks	SSH Brute Force, FTP Brute Force
Wednesday	DoS and Heartbleed	DoS Hulk, DoS GoldenEye, Heartbleed
Thursday	Web and Infiltration	Web Attack (XSS, SQLi, Cmd Injection), Infiltration
Friday	Botnet and Port Scan	Botnet, Port Scan, DDoS

In this work, I will explore time series analysis techniques used to detect Port Scan and DDoS attacks.

2.1. Port scan attacks. By performing port scans, attackers can gather information about a host's port numbers, operating system, and running applications by sending specific data to network ports and analyzing the responses. For example, a simple port scan can be performed by a method called a TCP scan, which checks whether a port is in use by attempting a 3-way handshake on the target port. However, since a 3-way handshake is performed even in standard TCP connections, it is difficult to distinguish between a port-scan attack and regular communication on a packet-by-packet basis. Therefore, it is necessary to extract features from multiple packets in order to identify port scans, such as the number of port accesses per unit of time from the same host to different hosts or the amount of traffic per port. There are many behaviors characterizing a scanning activity, such as:

- **numerous connection attempts**: a scanning mechanism dispatches a multitude of connection requests directed toward various ports on a designated machine;
- **sequential or patterned approach**: ports are typically examined in a sequential manner (for instance, 1, 2, 3, 4…) or according to a predetermined pattern (prioritizing common ports, e.g., 22, 80, 443);
- **transient connections**: connections are frequently of a fleeting nature – they are sufficiently brief to ascertain whether the port is open, closed, or filtered;
- **anomalous traffic volume**: relative to normative behavior, port scanning engenders an elevated number of connection attempts, frequently within a condensed period;

- **unexpected origin**: scanning activities predominantly originate from external or unrecognized IP addresses.

There can be distinguished different categories of scans:

- **SYN scan (half-open):** transmits SYN packets while abstaining from completing the TCP handshake;
- **FIN, XMAS, NULL scans:** dispatch unconventional TCP flags to circumvent firewalls or to discreetly identify ports;
- **UDP scan**: since UDP has **no handshake**, traditional scanning (like SYN scan for TCP) doesn't work. Instead, a UDP scan sends an **empty or protocol-specific packet** to a port and analyzes the response.

Depending on variations in scanning timing, scans can be divided into:

- **aggressive** (rapid scans with heightened likelihood of detection);
- **stealthy** (deliberate scans over extended periods to mitigate the risk of detection).

2.2. Distributed Denial of Service (DDoS) attacks. DDoS attacks remain a persistent threat in cybersecurity, evolving in scale and sophistication. Proactive defense strategies, combined with rapid incident response plans, are essential for minimizing their impact. Unlike a traditional Denial-of-Service (DoS) attack, which originates from a single source, a DDoS attack leverages a distributed network of compromised devices (a botnet) to amplify its impact, making mitigation more challenging.

DDoS has a variety of attack vectors, such as:

- **volume-based attacks:** flood the target with high traffic volumes to exhaust bandwidth. Examples include **UDP/ICMP floods**, which consist in sending spoofed UDP or ICMP packets, and **amplification attacks** exploiting protocols such as **DNS** or **NTP** to magnify traffic by triggering large responses to small requests.
- **protocol attacks: o**verwhelm the TCP handshake process, leaving connections half-open (**SYN floods**), or send malformed packets to crash systems (**Ping of Death**).
- **application-layer attacks:** target specific applications (e.g., web servers) with resource-intensive requests, **mimicking** legitimate user traffic to overload servers (**HTTP floods**, e.g., repeated page requests) or keeping server connections open indefinitely to exhaust resources (**Slowloris**).

3. Methods

Effective anomaly detection is essential for maintaining the security and stability of network infrastructure. One method for identifying unusual network behavior is through time series analysis. In this study, I have analyzed the time series corresponding to the features listed in Tables 2 and 3, which were derived from the columns of the CIC-IDS-2017 dataset.

Table 2. *Selected features for DDoS detection*

Features	Descriptions
Mean number of forward packets per source IP	Average number of packets transmitted from sources to receivers
Forward speed	Data transfer rate
Backward speed	Response data transfer rate

Table 3. *Selected features for Port Scan detection*

Features	Descriptions
Connection lifetime	Duration of connection between source and destination
Destination ports entropy per source IP	Computed entropy based on destination ports of each IP source
Unique destination ports count by source IP	Count of unique destination ports connected by each IP source

To examine these time series and identify potential anomalies, I apply a statistical Z-score computation to each *i*-th point using a sliding window of size *n*. The Z-score is calculated as follows:

$$Z_i := \frac{x_i - \mu(w_i)}{\sigma_i(w_i)}, \tag{1}$$

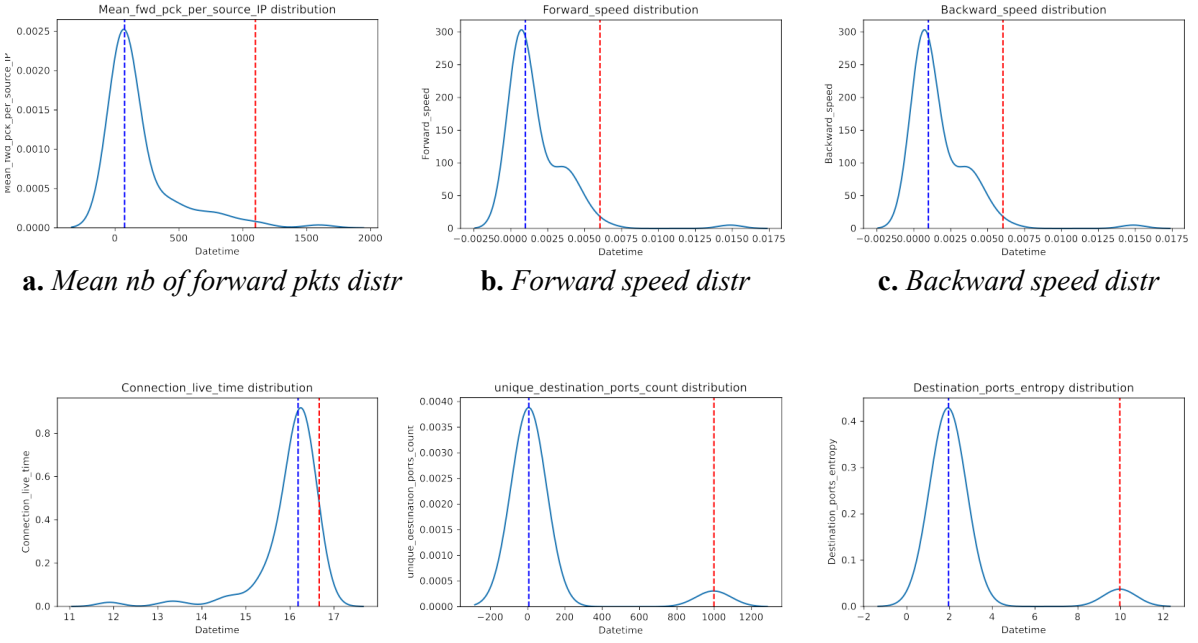
where the local mean

$$\mu(w_i) = \frac{1}{d} \cdot \sum_{j=i-d}^{i-1} x_j \tag{2}$$

and

$$\sigma(w_i) = \sqrt{\frac{1}{d} \cdot \sum_{j=i-d}^{i-1} \left[x_j - \mu(w_i)\right]^2} \tag{3}$$

is the standard deviation. The primary assumption underlying the use of the Z-score is that the analyzed features follow a normal distribution.



a. *Mean nb of forward pkts distr* **b.** *Forward speed distr* **c.** *Backward speed distr*

d. *Connection lifetime distr* **e.** *Unique dest. ports count distr* **f.** *Destination ports entropy distr*

Fig. 1. *Feature distribution functions*

An empirical study of the distribution of these features is shown in the graphs in Fig. 1. These distributions are near to Gaussian distribution, therefore the decision to use the Z-score in this work is justified. Below is a sample algorithm, based on the Z-score, to detect outliers in time series.

Algorithm 1. Detection of anomalies in time series

1: $T = \{x_1, \cdots, x_n\}$ – time series

2: d – size of window

3: t – threshold

4: **for** $i \in \{1, \cdots, n\}$ **do**

5: select window $w_i = \{x_{i-d}, x_{i-d+1}, \ldots, x_{i-1}\}$

6: compute Z-score $Z_i := \frac{x_i - \mu(w_i)}{\sigma(w_i)}$

7: **if** $|Z_i| \geq t$ **then**

8: label x_i as anomaly

9: **else**

10: label $x_i$as benign

11: **end if**

12: **end for**

Due to several factors that will be discussed in the next section, the algorithm based on Z-score calculations with sliding windows may prove inefficient. In such cases, I employ the modified Z-score, which can be calculated as follows:

$$Z_{modified_{score}} = \frac{0.6745 * x_i - median}{mad}, \tag{4}$$

where MAD stands for median absolute deviation and can be calculated as follows:

$$mad = \sqrt{\frac{1}{N} \cdot \sum_{j=i-N}^{i-1} \left(x_j - median\right)^2}. \tag{5}$$

4. Results and discussion

This section presents the results of the experimentation. Figs. 2−7 depict the time series obtained from the features for DDoS detection described in Table 2; Figs. 8−13 show the representation of the time series obtained from the features for Port Scan detection presented in Table 3. In these figures, green points on the time series indicate actual threats (DDoS and Port Scan), whereas red points indicate predicted threats. Points with overlapping colors indicate correctly predicted threats. A quick visual inspection of these results reveals that the modified Z-score algorithm achieves a high level of DDoS detection, as evidenced by the large number of bicolored points in this case. More formal performance evaluation metrics are presented.

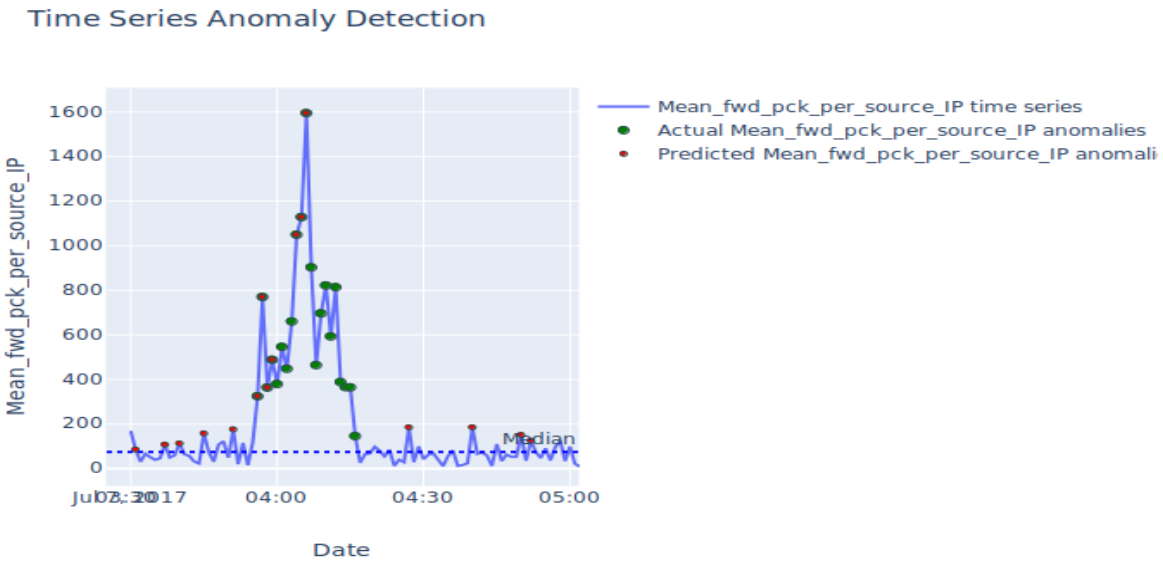


Fig. 2. *Z-score with mean forward packets time series (DDoS)*

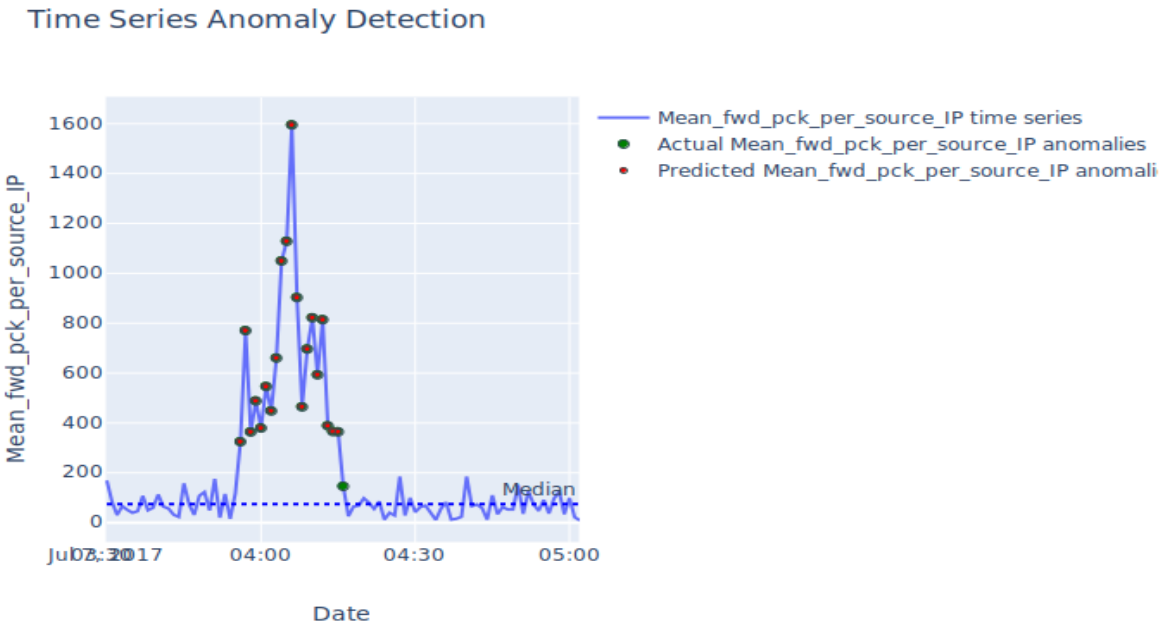


Fig. 3. *Modified Z-score with mean forward packets (DDoS)*

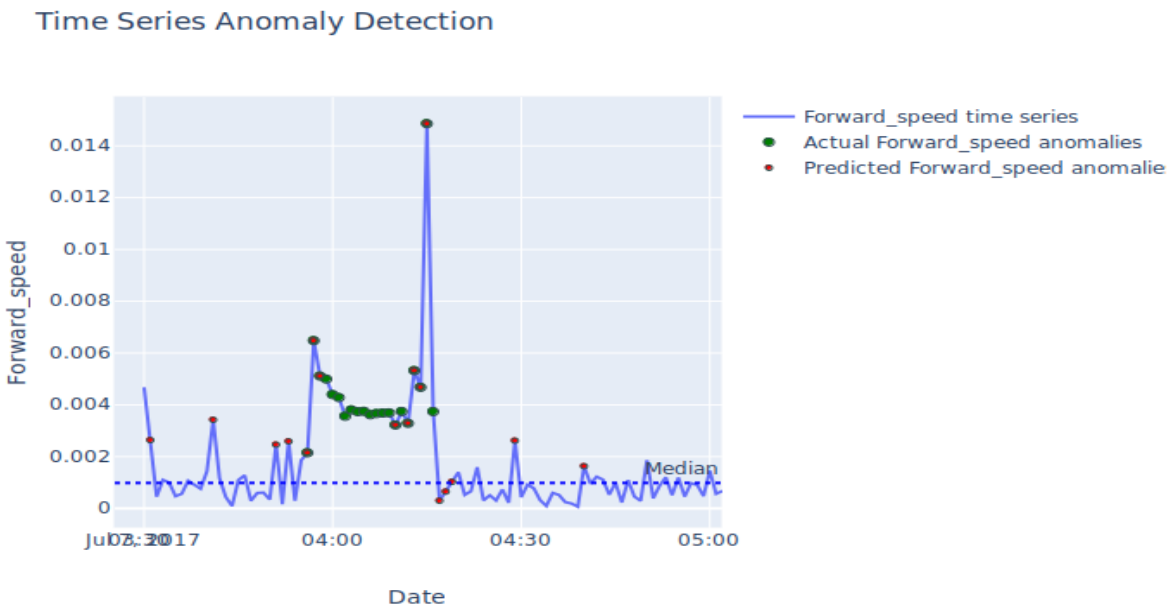


Fig. 4. *Z-score with forward speed time series (DDoS)*

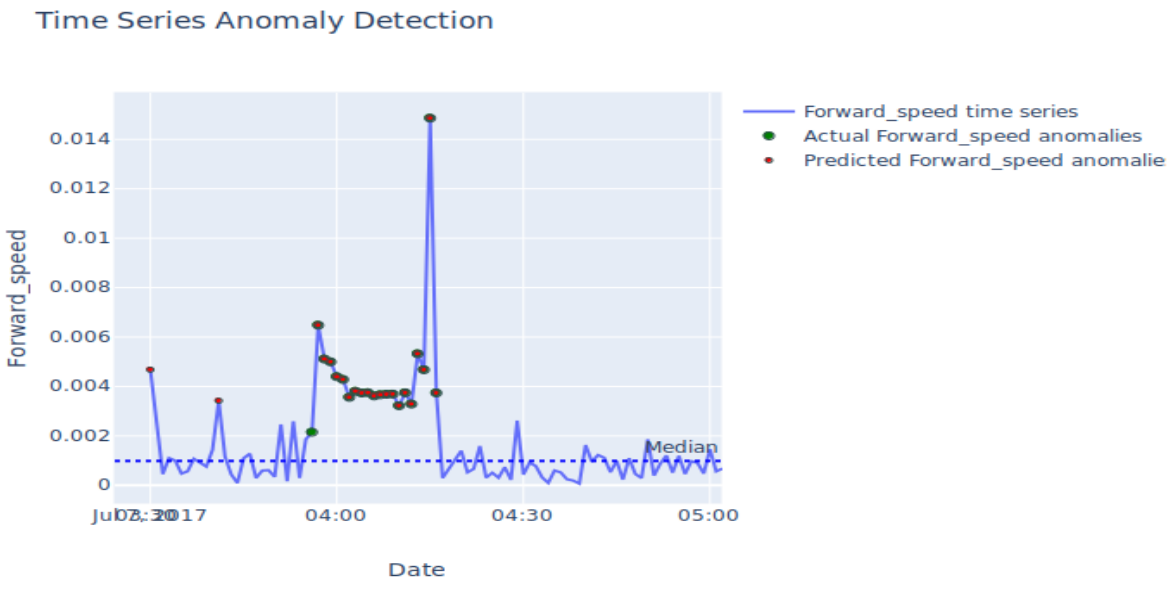


Fig. 5. *Modified Z-score with forward speed time series (DDoS)*

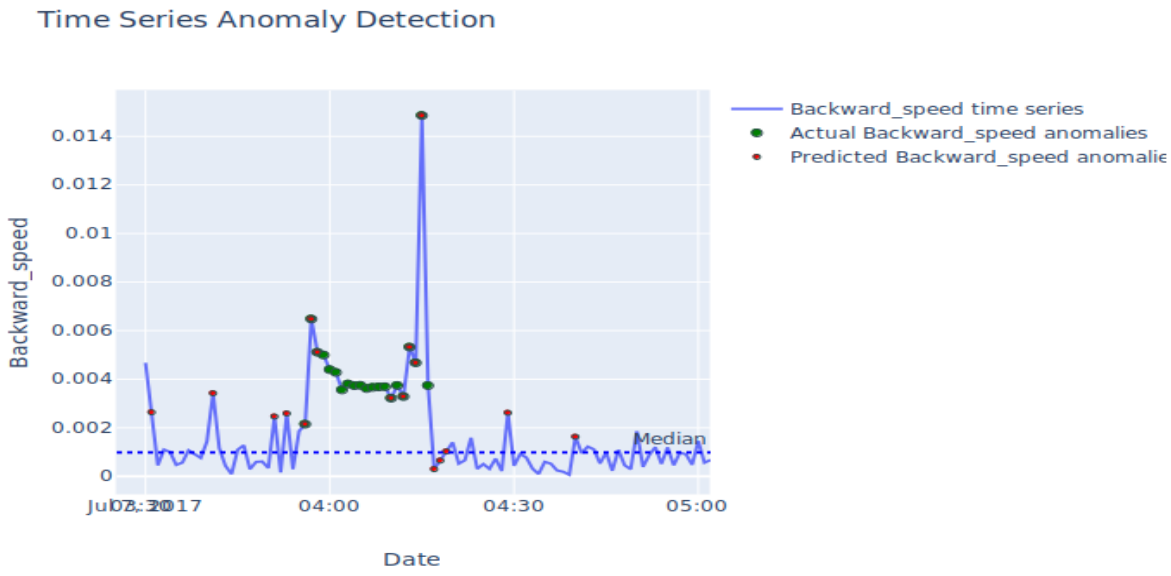


Fig. 6. *Z-score with backward speed time series (DDoS)*

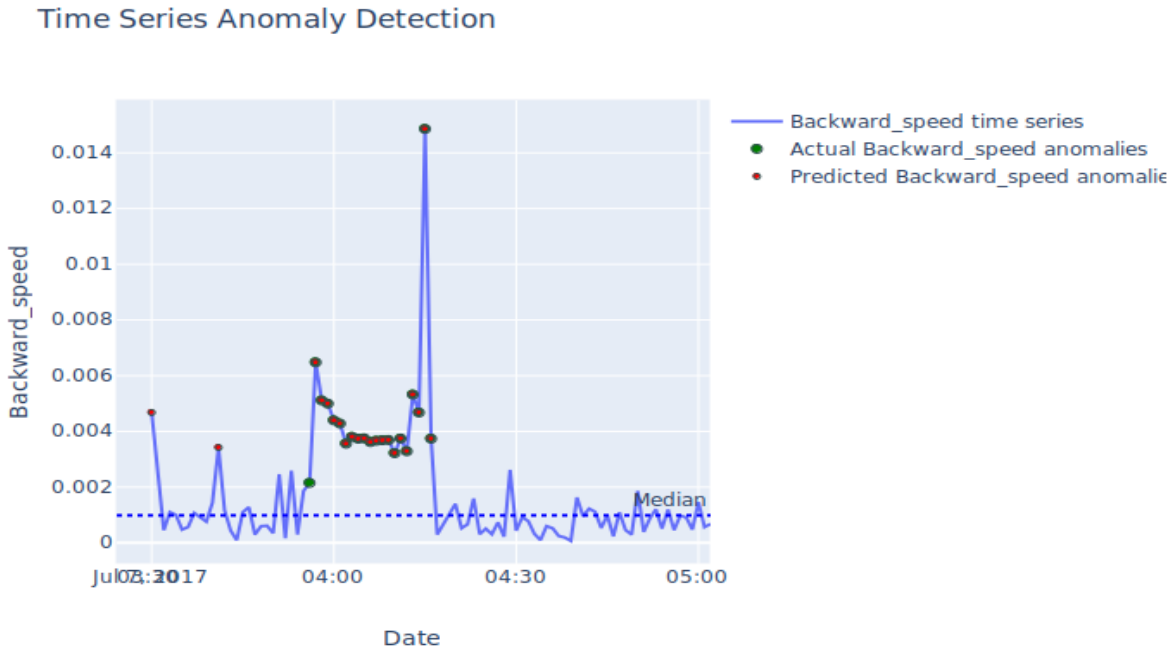


Fig. 7. *Modified Z-score with backward speed time series (DDoS)*

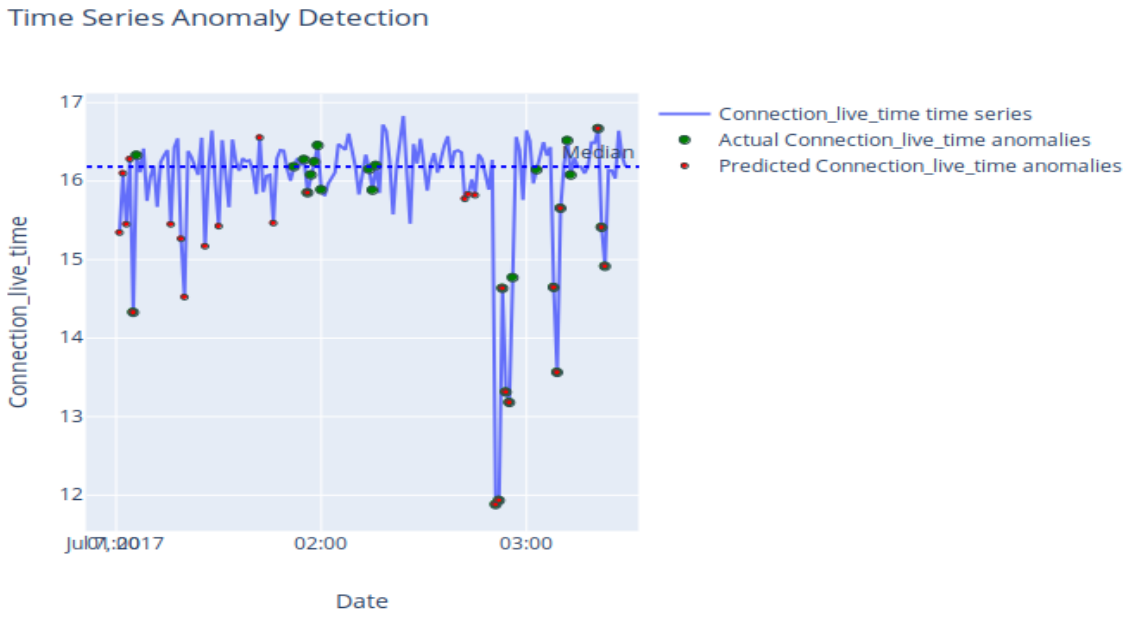


Fig. 8. *Z-score with connection lifetime time series (Port Scan)*

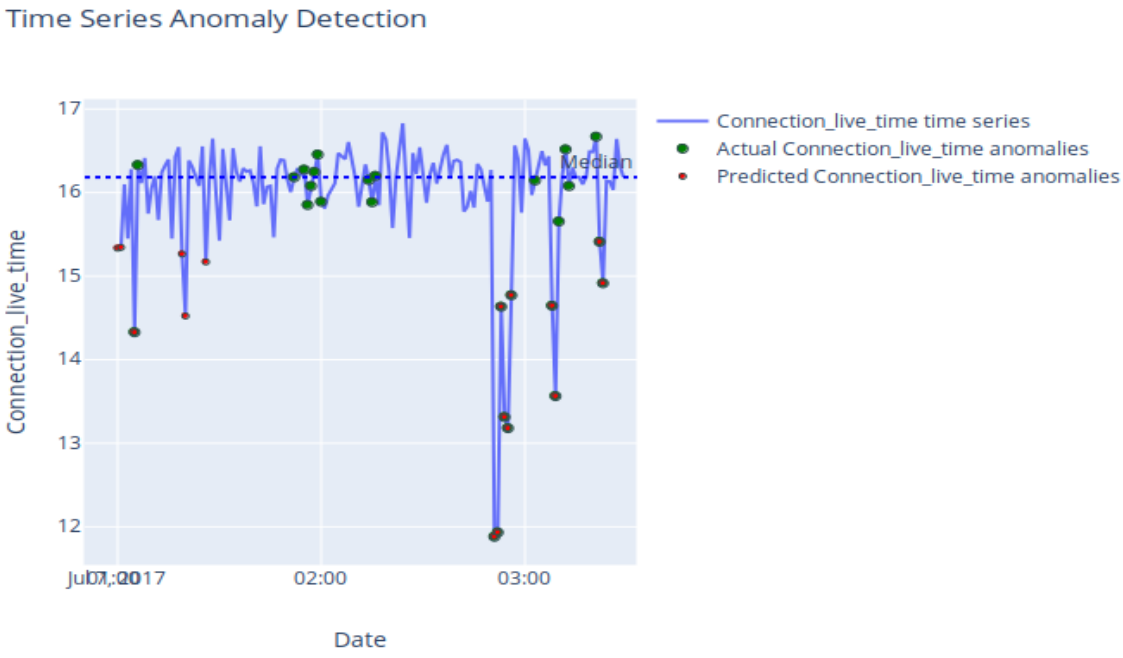


Fig. 9. *Modified Z-score with connection lifetime time series (Port Scan)*

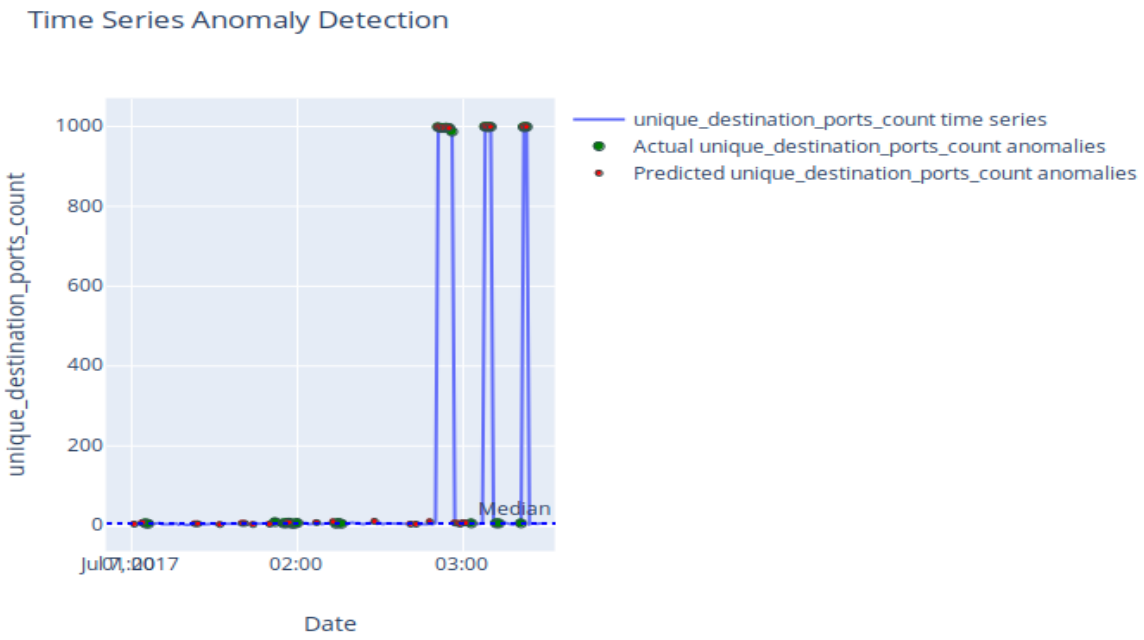


Fig. 10. *Z-score with unique destination ports count (Port Scan)*

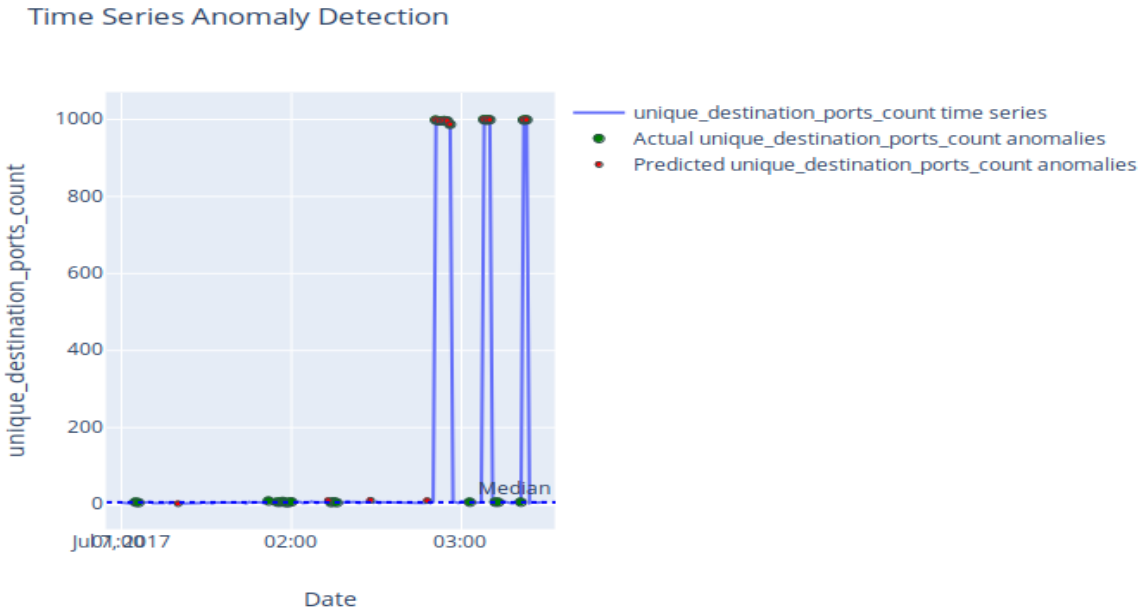


Fig. 11. *Modified Z-score with unique destination ports count (Port Scan)*

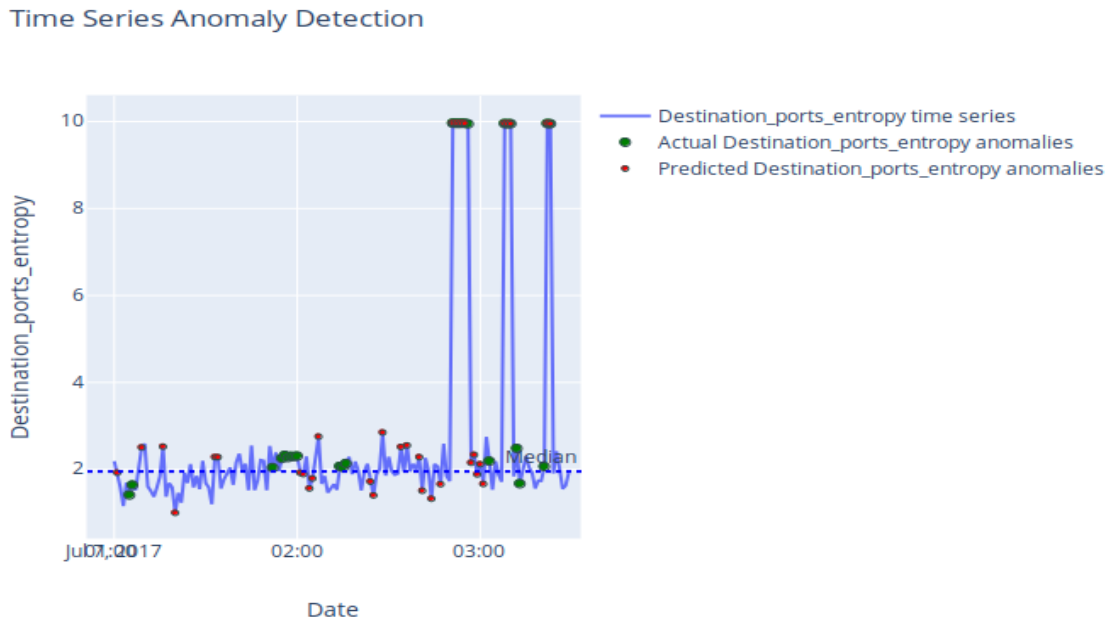


Fig. 12. *Z-score with destination ports entropy (Port Scan)*

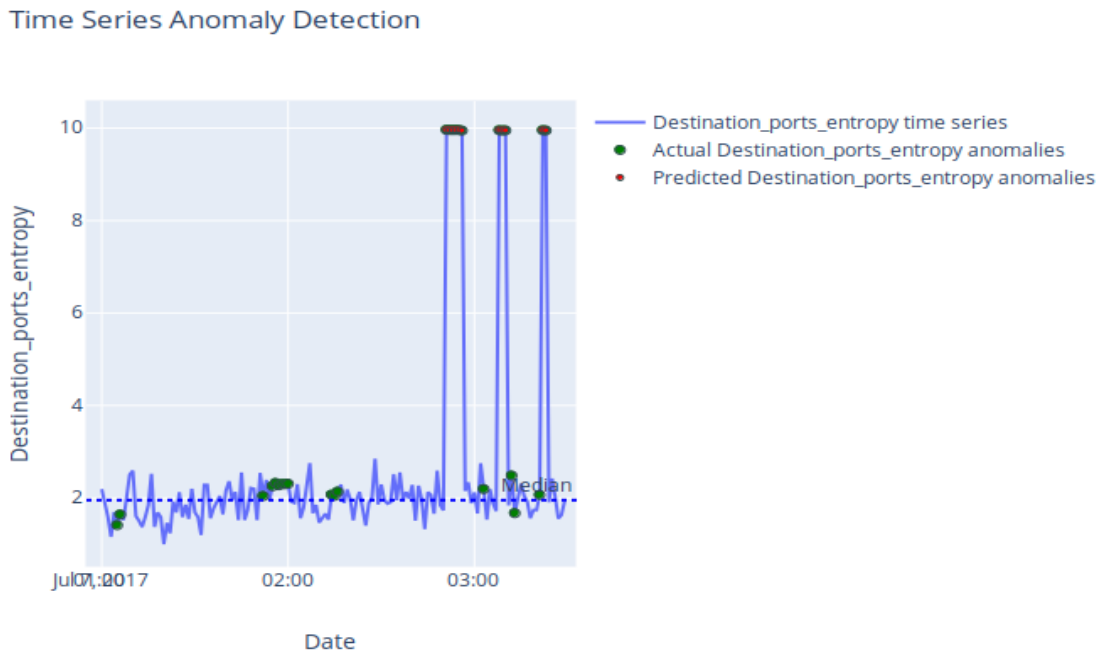


Fig. 13. *Modified Z-score with destination ports entropy (Port Scan)*

For a more detailed assessment of the performance of the two methods, I will use a confusion matrix (whose structure is explained in) to compute the metrics – accuracy, precision, recall, and F1 – using the following formulas:

$$Accuracy = \frac{TruePositives+TrueNegatives}{TotalPredictions}, \quad (6)$$

$$Precision = \frac{TruePositives}{TruePositives+FalsePositives}, \quad (7)$$

$$Recall = \frac{TruePositives}{TruePositives+FalseNegatives}, \quad (8)$$

and

$$F1 = \frac{2*Recall*Precision}{Recall+Precision}. \quad (9)$$

Table 4. *Confusion matrix structure*

	Predicted Negative	Predicted Positive
Actual Positive	False Negative (FN)	True Positive (TP)
Actual Negative	True Negative (TN)	False Positive (FP)

Table 5. *Confusion matrix for DDoS attack detection*

a. Z-score with mean forward packets

	Predicted Benign	Predicted DDoS
Actual DDoS	14	7
Actual Benign	63	9

b. Z-score with forward speed

	Predicted Benign	Predicted DDoS
Actual DDoS	13	8
Actual Benign	63	9

c. Z-score with backward speed

	Predicted Benign	Predicted DDoS
Actual DDoS	13	8
Actual Benign	63	9

d. Modified Z-score with mean forward packets

	Predicted Benign	Predicted DDoS
Actual DDoS	1	20
Actual Benign	72	0

e. Modified Z-score with forward speed

	Predicted Benign	Predicted DDoS
Actual DDoS	1	20
Actual Benign	70	2

f. Modified Z-score with backward speed

	Predicted Benign	Predicted DDoS
Actual DDoS	1	20
Actual Benign	70	2

Table 6. *Confusion matrix for Port Scan detection*

a. Z-score with connection lifetime

	Predicted Benign	Predicted Port Scan
Actual Port Scan	14	13
Actual Benign	109	14

b. Z-score with unique destination ports count

	Predicted Benign	Predicted Port Scan
Actual Port Scan	16	11
Actual Benign	103	20

c. Z-score with destination ports entropy

	Predicted Benign	Predicted Port Scan
Actual Port Scan	17	10
Actual Benign	98	25

d. Modified Z-score with connection lifetime

	Predicted Benign	Predicted Port Scan
Actual Port Scan	16	11
Actual Benign	118	5

e. Modified Z-score with unique destination ports count

	Predicted Benign	Predicted Port Scan
Actual Port Scan	16	11
Actual Benign	119	4

f. Modified Z-score with destination ports entropy

	Predicted Benign	Predicted Port Scan
Actual Port Scan	16	11
Actual Benign	123	0

Based on the confusion matrices presented in

Table 5 and Table 6, I calculated the metrics summarized in Tables 7 and 8. The results lead to the following observations:

- low accuracy in the evaluation corresponds to a higher number of false positives,
- low recall indicates a higher number of false negatives.

When applying the sliding window algorithm with the standard Z-score, both DDoS and Port Scan detections yielded low accuracies and recalls (generally below 50%) across all selected features. In contrast, the global modified Z-score achieved strong performance for DDoS detection, with accuracy, recall, and F1-score typically exceeding 91%. However, its effectiveness for Port Scan detection remains less consistent.

The main hypothesis explaining the strong performance of the sliding window Z-score algorithm is the non-stationary nature of the time series. To confirm this hypothesis, I computed the p-values of the ADF and KPSS tests, as shown in Table 9, and examined how the mean and standard deviation of the time series evolve over time (see Fig. 14).

To better understand the performance of the modified Z-score in detecting port scanning activities, Fig. 9, Fig. 11, and Fig. 13 are particularly insightful. They show that the detected threats exhibit high variance, while those with low variance often remain undetected. Comparing the time intervals of high-dispersion threats with the original dataset and

considering the number of flows labeled as Port Scan, we can see that these intervals correspond to fast Port Scan attacks. This leads to a conclusion that the modified Z-score method effectively detects fast port scans but fails to identify slow ones.

Table 7. *DDoS detection metrics (accuracy, precision, recall, and F1)*

			Accuracy	Precision	Recall	F1	Support
Mean forward packets by source IP	Z-score	BENIGN	0.75	0.82	0.88	0.85	72
		DDoS		0.44	0.33	0.38	21
	Modified Z-score	BENIGN	0.99	0.99	1	0.99	72
		DDoS		1	0.95	0.98	21
Forward speed	Z-score	BENIGN	0.76	0.83	0.88	0.85	72
		DDoS		0.47	0.38	0.42	21
	Modified Z-score	BENIGN	0.97	0.99	0.97	0.98	72
		DDoS		0.91	0.95	0.93	21
Backward speed	Z-score	BENIGN	0.76	0.83	0.88	0.85	72
		DDoS		0.47	0.38	0.42	21
	Modified Z-score	BENIGN	0.97	0.99	0.97	0.98	72
		DDoS		0.91	0.95	0.93	21

The following paragraph examines how local non-stationarity in feature time series affects the detection of port scanning activities. To achieve this, we calculate the p-values of the ADF and KPSS statistics. The p-value can be expressed as follows: $p_{value} = P(x_{obs} \vee H_0)$, where H_0 is the null hypothesis, x_{obs} is the calculated statistic (ADF or KPSS). The general form of the ADF test regression is:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta_{t-i} + \epsilon_t \, . \tag{10}$$

Null hypothesis (H_0): $\gamma = 0 \rightarrow$ unit root is present (non-stationary). Alternative hypothesis (H_1): $\gamma < 0$ →no unit root (stationary). Then we can compute $ADF_{stat} = \frac{\hat{\gamma}}{SE(\gamma)}$. As $p_{value}$is more than 0.05, i fail to reject the null hypothesis. In the next step, i estimate the KPSS Test statistic formula as:

$$KPSS = \frac{1}{T^2} \frac{\sum_{t=1}^{T} S_t^2}{\hat{\sigma}^2} \, , \tag{11}$$

where:

T: The number of observations;

$S_t = \sum_{i=1}^{t} \hat{e}_i$: The **cumulative residuals** from the OLS regression of the time series on a constant (or constant + trend);

$\hat{\sigma^2}$*:* **Long-run variance** of the residuals, estimated using a **Newey-West estimator** (or similar), which accounts for autocorrelation.

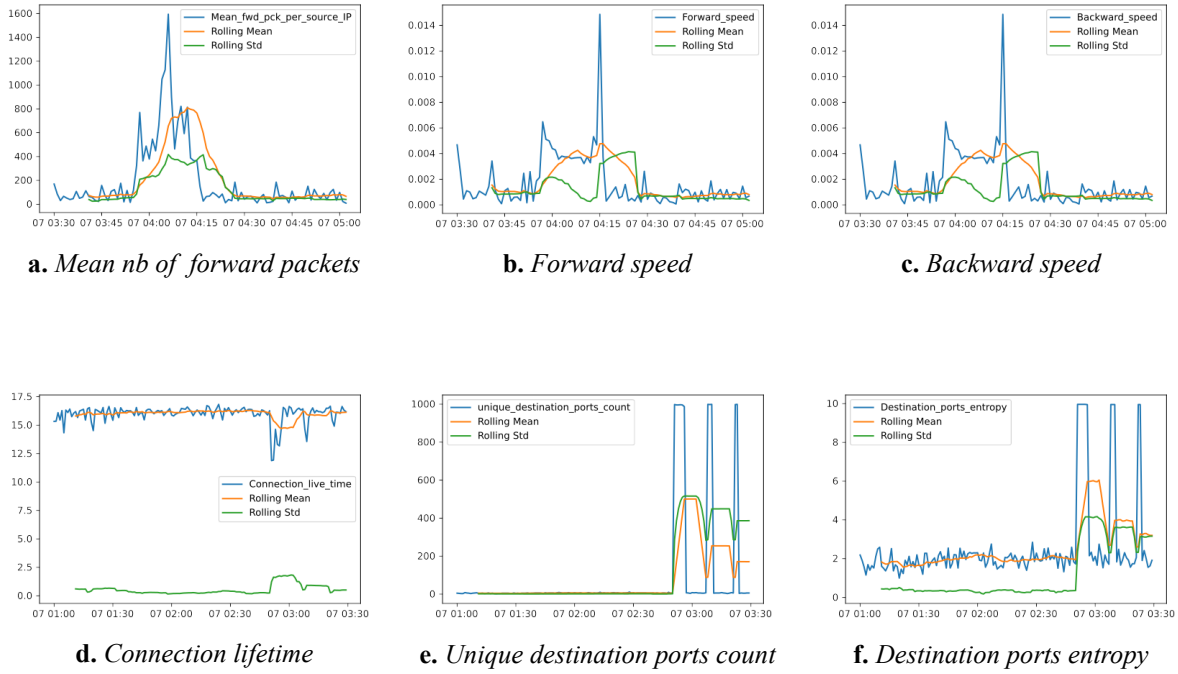
Null Hypothesis (H_0): The series is **stationary** (level or trend stationary). **Alternative Hypothesis (H_1)**: The series has a **unit root** (non-stationary). I **reject H_0** (stationary) if the $p_{value} < 0.05$ (KPSS statistic is **greater than the critical value)**.

Table 8. *Port Scan detection metrics (accuracy, precision, recall, and F1)*

			Accuracy	Precision	Recall	F1	Support
Connection lifetime	Z-score	BENIGN	0.813	0.89	0.89	0.89	123
		Port Scan		0.48	0.48	0.48	27
	Modified Z-score	BENIGN	0.860	0.88	0.96	0.92	123
		Port Scan		0.69	0.41	0.51	27
Unique destination count by source IP	Z-score	BENIGN	0.76	0.87	0.84	0.85	123
		Port Scan		0.35	0.41	0.38	27
	Modified Z-score	BENIGN	0.867	0.88	0.97	0.92	123
		Port Scan		0.73	0.41	0.52	27
Destination port entropy	Z-score	BENIGN	0.72	0.85	0.80	0.82	123
		Port Scan		0.29	0.37	0.32	27
	Modified Z-score	BENIGN	0.893	0.88	1.00	0.94	123
		Port Scan		1.00	0.41	0.58	27

Table 9. *ADF and KPSS statistics and p-values of the time series*

	ADF		KPSS	
	ADF Statistic	**p-value**	**Kpss Statistic**	**p-value**
Mean forward packets	-2.68053	0.0774668	0.279649	0.1
Forward speed	-3.6409978	0.0050239	0.3529392	0.0974400
Backward speed	-3.6409978	0.0050239	0.3529392	0.0974400
Connection lifetime	-4.8111134	5.1785264e-05	0.1351005	0.1
Unique destination ports count	-1.9973058	0.2877557	0.5839902	0.0240918
Destination ports entropy	-1.9618836	0.3035317	0.6503301	0.0180609



a. *Mean nb of forward packets* **b.** *Forward speed* **c.** *Backward speed*

d. *Connection lifetime* **e.** *Unique destination ports count* **f.** *Destination ports entropy*

Fig. 14. *Rolling statistical characteristics for the feature time series*

From the p-value results of the KPSS test applied to the time series used for port scan detection Table 9, we can conclude that the **'Connection lifetime'** time series is stationary, while the **'Unique destination ports count'** and **'Destination ports entropy'** series are non-stationary. The non-stationarity of the latter series results from traffic spikes caused by rapid port scanning activities, as illustrated in Fig.14 (d, e, f). Consequently, a high KPSS statistic in the local stationarity analysis can serve as a useful indicator of periods characterized by fast port scanning behavior.

Conclusion

This study evaluates the effectiveness of statistical methods – specifically the Z-score and the modified Z-score – for processing time series data in the detection of Port Scan and DDoS attacks. The findings indicate that the modified Z-score is better suited for identifying anomalies in time series with asymmetric distributions. Experimental results demonstrate that DDoS attacks can be readily detected using statistical approaches. However, the effectiveness of these methods in identifying port scans varies depending on the scanning type. The analysis shows strong detection performance for fast scans but complete ineffectiveness for slow ones. Further investigation reveals that this difference arises from local disruptions in the stationarity of time series during fast scans, as evidenced by a high KPSS coefficient. Consequently, two main conclusions are drawn: slow scans exhibit statistical characteristics similar to normal traffic; the analysis of local stationarity in time series can serve as a foundation for detecting fast port scans.

References

1. Scaranti, G. F., Carvalho, L. F., Barbon, S., Lloret, J. and Proença, M. L. (2022), "Unsupervised online anomaly detection in software defined network environments", *Expert Systems with Applications*, vol. 191, pp. 4–6.

2. Birkinshaw, C., Rouka, E. and Vassilakis, V. G. (2019), "Implementing an intrusion detection and prevention system using software-defined networking: defending against port-scanning and denial-of-service attacks", *Journal of Network and Computer Applications*, vol. 136, pp. 71–85.
3. Abrantes, R., Mestre, P. and Cunha, A. (2022), "Exploring dataset manipulation via machine learning for botnet traffic", *Procedia Computer Science*, vol. 196, pp. 133–141.
4. Ono, D., Guillen, L., Izumi, S., Abe, T. and Suganuma, T. (2021), "A proposal of port scan detection method based on Packet-In messages in OpenFlow networks and its evaluation", *International Journal of Network Management*, vol. 31, pp. 5–8.
5. Hartpence, B. and Kwasinski, A. (2020), "Combating TCP port scan attacks using sequential neural networks", *in 2020 International Conference on Computing, Networking and Communications* (ICNC).
6. Al-Haija, Q. A., Saleh, E. and Alnabhan, M. (2021), "Detecting port scan attacks using logistic regression", *in 2021 4th International Symposium on Advanced Electrical and Communication Technologies* (ISAECT).
7. Almseidin, M., Al-Kasassbeh, M. and Kovacs, S. (2019), "Detecting slow port scan using fuzzy rule interpolation", *in 2019 2nd International Conference on New Trends in Computing Sciences* (ICTCS).
8. Nisa, M. U. and Kifayat, K. (2020), "Detection of slow port scanning attacks", *in 2020 International Conference on Cyber Warfare and Security* (ICCWS).
9. Sagatov, E. S., Mayhoub, S., Sukhov, A. M., Esposito, F. and Calyam, P. (2021), "Proactive detection for countermeasures on port scanning based attacks", *in 2021 17th International Conference on Network and Service Management* (CNSM)
10. Baah, E. K., Yirenkyi, D., Oppong, S. O., Opoku-Mensah, E., Partey, B. T., Sackey, A. K., Kornyo, O. and Obu, E. (2022), "Enhancing port scans attack detection using principal component analysis and machine learning algorithms", *in Frontiers in Cyber Security*, Singapore.
11. Ring, M., Landes, D. and Hotho, A. (2018), "Detection of slow port scans in flow-based network traffic", *PLOS ONE*, vol. 13, pp. 1–18.

Information about the author:
A. M. A. E. Djeguede – Postgraduate student in the Department of Mathematical Modeling and Artificial Intelligence, Peoples' Friendship University of Russia (RUDN University) (6, Miklukho-Maklaya st., Moscow, Russia, 117198), https://orcid.org/0000-0002-8476-8994.

Информация об авторе:
А. М. А. Э. Джегюеде – аспирант кафедры математического моделирования и искусственного интеллекта Российского университета дружбы наров (117198, Россия, Москва, ул. Миклухо-Маклая, д. 6), https://orcid.org/0000-0002-8476-8994.