Multifaceted DDoS Attack Prediction by Multivariate Time Series and Ordinal Patterns

Ligia F. Borges*, Anderson B. de Neira†, Lucas Albano*, Michele Nogueira*†
*Department of Computer Science - Federal University of Minas Gerais, Brazil
†Department of Informatics - Federal University of Paraná, Brazil
Emails: {ligia.borges, lucasalbano, michele}@dcc.ufmg.br, andersonneira@ufpr.br

***Abstract*—Distributed Denial of Service (DDoS) attacks are recurrent threats, reaching unprecedented malicious network traffic volume and speed against targets. Predicting attacks is paramount to reduce costs in mitigating or remediating them. But, it is a challenging task due to attack multifaceted properties (*e.g.*, different attack vectors and network traffic). The multidimensional nature of these attacks requires equally multifaceted defenses. Existing solutions to DDoS attack prediction employ homogeneous data sources for model training, limiting the perspective in the face of data variability. The proposal benefits from multivariate time series correlation and noise tolerance from ordinal patterns transformation, a method suitable for IoT environments. The method predicts a DDoS attack up to 35 minutes before it effectively begins, surpassing the literature relying on approaches that require labeled data and solutions based on complex neural networks.**

***Index Terms*—Network Security, DDoS Attack Prediction, One Class SVM, IoT**

I. Introduction

Distributed Denial of Service (DDoS) attacks are one of the most harmful and recurrent threats [1]. A recent investigation revealed that 7,858,705 DDoS attacks occurred only in the first half (H1) of 2023, *i.e.*, an increase of 16% compared to 2022 [2]. Current DDoS attacks have a sudden, fast, and massive impact on targets, making detection mechanisms insufficient. As highlighted by Corero [3], if an automated mitigation solution takes over 10 seconds to counter a DDoS attack, the potential financial loss can reach as high as $36 million. Hence, it is paramount to investigate how to anticipate the steps of attacks. Predicting DDoS attacks considerably reduces the cost of mitigating and remediating an ongoing attack. However, identifying its signals is challenging due to noisy data and the multifaceted properties of DDoS attacks.

Multidimensional network traffic carries heterogeneous information from different network layers of the protocol stack. It offers multiple perspectives and enriches the analysis of the system. Aceto *et al.* have demonstrated that multifaceted approaches harness traffic data according to different inputs and enhance traffic classification [4]. However, multidimensional data undergoes noises, compromising its analysis if observed data follows no preprocessing [5]. The correlation of multifaceted data holds promise in DDoS attack prediction because combining information from different layers yields a comprehensive and reliable view, as it avoids the impact of erroneous or noisy data within a single information source.

The literature highlights the potential of multifaceted data and sources [6], [7]. In [6], the authors showed that data from different modalities enables valuable insights from pre-trained models and improves attack detection. In [7], the authors explored multimodal traffic data to improve detection models. Borges *et al.* [8] introduced a non-parametric approach based on time series for identifying botnets investigating temporal dynamics through a multiscale ordinal pattern transformation. These solutions target only attack detection. Few works focus on DDoS attack prediction as [9]–[11]. However, they exploit only labeled no-multifaceted data.

This work presents a method for predicting DDoS attacks founded on multifaceted data (*i.e.*, cross-layer feature and traffic) addressed as multivariate time series through ordinal patterns extracted from Bandt-Pompe symbolization [12]. The method transforms multivariate time series into ordinal patterns and transition graphs to engineer representative features. The engineered features characterize multivariate time series by data distribution. Therefore, correlated data (*i.e.*, extracted directly from different layers of the protocol stack) serves as input to predict DDoS attacks. The method automates the prediction using the One-Class Support Vector Machine (SVM) that does not require labeled data to perform the prediction [13], works well in unbalanced scenarios, and has already been tested in IoT environments [14].

The performance evaluation employs four datasets with different types of DDoS attacks and multifaceted network traffic. On dataset 1, the method predicts DDoS attacks up to 34 minutes with an accuracy of 91.52%. The attack prediction occurs 4 minutes and 23 seconds on dataset 2. On dataset 3, the method predicts an attack in 13 minutes and 22 seconds. Finally, on dataset 4, the method identifies an attack 35 minutes after bots infection in an IoT network with an accuracy of 72%. Results show improvements in DDoS attack prediction time compared to the literature [10], [11], [15].

This paper proceeds as follows. Section II presents the related works. Section III details the DDoS attack prediction technique. Section IV presents the results. Finally, Section V concludes the paper.

This work was supported by National Council for Scientific and Technological Development (CNPq/Brazil), grants #309129/2017-6 and #432204/2018-0, by São Paulo Research Foundation (FAPESP), grants #2022/06802-0 and #2022/06840-0, and CAPES, grant #88887.509309/2020-00.

II. Related Works

Attackers often look for ways to hide their activities. Despite a rich literature focusing on detection, it is vital to advance the state-of-the-art and investigate techniques to highlight signals of attack preparation [11]. The literature has investigated solutions to predict behavioral stages of attacks on the network. For example, in [9], the authors proposed a system based on deep learning to identify signals of DDoS attack orchestration. The system extracts features related to skewness and kurtosis from the collected network layer traffic. The system predicts attacks but requires a high consumption of time and resources. In [10], the authors presented a system based on long short-term memory autoencoder to identify signals of attack preparation through features related to skewness, kurtosis, and coefficient of variation, also extracted from the network layer traffic.

Related to data correlation, the majority of the existing works addresses attack or anomaly detection, not attack prediction. For instance, [16] focused on DoS attack detection based on multivariate correlation analysis (MCA) from network traffic [16]. Peng et al. utilized correlation information from training data and the CKNN algorithm for classification purposes [17]. Riyanat et al. presented the ACSAnIA model, which utilizes post-correlation methods, event correlation, and alert logs for network attack detection [18]. Chagas et al. applies ordinal patterns and information theory descriptors in the analysis of temporal data for botnet detection in IoT [19].

In general, only few works perform DDoS attack prediction, and they use approaches that require labeled data [11], [15] or propose solutions based on complex neural networks [9], [10], that have long training time and require high computational power. These solutions employ a reduced set of features, adhering to a single-modal approach, which provides a less comprehensive understanding of data variability and reduces the prediction time. This work advances the literature presenting a DDoS attack prediction method based on the correlation of multifaceted data through time series and ordinal patterns [12], a noise-resistant method to identify outliers [12]. The solution does not require labeled data, preventing the model from being tied to specific attacks.

III. DDoS Attack Prediction Method by Multifaceted Data

This section describes the proposed method to predict DDoS attacks using multifaceted network traffic. The method applies data transformation into ordinal patterns to engineer features that highlight signs of an attack preparation. The previous study [20] validated the ordinal patterns to correlate temporal series and predict attacks but did not consider the multifaceted characteristics of network traffic. The method follows an unsupervised approach, and it comprises four steps (Fig.1). The next subsections detail each step: network data capture; feature engineering; model training; and DDoS attack prediction.

A. Network Traffic Capture

In this step, a monitor integrated into a firewall captures network packet headers of the network traffic. The monitor forwards a copy of packet headers to a Packet Capture (PCAP) file server that stores all packet headers. This work refers to the captured packet headers as simply data. The collected data is saved into a PCAP file format to enable the extraction of network traffic attributes. The saved files contain values for the packet header fields from each protocol stack layer. The PCAP file server must have sufficient computational resources to store network traffic temporarily, since a copy of thousands of packet headers can result in several megabytes.

B. Feature Engineering

In this step, the method performs the engineering of relevant features for DDoS attack prediction. The feature engineering process starts extracting multifaceted traffic attributes directly from the packet headers in the stored PCAP files (Step 1). The method follows the assumption that the attack preparation impacts network attributes, e.g., the number of devices exchanging packets or the number of exchanged packets that change during the attack preparation phase given pre-attack tests [21]. In [22], the authors present 40 representative network attributes for detecting command and control (C&C) communication. C&C communication is the channel to control and coordinate botnets, including those that launch DDoS attacks. As C&C communication occurs before DDoS attacks, they serve as basis to predict them.

The method performs attribute extraction per unit of time, resulting in Multivariate Time Series (MTS). Each selected attribute is a dimension in MTS. This method employs as attributes: the number of IP addresses of the source and destination (network layer), the number of ICMP protocol packets (network layer), the number of TCP protocol packets (transport layer), and the number of frames (link layer).

Next, the method processes MTS using the Bandt-Pompe transformation methodology [12]. Bandt-Pompe transformation has been proven to be less sensitive to noise in data; this generates fewer false positives. The transformation has MTS as input and ordinal patterns, representative from MTS, as output. The transformation process works per subset of MTS, following two steps: (i) the division of MTS into subsets of size D and (ii) the generation of ordinal patterns.

For each subset of size $D \in \mathbb{N}$ from MTS, the method sorts its values in ascending order. Thus, given an MTS $x = (x_1, ..., x_n)$ of total length n and a delay $\tau \in \mathbb{N}$ (difference between points in the subset) at each time $t = \{1, ..., n-(D-1)\tau\}$, a sliding window $\mathbf{w}_t$ is defined as:

$$\mathbf{w}_t = (x_t, x_{t+\tau}, \ldots, x_{t+(D-2)\tau}, x_{t+(D-1)\tau}), \quad (1)$$

where the ordinal pattern for each sliding window consists in the permutation required on the elements of w_t to be ascending sorted. For each time t, the ordinal relation consists of the permutation $\pi = \{r_1, r_2, \ldots, r_D\}$ of $(1, 2, \ldots, D)$ as:

$$x_{t+r_1-1} \leq x_{t+r_2-1} \leq \cdots \leq x_{t+r_{D-1}-1} \leq x_{t+r_D-1}. \quad (2)$$

The proposed method converts the time series into a set of ordinal patterns $\Pi = \{\pi_1, \ldots, \pi_m\}$, where $m = n-1(D-1)\tau$, and each π_i represents a combination from the set of $D!$

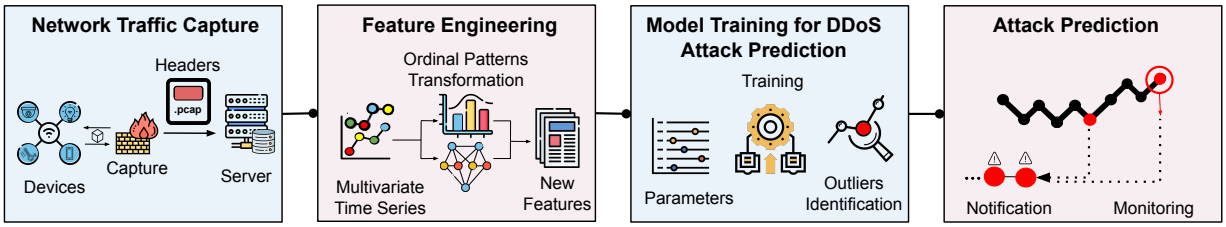

Fig. 1. Overview of the proposed method

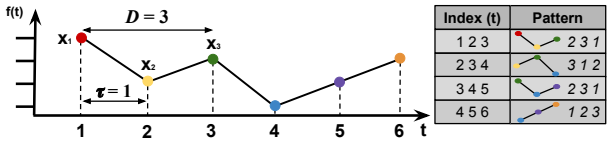

Fig. 2. Construction of the ordinal pattern of $D = 3$, delay τ=1 and $n = 6$

possible permutations. Eq. 3 explains the $D!$ possible permutations. Because the number of elements in the subset (m) is always the size of D and as the permutation will occur over all elements of the subset, the value of p in Eq. 3 is also D. Thus, $D!$ represents the total of possible permutations.

$$Permutations(m, p) = \frac{m!}{(m-p)!} \tag{3}$$

The choice for D depends on the length n of MTS and must satisfy $n \gg D$ to achieve reliable statistics [8]. Fig. 2 illustrates a time series transformation process. The first sliding window with $D = 3$ and $\tau = 1$ is given by $\mathbf{w}_1 = (x_1, x_2, x_3)$. The necessary permutation lies in moving the first element to the end, the second to the beginning, and the last to the middle (*i.e.*, ascending order), resulting in the $\mathbf{w'}_t$=(x_2;x_3;x_1) that matches the pattern $\pi = \{2, 3, 1\}$ in relation to t. This process is applied to each dimension of MTS, resulting in a set of ordinal patterns for each dimension.

Once MTS has been transformed in ordinal patterns, the method represents them in a new domain (transition graphs and probability distribution). This process is important because the resulting distribution becomes less noise-sensitive [23]. As known, noise disrupts data distribution in various ways, often resulting in imbalances and yielding unexpected outcomes. This leads consequently to models failing to classify an attack.

The method analyzes the sets of ordinal patterns obtained from MTS by frequency histograms and their transitions. This process entails individually analyzing the occurrence frequency of each pattern, facilitating the identification of dominant patterns, less common occurrences, and anomalies. Transition graphs and histograms, which visually depict the transition probabilities between individual patterns, offer a means of visualizing this distribution within MTS with no requirement for long time series [23].

The **probability distribution** of ordinal patterns results from the frequency histogram. Hence, it is necessary to identify the probability distribution of permutations $p_\pi = \{p(\pi_t) : \forall t \in 1, \ldots, D!\}$ from a set of ordinal patterns Π obtained from MTS transformation:

$$p(\pi_t) = \frac{|s_{\pi_t}|}{n - (D-1)\tau} \tag{4}$$

where $\pi_t \in \Pi$ represents each possible permutation where $t \in \{1, \ldots, D!\}$, be $|s_{\pi_t}| \in \{0, \ldots, m\}$ and $|s_{\pi_t}| \in \{0, \ldots, m\}$ is the number of observed patterns of type π_t, satisfying $p(\pi_t) \geq 0$ and $\sum_{\pi_t} p(\pi_t) = 1$. The probabilities remain invariant under monotonic transformations, then the presence of multiplicative noise does not impact the outcomes of the generated patterns [24].

Transition graphs: The method uses the analysis of the ordinal patterns set transformed into transition graphs to characterize MTS according to their behavior to support the prediction of DDoS attacks. Depicting a set Π of ordinal patterns as a directed graph $G = (V, E)$ serves to highlight the transitions between consecutive ordinal patterns across time t. Each vertex means a unique pattern, and the edge weights denote the relative frequency of these transitions [24].

$$V = \{v_{\tilde{\pi}_t^D}\}, \text{ and } E = \{(v_{\tilde{\pi}_t^D}, v_{\tilde{\pi}_{t+1}^D}) : v_{\tilde{\pi}_t^D}, v_{\tilde{\pi}_{t+1}^D} \in V\}. \tag{5}$$

where each edge receives a weight $W = \{w_{v_{\tilde{\pi}_i^D}, v_{\tilde{\pi}_j^D}} : v_{\widetilde{pi}_i^D}, v_{\tilde{\pi}_j^D} \in V\}$ which represents the transition probability between each pattern according to Eq. 6.

$$w_{v_{\tilde{\pi}_i^D}, v_{\tilde{\pi}_j^D}} = \frac{|\Pi_{\tilde{\pi}_i^D, \tilde{\pi}_j^D}|}{T - (D-1)\tau - 1} \tag{6}$$

where the term $|\Pi_{\tilde{\pi}_i^D, \tilde{\pi}_j^D}|$ represents the number of transitions between patterns (*i.e.*, from $\tilde{\pi}_i^D$ to $\tilde{\pi}_j^D$) respecting the restriction $\sum_{v_{\tilde{\pi}_i^D}, v_{\tilde{\pi}_j^D}} w_{v_{\tilde{\pi}_i^D}, v_{\tilde{\pi}_j^D}} = 1$ and the denominator is the number of transitions in the series of length $T-(D-1)\tau$. The calculation of this transition graph incorporates the key benefits of the permutation pattern approach: it boasts conceptual simplicity, offers lower computational overhead, and exhibits resilience in the face of noisy data [23]. The method engineers features from the probability distribution and transition graph obtained from the transformation into ordinal patterns. Hence, it employs information theory quantifiers to characterize the dynamic behavior of MTS [8], [25]. The representative engineered features are: (i) normalized permutation entropy; (ii) statistical complexity; (iii) Fisher information measure; (iv) the number of vertices in the transition graph; (v) normalized permutation entropy of the edge weighted distribution; (vi) statistical complexity of the edge weight distribution; (vii)

Fisher information measure of weighted edge distribution, and $(viii)$ the probability of vertex self-transition.

The **permutation entropy** measures the complexity and unpredictability of a sequence of ordinal patterns. A higher permutation entropy suggests more diverse and unpredictable patterns in the time sequence, while lower values indicate a deterministic original time series [12].

$$H = -\sum_{t=1}^{D!} p(\pi_t) \log p(\pi_t) \quad (7) \qquad H_{norm} = \frac{H}{\log D!} \quad (8)$$

where H is the classical Shannon Entropy and H_{norm} is the normalized entropy, $0 \leq H \leq \log D!$ and $0 \leq H_{norm} \leq 1$.

Statistical complexity provides insights into the complexity and unpredictability of the structure of ordinal patterns in the original series sequence. This measure helps to recognize patterns and detect anomalies in the captured time series data. Here, the statistical complexity of ordinal patterns is calculated by considering the permutation entropy as a measure of imbalance (*i.e.*, that quantifies the difference between an expected and measured distribution).

$$C_{JS}[p_\pi] = Q_{JS}[p_\pi, p_u] H_{norm}, \quad (9)$$

where $p_\pi = \{p(\pi)\}$ is the probability distribution of ordinal patterns, p_u is the uniform distribution of $\{1, 2, \ldots, D!\}$, and H_{norm} is the normalized Shannon entropy. The measure of imbalance $Q_{JS}[p_\pi, p_u]$ follows Eq. 10 [26]:

$$Q_{JS}[p_\pi, p_u] = Q_0 \left\{ S\left[\frac{p_\pi + p_u}{2}\right] - \frac{S[p_\pi] + S[p_u]}{2} \right\}, \quad (10)$$

$$where, Q_0 = -2\left\{ \left(\frac{D!+1}{D!}\right) \ln(D!+1) - 2\ln(2D!) + \ln(D!) \right\}^{-1}, \quad (11)$$

S is the Shannon entropy, Q_0 is the normalization constant according to $0 \leq Q_{JS} \leq 1$.

The **Fisher information** characterizes the statistical complexity of the patterns found in time series and provides insights into the complexity and imbalance of data. This statistical quantifier presents a locality property as it considers differences between consecutive probabilities in the distribution. The Fisher information rises with greater probability distribution specificity (*i.e.*, when probabilities are close).

$$F[p_\pi] = F_0 \sum_{t=1}^{D!-1} (\sqrt{p_{t+1}} - \sqrt{p_t})^2, \quad (12)$$

where F_0 is a normalization constant defined as:

$$F_0 = \begin{cases} 1 \text{ if } p_{i^*} = 1 \text{ to } i^* = 1 \text{ or } i^* = N \text{ and } p_i = 0, \forall i \neq i^*, \\ 1/2 \text{ otherwise.} \end{cases} \quad (13)$$

The **probability of self-transition** in the graph (*i.e.*, consecutive ordinal patterns) is an important characteristic in the dynamics of time series, as it is related to their temporal correlation. The self-transition probability p_{st} follows Eq. 14.

$$p_{st} = p(\pi_i, \pi_i) = \sum_{i \in \{1, \ldots, D!\}} w(v_{\pi_i}, v_{\pi_i}). \quad (14)$$

The sum of the main diagonal of the adjacency matrix of G_π equals the probability of self-transition, calculated as:

$$p_{st} = \sum_{\pi_i \in \Pi} a_{\pi_i, \pi_i} \quad (15)$$

C. Model Training for DDoS Attack Prediction

In Step 3, One-Class SVM machine learning algorithm detects outliers using as basis the engineered features (Step 2). This algorithm follows a semi-supervised learning that requires only one class (in this case, normal network traffic) of training data before classifying outliers [13]. The algorithm finds a decision boundary separating normal data from outliers. A hyperplane that maximizes the margin between data defines the decision boundary. This limit establishes new data points (normal/abnormal) depending on which side they are on. One-Class SVM deals with high-dimensional data depending on parameters as a limit on the training error rate and a limit on the proportion of support vectors. By adjusting parameters, it is possible to control the number of data points considered outliers during training and also determine how data is separated within space (*e.g.*, linear, poly, and sigmoid).

D. Attack Prediction

In Step 4, the method utilizes the One-Class SVM to identify unusual observations (outliers) in a non-labeled data set. Outliers represent changes in network traffic that may not be noticeable. Step 2 has highlighted these changes. The ordinal transformation and feature engineering enable the correlation of diverse data patterns in MTS to identify benign or malicious traffic data.

IV. Evaluation

The method evaluation follows four experiments with datasets that have the information when DDoS attacks start and when devices are infected. For evaluation, only the network traffic before the effective beginning of a DDoS attack (*i.e.*, when bots start to overload targets) has taken place. This approach ensures that the model is not dependent on the attack type. The evaluation verifies the method efficiency in identifying evidence of attack preparation. The datasets show attackers' actions, such as bot infection, attack tests, and multifaceted network traffic. In all experiments, One-Class SVM follows standard parameters values from the Scikit-learn.All results are available online (https://github.com/ligiafb/ICC24-W18).

A. Experiments Definition

Experiment 1 (Exp 1) follows traffic from a local network available in CTU-13 capture 51 [27]. The capture has 8803 seconds, 41 GB, 46 million packets, Internet Control Message Protocol (ICMP) and User Datagram Protocol (UDP) flood attacks, and ten bots. Researchers has launched the attacks at 5632 second and combined the attacks with real data. One Class SVM had a training step that did not use labeled data. This work used only a third of the dataset to train it. Thus, the training went from the beginning of the dataset until the second 2934. As the goal was to predict attacks, this experiment only analyzes network traffic before the effective beginning of the attack. Thus, the test went up to the second 5632.

Experiment 2 (Exp 2) employs local network traffic collected in capture 52 of [27]. The capture is 972 seconds

long, 555 MB, 6 million packets, an ICMP flood attack, and three bots. The researchers have conducted the attack on the second 778 of the capture and combined it with real data. This experiment also utilized one-third of the dataset for training the One-Class SVM (up to 324 seconds), with the testing period concluding at the start of the attack (up to 778 seconds). Experiment 3 (Exp 3) evaluates network traffic from the CIC-DDoS2019 [28] dataset, with 19 attacks launched by researchers in two days. The dataset has 27 GB of attack and real data, 61 million packets. Bots connected to the victim over the Internet. The first attack started on second 1484. The experiment employs one-third of the dataset (up to 674 seconds) and evaluates it until the attack began.

Experiment 4 (Exp 4) employs the IoT-23 dataset with 23 DDoS attack scenarios in IoT environments [29]. Scenario 17 contains more than one infected and active bot. The scenario has 8.3 GB of data and 109 million packets sent in 24 hours. The researchers started the capture at 06:43:20, and the malware execution was at 11:43:43. Thus, the pre-infection traffic capture has legitimate traffic, and the post-infection traffic contains traces of the attack preparation. The documentation did not specify the time when the attacks started.

In Exps 1, 2, and 3, a one-second interval was used to group packets aiming to obtain more accurate predictions [11]. A one-minute interval is applied in Exp 4 since the dataset contains 24 hours of traffic. This study labeled each interval as normal or malicious. Data labeling only quantifies the results, since the One-Class SVM does not require labels to train or predict attacks. The normal interval comprises all network traffic, where bots send or receive no packets. The malicious interval contains network traffic where at least one packet has a bot as its origin or destination.

Attribute selection has considered the multifaceted characteristic of the data to offer a more robust view of data variability. Thus, all experiments consider the network traffic attributes: the number of source/destination IP addresses (network layer), the number of ICMP protocol packets (network layer), and the number of TCP protocol packets (transport layer). These attributes were selected because IP spoofing is common in DDoS attacks [1]. Experiments 1, 2 and 3 used the number of frames in the link layer, a significant attribute identified in [22]. In Exp 4, the total length of the IP header (network layer) is used instead of the number of packets because the dataset IoT-23 has different characteristics. The method engineers eight features (Subsec. III-B). It employs the fixed-size sliding window [11] to eliminate erroneous trends and evaluate the proposal (empirically, we selected 5% for Exps 1, 2, and 3 and 10% for Exps 4 to maximize the prediction time). This work uses weighted average precision and recall due to imbalanced data.

B. Results

Table I summarizes the results of all experiments and compares them to results of literature that predict DDoS attacks applying different methodologies (N/A = not appear) [10], [11], [15]. In Exp 1, three malicious intervals were correctly identified, where bots transfer data (true positive – TP). Thus, the proposed method issue two alerts indicating the DDoS attack. The first correct alert occurred 34 minutes and 55 seconds before the attack started. The method correctly identified 2469 normal intervals (true negative — TN), 15 normal intervals as outliers (false positives — FP), and 214 malicious intervals were incorrectly identified as normal intervals (false negatives – FN). The results indicate an accuracy of 91.52%, weighted precision of 85.6%, and weighted recall of 91.52%.

TABLE I
EXPERIMENTS RESULTS

Exps	Acc.	Prec.	Rec.	Prediction	Dataset
Exp 1	91.52%	85.6%	91.52%	34m55s	CTU-13 (51)
Exp 2	89.65%	90.58%	89.65%	4m23s	CTU-13 (52)
Exp 3	69.26%	78.72%	69.26%	13m22s	CIC-DDoS19
Exp 4	72.31%	78.70%	72.31%	35min	IoT-23
[10]	97.89%	97.4%	97.9%	29m51s	CTU-13 (51)
[15]	N/A	N/A	N/A	5m41s	CTU-13 (51)
[11]	98.87%	N/A	N/A	3m49s	CTU-13 (52)
[11]	99.60%	N/A	N/A	3m55s	CIC-DDoS19

In Exp 2, the method has correctly identified three malicious intervals (where bots send/receive packets). Thus, the method generates three alerts indicating a DDoS attack would happen. The prediction of the attack (alert) happened 4 minutes and 23 seconds before the attack. The method has wrongly identified 26 normal intervals as outliers, generating 26 FPs. Regarding FN, 21 malicious intervals were wrongly identified as normal intervals. The result indicates that 404 normal intervals are TN. Finally, the results show an accuracy of 89.65%, weighted mean precision of 90.58%, and weighted mean recall of 89.65%. In Exp 3, the method has correctly identified one malicious interval (TP). Thus, it would issue one alert indicating an upcoming DDoS attack. The attack prediction occurred in 13 minutes and 22 seconds. FP was 0, with 249 FN and 560 TN. Results indicate an accuracy of 69.26%, weighted precision of 78.70%, and weighted recall of 69.26%.

Exp 4 results refers to 80 minutes before and 50 minutes after the malware execution. The analysis has considered 50 minutes to reinforce that the proposed approach can quickly identify the signals of the attack preparation, even considering an imbalanced scenario. The analysis has correctly indicated 15 TP, 1 FP, 35 FN, and 79 TN with an accuracy of 72.31%, a weighted precision of 78.70%, and a weighted recall of 72%. The method has identified the attack preparation 35 minutes after the execution of malware on an IoT network.

C. Discussion

The proposed method has surpassed the prediction time of works that use labeled data [11], [15]. In Exp 1, the attack prediction occurred 34 minutes and 55 seconds before the attack effectively started (*i.e.*, 19 seconds after the start of device infections by bots, outperforming [10], [11], [15]). The accuracy exceeds that gotten in [11], but it is lower than [10], [15]. In contrast, [10] used a time and resource-intensive neural network, taking 42 and 28 minutes for configuration

and training on CTU-13 (51/52). The method presented in this paper has achieved the same results in just 0.0565 and 0.0028 seconds using One-Class SVM.

The dataset employed in Exp 2 is shorter (972s), making DDoS attack prediction challenging. The prediction occurred in 4 minutes and 23 seconds, overcoming [11]. In Exp 3, where Internet links the victim and attackers, the proposal has outperformed [11] by detecting the first attack in 13 minutes and 22 seconds. As the dataset (Exp 3) does not highlight the beginning of the infection, it is possible that the labeling process incorrectly indicated a malicious interval, lowering the accuracy. The dataset used in Exp 4 has introduced novel aspects (IoT scenario) to the experimentation. Although it was not possible to measure how long before attack onset prediction occurred, Exp 4 was an excellent example of how important attack prediction is. Results suggest that the prediction on IoT would occur 35 minutes after bot infection.

The proposal increases the DDoS prediction time compared to the literature. Given these attacks' rapid and voluminous nature, a mere 10 seconds of an ongoing attack can significantly impact targets, underscoring the importance of each additional minute in prediction time to avert potential losses. The prevalence of false negatives is the primary factor influencing results. This work hypothesizes that the traffic generated by bots is not always enough to impact network traffic. While, its impact is smaller than false positives, which may incorrectly point to a DDoS threat. However, this is a minor concern since the method focuses on attack prediction. False positives account for only 2%, 6%, 2%, and 1% of total samples in Exps 1, 2, 3, and 4, respectively. These results are minimal since the solution does not rely on labeled data for prediction, and the data is unbalanced. Future work will study solutions to improve accuracy, such as the use of different information sources and feature selection methods.

V. Conclusion

This work presented a DDoS attack prediction method that benefits from MTS and ordinal patterns. The proposal combines the transformation of time series into ordinal patterns and their symbolization in transition graphs to engineer relevant features of multifaceted network traffic to predict DDoS attacks. The proposed method uses initially attributes from different layers of the network protocol stack, characterizing a multifaceted approach. Then, an One-Class SVM ML model takes as basis the engineered features to predict DDoS attacks in an unsupervised way. The method predicts a DDoS attack up to 35 minutes before it effectively begins.

References

[1] N. Jyoti and S. Behal, "A meta-evaluation of machine learning techniques for detection of DDoS attacks," in *INDIACom*. India: IEEE, 2021, pp. 522–526.

[2] Netscout, "Findings from 1st half 2023[(Accessed on: October 2023)]," https://abrir.link/yFadA, Netscout, 2023.

[3] CORERO, "The need for always-on in real-timeddos security solutions[(Accessed on: October 2023)]," https://abrir.link/3L3Jo, CORERO NETWORK SECURITY, 2020.

[4] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE TNSM*, vol. 16, no. 2, pp. 445–458, 2019.

[5] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer science & business media, 2009.

[6] H. Elubeyd, D. Yiltas-Kaplan, and Ş. Bahtıyar, "A multi-modal deep transfer learning framework for attack detection in software-defined networks," *IEEE Access*, 2023.

[7] H. He, X. Sun, H. He, G. Zhao, L. He, and J. Ren, "A novel multimodal-sequential approach based on multi-view features for network intrusion detection," *IEEE Access*, vol. 7, pp. 183 207–183 221, 2019.

[8] J. B. Borges, J. P. Medeiros, L. P. Barbosa, H. S. Ramos, and A. A. Loureiro, "IoT botnet detection based on anomalies of multiscale time series dynamics," *IEEE TKDE*, 2022.

[9] G. L. F. M. E Silva, A. B. de Neira, and M. Nogueira, "A deep learning-based system for DDoS attack anticipation," in *LATINCOM*, 2022, p. 6.

[10] D. Brito, A. Neira, L. Borges, A. Araújo, and M. Nogueira, "Um sistema autônomo para a predição de ataques de DDoS em redes locais e internet," in *WGRS*. Brasil: SBC, 2023, pp. 29–42.

[11] A. B. de Neira, A. M. de Araujo, and M. Nogueira, "An intelligent system for DDoS attack prediction based on early warning signals," *IEEE TNSM*, vol. 20, no. 2, pp. 1–13, 2023.

[12] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *PL*, vol. 99, no. 17, p. 1541, 2002.

[13] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," *ACM SIGKDD*, pp. 8–15, 2013.

[14] V. H. Bezerra, V. G. T. da Costa, S. Barbon Junior, R. S. Miani, and B. B. Zarpelão, "Iotds: A one-class classification approach to detect botnets in internet of things devices," *Sensors*, no. 14, p. 3188, 2019.

[15] B. M. Rahal, A. Santos, and M. Nogueira, "A distributed architecture for DDoS prediction and bot detection," *IEEE Access*, vol. 8, p. 17, 2020.

[16] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE TPDS*, vol. 25, no. 2, pp. 447–456, 2013.

[17] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting ddos attacks against data center with correlation analysis," *JCC*, vol. 67, pp. 66–74, 2015.

[18] R. Shittu, A. Healing, R. Ghanea-Hercock, R. Bloomfield, and M. Rajarajan, "Intrusion alert prioritisation and attack detection using post-correlation analysis," *Computers & Security*, vol. 50, pp. 1–15, 2015.

[19] E. T. Chagas, J. B. Borges, and H. S. Ramos, "Uso de padrões ordinais na caracterização e análise de ataques de botnets em internet das coisas (IoT)," in *XXVIII WebMedia*. SBC, 2022, pp. 133–137.

[20] L. Albano, L. F. Borges, A. Neira, and M. Nogueira, "Predição de ataques ddos pela correlação de séries temporais via padrões ordinais," *SBSeg*, pp. 1–14, 2023.

[21] A. N. Jaber, M. F. Zolkipli, M. A. Majid, and S. Anwar, "Methods for preventing distributed denial of service attacks in cloud computing," *ASL*, vol. 23, no. 6, pp. 5282–5285, 2017.

[22] Y. Feng, H. Akiyama, L. Lu, and K. Sakurai, "Feature selection for machine learning-based early detection of distributed cyber attacks," in *DASC*. Greece: IEEE, 2018, pp. 173–180.

[23] M. Zanin and F. Olivares, "Ordinal patterns-based methodologies for distinguishing chaos from noise in discrete time series," *CP*, vol. 4, no. 1, p. 190, 2021.

[24] J. B. Borges, H. S. Ramos, R. A. Mini, O. A. Rosso, A. C. Frery, and A. A. Loureiro, "Learning and distinguishing time series dynamics via ordinal patterns transition graphs," *AMC*, vol. 362, p. 124554, 2019.

[25] H. V. Ribeiro, M. Jauregui, and E. K. Zunino Luciano, Lenzi, "Characterizing time series via complexity-entropy curves," *PR*, vol. 95, pp. 1–15, 2017.

[26] O. A. Rosso, H. Larrondo, M. T. Martin, A. Plastino, and M. A. Fuentes, "Distinguishing noise from chaos," *PLR*, vol. 99, no. 15, p. 1541, 2007.

[27] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *C&S*, vol. 45, pp. 100–123, 2014.

[28] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *ICCST*, 2019.

[29] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IoT-23: A labeled dataset with malicious and benign IoT network traffic," 2020.