An Intelligent System for DDoS Attack Prediction Based on Early Warning Signals

Anderson B. de Neira *Student Member, IEEE*, Alex M. de Araujo and Michele Nogueira, *Senior Member, IEEE*

***Abstract*—Among the different threats causing significant losses in cyberspace, the distributed denial of service (DDoS) attack is one of the most dangerous. The literature shows that the most reasonable manner to reduce the impacts of a DDoS attack is to prevent an attacker from launching it. Prevention is essential because attack sophistication allows them to reach massive traffic volumes, bypassing defenses. Defense mechanisms need time to detect and mitigate attacks. Hence, it is paramount to manage signals of the attack preparation before the attacker effectively launches it. This work presents COOPRED DDoS, a cooperative system for predicting DDoS attacks based on early warning signals extracted from the preparation of DDoS attacks. Its goal lies in increasing the time to prevent DDoS attacks. This work has followed four experiments utilizing two datasets widely employed in the literature. The results show that COOPRED DDoS identifies signals of attacks before the attacker effectively launches them. The system predicts one of the investigated attacks up to 3 minutes and 49 seconds in advance and the other attack up to 3 minutes and 55 seconds. The accuracy of the experiments varies from 99.60% to 99.87%.**

***Index Terms*—Security Management, DDoS Prediction, Network Traffic Analysis, Machine Learning.**

I. Introduction

Security management has never been more necessary than it is now because, with an overwhelming variety of applications and services available over the Internet, users access various forms of work, entertainment, and platforms. During the coronavirus pandemic, connectivity enabled online learning, conferences, medical appointments by video call, and teleworking. Driven by new habits acquired during the pandemic, in 2020, e-commerce has grown by 42% compared to 2019, moving US$ 813 billion [1]. However, society and network administrators must be careful to avoid losses. Distributed Denial of Service (DDoS), phishing, and malware-based attacks are examples described in the literature [2]. DDoS attacks are among the most dangerous existing cyber threats [3]. In 2020, experts identified 1,560,000 more DDoS attacks than in 2019 [4]. Platform-as-a-service cloud environments contribute to increasing the number of DDoS attacks. It is possible to rent a DDoS attack for US$ 100 per day [5].

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), grant #2018/23098-0, by the Coordination for the Improvement of Higher Education Personnel (CAPES/Brazil), grant #88887.509309/2020-00.

Anderson Bergamini de Neira and Alex Medeiros de Araujo are with the Department of Informatics, Federal University of Paraná, Curitiba, Brazil. E-mails: abneira, amaraujo@inf.ufpr.br. Michele Nogueira is with the Computer Science Department at both Federal University of Paraná and Federal University of Minas Gerais, Brazil. E-mail: michele@dcc.ufmg.br.

Due to the importance of network security, there is a vast literature on efforts to reduce the damage caused by DDoS attacks. Prevention, detection, and mitigation are defense mechanisms against DDoS attacks available to network administrators [6]. Preventive actions are installing a firewall, maintaining the infrastructure up to date, and disabling unused services. Several studies address attack detection. In [7], the authors combined two types of entropy to detect DDoS attacks. Utilizing Stacked Auto Encoder, a deep learning technique, the authors achieved 94% accuracy in detecting attacks. In [8], the authors proposed a hybrid approach based on Autoencoder and Multi-layer Perceptron (MLP) to detect DDoS attacks. The accuracy of the proposal surpassed 98%. In [9], the authors presented a solution to mitigate attacks. The solution conducts legitimate users to the service and isolates attackers. The results show that the proposal mitigates DDoS attacks, even with the quality of service degradation.

Network administrators have limited time to avoid the losses caused by DDoS attacks [10]. After the attacker launches the attack, network administrators are surprised by resource consumption growth, harming the access of real users. Two reasons influence the limited time to respond against DDoS attacks. Firstly, attacks have constantly been evolving, reaching massive traffic volumes quickly. In 2021, an attack that reached 21.8 million requests per second was one of the biggest DDoS attacks documented [11]. In 2022, Microsoft Corporation reported that it was the target of a DDoS attack that, in just one minute, reached the level of 3.47 terabits per second [12]. Second, defense mechanisms take time to detect and mitigate DDoS attacks. The attack must show service degradation signals to detect DDoS attacks to work correctly. Furthermore, even after detecting the attack, avoiding losses is not a trivial task [6]. Every second is critical in combating DDoS attacks due to their dangerousness, difficulty in alleviating the damage, and limited time to stop the attacks.

This work presents the COOPRED DDoS system, a cooperative system for predicting DDoS attacks. The system aims to increase the time to manage scenarios with DDoS attacks and prevent them. The COOPRED DDoS system follows the premise that network administrators must stop the attack before it effectively occurs [10], [13], being necessary to identify signals of the preparation of DDoS attacks before the attacker launches them. Distributed processing, early warning signal (EWS) engineering, cooperation, and machine learning are the underlying concepts of the system. The system's differential is the intelligence center that performs the predictions without prior knowledge of attack signatures or alerts generated by Intrusion Detection Systems (IDS). The intelligence center

makes cooperation possible by synchronizing the reception of signals collected by the distributed agents. Furthermore, the intelligence center automates the discovery of attack preparation signals. Therefore, machine learning in the intelligence center recognizes difficult patterns for experts to identify.

This work follows four experiments to evaluate the COOPRED DDoS system and the EWS engineering. Experiments 1 and 3 have utilized data provided by the Czech Technical University. The system predicted the attack up to 3 minutes and 49 seconds before the attack. The best accuracy obtained by the system in Experiments 1 and 3 was 99.69%. Experiments 2 and 4 have utilized data provided by the University of New Brunswick. In Experiments 2 and 4, the system performed the attack prediction up to 3 minutes and 55 seconds before the attack. The best accuracy obtained by the system in Experiments 2 and 4 was 99.76%. Therefore, in all experiments, the COOPRED DDoS system could notify network administrators about the possibility of an attack before the attacker effectively launches the attack. These results indicate that the system identifies signals of the preparation of DDoS attacks. In addition, the system provides more time for security management because the network administrators stop the attack before it causes losses.

The main contributions of this article are (i) a distributed and cooperative system to predict DDoS attacks, (ii) an early warning signals engineering to identify DDoS attack preparation, and (iii) a method to evaluate distributed and cooperative solutions for DDoS attack prediction. The main contribution is a system for DDoS attack prediction. Distributed processing, EWS engineering, cooperation between devices, and machine learning are the concepts that comprise the proposed system. Furthermore, the proposed system is a generic system adaptable to different network topologies. EWS engineering provides the capacity to identify the preparation of DDoS attacks. EWS engineering only relies on network traffic analysis to extract signals of DDoS attack preparation.

This work proceeds as follows. Section II presents related works to DDoS attack prediction and detection. Section III details the proposed system. Section IV presents the system evaluation. Finally, Section V concludes this work.

II. Related Works

The literature presents several approaches for DDoS attack detection. In [14], the authors developed a solution for detecting DDoS attacks on smart homes. The authors identified that Smart Homes include four device profiles distinguished by the difficulty of detecting attacks from these devices. The solution utilizes these different profiles to train a specialized boosting method of logistic model trees for each profile. During the evaluation of DDoS attacks, the solution obtained accuracy between 99.92% and 99.99%. The study of [15] proposes a solution for detecting DDoS attacks in software-defined networks. The solution uses flow attributes to determine entropy as new packets arrive. The solution identifies attacks in progress when the calculated entropy exceeds a predefined threshold. The solution showed a 98.2% of detection rate and a 0.04% false positive rate during testing.

Despite the diversity of detection solutions, the literature on DDoS attack prediction is limited. In [16], the authors analyze alerts generated by IDSs to predict DDoS attacks. The solution measures the degree of randomness of the information contained in the alerts (entropy) in a centralized way. With entropy, the system groups IDS alerts utilizing K-means, an unsupervised machine learning algorithm. The aim is to identify groups of alerts that symbolize the preparation of DDoS attacks before the attacker launches them. The solution predicts possible attacks seconds before the attack begins.

In [17], the authors propose solutions that monitor relevant texts found in online data sources. Social networks such as Twitter are relevant online data sources for the proposed analysis. The purpose of monitoring social media texts is to centralize the processing of tweets and use them to build deep learning models to predict the probability of a DDoS attack. Based on the words included in the tweets, the models predict the occurrence of some attacks the next day. The study in [18] uses metastability theory to identify early attack signals. The proposed solution captures the network traffic, prepares the data, and calculates the prediction signals carefully adapted to the area of computer networks. The solution analyzes these signals to check for evidence of future attacks and to produce alerts. The authors evaluated the proposal utilizing DARPA datasets and one scenario provided by the Czech Technical University (CTU-13) dataset [19].

The study in [20] proposes a solution focused on detecting botnets during the Command and Control (C&C) stage, and the C&C communication is evidence of potential DDoS attacks. The authors centrally analyze network traffic containing traces of botnets' actions. The authors select attributes based on network traffic from this analysis and compare the hit rate between four machine learning techniques: Random Forest, Support Vector Machines, Logistic Regression, and MLP. Utilizing 37 attributes, the authors obtained 97.8% accuracy. In [21], the authors propose one solution to predict DDoS attacks utilizing the Recurrent Neural Echo State Network (SCESN). The solution processes network traffic and forecasts network traffic behavior in the next few seconds. With future behavior, the solution predicts the error of this forecast utilizing SCESN. SCESN is pre-trained based on historical network data. The solution analyzes this error utilizing the Lyapunov exponent to predict the attack. The best result was predicting attacks 20 seconds before the attack began.

Despite these studies, much work can evolve the DDoS attack prediction. The creation of distributed and cooperative solutions, the specialization of solutions for different environments, the reduction of labeled data for model training, the application of deep learning techniques, and the creation of new studies capable of making long-term predictions with high success rates are still open questions. In addition, it is essential to reduce dependence on data collected from the Internet and alerts generated by IDS. If the data collected on the Internet does not show signs of attacks, solutions based on this data may not predict DDoS attacks. Furthermore, if IDSs do not identify malware propagation or unusual communications, solutions utilizing alerts may not work correctly. Another limitation is that some solutions need to know botnet C&C

communication methods. The solution may become obsolete if the attacker modifies how botnet communication occurs. Finally, because attackers seek to hide their activities, it is vital to research manners to identify the signals of attack preparation. These researches bring to network security management more ways to reduce the damage caused by DDoS attacks. Finally, [22] shows that the literature on DDoS attack detection resolved similar open issues to those presented by this work.

III. COOPRED DDoS System

This section introduces the COOPRED DDoS system, a cooperative system for predicting DDoS attacks. The system utilizes EWS engineering to identify DDoS attack preparation and assist in security management. This work presents a complete system to automate the identification of attack preparation signals and demonstrate the use of EWS engineering. Although EWS engineering is a step in the COOPRED DDoS system, it is worth mentioning that EWS engineering is standalone, *i.e.*, it can be utilized separately in other systems. This section details EWS engineering and the COOPRED DDoS system. The system collects and processes network traffic in a distributed way and applies EWS engineering to network traffic. It cooperates to join the signals dispersed by distributed processing and classifies them with machine learning models. Thus, the COOPRED DDoS system addresses issues identified in the previous section, such as the independence of data collected on the Internet, creating a cooperative and distributed solution, utilizing deep learning to predict DDoS attacks, and the proposal is independent of the botnet C&C communication.

The COOPRED DDoS system utilizes two software instances: agent and intelligence center. Agents are software instances programmed to collect and process network traffic in a distributed manner. Fig. 1 highlights the operation of agents, where agents collect network traffic from different network segments; i.e., agent A collects data from nodes present on subnet A. After collection, agents perform EWS engineering to transform network traffic into EWS to anticipate attacks. The output of EWS engineering represents a DDoS attack before it begins. The cooperation ensures that all agents contribute to predicting attacks, keeping the relationships between data previously separated by the distributed processing performed by the agents. All agents send the signals to the intelligence center to cooperate. The intelligence center is the second instance of software utilized in the COOPRED DDoS system. The intelligence center utilizes machine learning to identify changes in characteristics of distributions and identify signals of possible DDoS attacks because it is not a trivial problem. Fig. 1 shows the system and presents the five steps necessary to perform the prediction of attacks. The steps are system preparation, network traffic collection, signal engineering execution, signal centralization, and administrator notification.

A. System setup

It is necessary to prepare the COOPRED DDoS system defining the configuration of agents and the intelligence center because these configurations change according to the characteristics of the networks in which the system will operate.

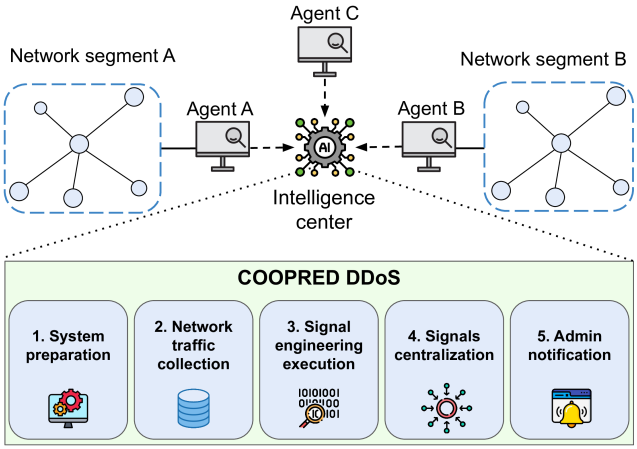


Fig. 1. Proposal overview.

The first definition is the configuration of the agents. Agents are the software instances that capture network traffic and execute EWS engineering in a distributed manner. The number of agents varies depending on hardware availability, network topology, and network size. The location of the agents will depend on the network where the COOPRED DDoS system will operate. The literature shows that the position of agents can be near the origin of the attack, in intermediary networks, or near the victim network [23]. Furthermore, the literature proposes methods to define the positioning of agents [24]. This work does not explore these issues as they are not the main focus. This work follows the results of the literature.

Another definition is the network traffic attributes collected by the agents. Since the attackers want to hide their actions, the prelude to the attack can impact a few network traffic attributes. The system accepts attributes from network traffic if it is a numeric variable because EWS engineering only accepts numeric variables. Numerical variables are values measured from observation, i.e., the amount or the average size of packets. Attribute selection is not a trivial task; therefore, this version of the COOPRED DDoS system utilizes the most relevant attributes defined by [25] because the results indicate that these attributes can represent the C&C phase. However, selecting the most suitable attributes for EWS engineering is future work. Table I shows the attributes utilized by the system to predict DDoS attacks; however, the user can define the attributes that the system will use. COOPRED DDoS system collects these attributes from each network segment, i.e., the number of packets sent in the monitored network segment. This work highlights the number of packets sent, the ratio of the size of the received packets, and the ratio of the time interval of the received packets because the Computational and network seCurity SCience research team utilized them in previous works. Finally, the system does not use the payload data to preserve users' privacy.

Choosing capture cycles is the last definition of agents. Capture cycles correspond to the time all agents have to start and end each data collection and processing (EWS engineering). Agents need to sync with the current date and time because the intelligence center will use the date and time sent by the agents to classify the signal processing.

TABLE I
NETWORK TRAFFIC ATTRIBUTES AVAILABLE IN THE SYSTEM [25].

Description
Number of packets sent
Ratio of packets with flag RESET in a session
Ratio of the sent packets having the size of 100-199 bytes
Ratio of the sent packets having the size of bytes
Ratio of packets with flag S in a session
Variance of sent packets
Maximum interval time in received packets
Variance of interval time of received packets
Ratio of the sent packets having the size of 500-599 bytes
Variance of packet size of the received packets

The COOPRED DDoS system uses different time units such as minutes, seconds, and milliseconds. Small capture cycles are insignificant to performing the attack prediction due to little information available. However, the system can finish processing large cycles after the attack begins, losing the advantage of performing DDoS attack prediction. The default value is one second; however, the user can change it.

The intelligence center is the second instance of software to be prepared. The intelligence center utilizes machine learning techniques to classify the cycles. The intelligence center supports shallow and deep machine learning techniques. The default machine learning technique is Adaptive Boosting (Adaboost). This technique performs classification and regression tasks and follows the strategy of training different instances of the same machine learning technique and combining the outputs of all of them to generate an output. This work chose Adaboost because it has fast training, classification, and relevant results reported in the literature [26]. Finally, the intelligence center must be accessible to the agents since they need to communicate with the intelligence center to sync the definitions and send the processed signals.

B. Network traffic collection

To start collecting network traffic, agents synchronize the definition of network traffic attributes, the capture cycle time, and actual time. Synchronization is vital as the intelligence center uses the moment of capture to identify signals of possible DDoS attacks. Synchronization provides the new definitions made by the user to the agents. Each agent uses a network monitor to collect network traffic. Computers or virtual machines, for example, support the network monitor. The network monitor of agents operates in packet analysis mode for real-time data processing and offline mode, utilizing historical data as input. Network segment traffic is mirrored to the agent designated to perform the analysis. For example, in Fig. 1, all network traffic from segment A is mirrored to Agent A. Therefore, the agents collect the attributes (Table I) from a network segment traffic. The agent repeats the collection process while the system is active. At the end of each capture cycle, the agent has the data collected on the network in the form of previously defined attributes.

C. Signal engineering

The agents organize data in the form of time series to execute EWS engineering and process network traffic. A time series comprises observations performed sequentially in time [27]. Each agent has one time series for each attribute collected in the network traffic. Agents renew the time series to save resources by removing the oldest data. This action occurs when the time series reaches the maximum value defined by the user based on the device's hardware capability.

After preparing the time series, the agents execute EWS engineering to process the attributes collected from the network traffic. The agents calculate EWS on the time series of the attributes, and the result of this action identifies signals that precede critical transitions [28], [29]. A critical transition is an extreme change between states of a system [29]. During a critical transition, the observed system goes through moments of instability; this instability can interrupt the services or even guide species extinction [29]. This work associates critical transitions with DDoS attacks. Agents repeatedly transform network traffic attributes into EWS at each collection cycle. At the end of EWS engineering, agents send EWS, a local identifier, and the start and end date of the capture cycle to the intelligence center. Agents only send EWS and not traffic attributes to prevent adversarial learning [30].

Kurtosis (Eq. 1) is the first EWS utilized in this work. The term N represents all-time series observations. The $\hat{y}$ result, defined by Eq. 2, is necessary to calculate Kurtosis. The term x_t refers to each item in the time series. Furthermore, the term $\bar{x}$ refers to the simple arithmetic mean [31]. There are controversies about the interpretation of Kurtosis [32]. Despite the different interpretations of Kurtosis, one of the most widespread and controversial is the value of Kurtosis to the degree of flattening of the time series curve (peak of the distribution). Currently, some authors argue that the interpretation of the Kurtosis value should leave aside the relationship with the distribution peak and focus on the distribution tail [33]. Even with the different interpretations of Kurtosis and the fact that Kurtosis was not specific to identify EWS, Kurtosis produces EWS because there are indications that the value of Kurtosis may grow or be present at peaks close to critical transitions [28], [34]. In this way, distribution can change the structure of near-critical transitions [28].

$$Kurtosis = \frac{(N-1)}{(N-2)(N-3)}(N-1)\hat{y} + 6 \tag{1}$$

$$\hat{y} = \frac{N\sum(x_t - \bar{x})^4}{[\sum(x_t - \bar{x})^2]^2} \tag{2}$$

Skewness is the second EWS utilized in this work (Eq. 3). The term x_t refers to each time series sample. The N represents the total number of observed items. The $\bar{x}$ refers to the simple arithmetic mean, and the term s represents the standard deviation [35]. Skewness measures the degree of asymmetry of observations in a time series. The degree of symmetry indicates whether the time series is symmetrically centered on the mean, skewed to the left or the right [35]. Although not an equation designed to produce EWS, increases in distribution asymmetry indicate a reliable EWS [36].

$$Skewness = \frac{N\sum_{t=1}^{N}(x_t - \bar{x})^3}{(N-1)(N-2)s^3} \tag{3}$$

The third EWS utilized in this work is obtained by calculating the power spectrum and is called the maximum

power spectrum (S_{max}). When processing signals utilizing the power spectrum, researchers identify the existence of repetitive patterns and correlations. It is possible to estimate the power spectrum utilizing the Fourier transform [37]. It is necessary to calculate the arithmetic mean of overlapping periodograms. In [37], the author defines the periodogram as in Eq 4, where k varies as $k = 1, 2, 3...K$, and K is the total of periodograms. In [37], the author defines f_n as $f_n = \frac{n}{L}$ where $n = 0, ..., L/2$. L represents the size of the segment analyzed by the periodogram, and $U = \frac{1}{L}\sum_{j=0}^{L-1} W^2(j)$. $W(j)$ represents the time series data analyzed by the periodogram, that is, $j = 0, ..., L-1$. $A_k(n)$ is the Fourier transform defined by Eq. 5. It is necessary to identify the maximum value between the last power spectrum observations to calculate S_{max} [38].

$$I_k(f_n) = \frac{L}{U}|A_k(n)|^2 \tag{4}$$

$$A_k(n) = \frac{1}{L}\sum_{j=0}^{L-1} X_k(j)W(j)e^{-2kijn/L} \tag{5}$$

S_{null}, S_{fold}, and S_{hopf} are variations of S_{max}. The observed system is probably far from critical transitions if the S_{null} (Eq. 6) presents a relatively flat spectrum. The S_{fold} (Eq. 7) and S_{hopf} (Eq. 8) grow as critical transitions approach. The real part of the eigenvalues of the Jacobian matrix of the system corresponds to the parameters λ and θ. σ corresponds to the standard deviation of the distribution, and ω corresponds to the power spectrum. Finally, ω_0 is a constant that takes on different values between models [38].

$$S_{null}(\omega;,\sigma) = \frac{\sigma^2}{2\theta} \tag{6}$$

$$S_{fold}(\omega;\sigma,\lambda) = \frac{\sigma^2}{2\theta}\frac{1}{\omega^2+\lambda^2} \tag{7}$$

$$S_{hopf}(\omega;\sigma,\mu,\omega_0) = \frac{\sigma^2}{4\theta}\left(\frac{1}{(\omega-\omega_0)^2+\mu^2}+\frac{1}{(\omega+\omega_0)^2+\mu^2}\right) \tag{8}$$

This work chooses Kurtosis, Skewness, and S_{max} because they can present extreme variations in the observed values. Fig. 2 shows this possible variation. Frame A presents a system undergoing a critical transition. While the observed system is far from the critical transition (frame B), the data distribution has a flat pattern. When the observed system is near the critical transition, the data distribution characteristics can change to a peaked pattern (frame C). These variations indicate changes in the data distribution, and these changes can anticipate critical transitions, consequently predicting DDoS attacks. The S_{null}, S_{fold}, and S_{hopf} are EWS indicating what type of critical transition is approaching. The purpose of choosing these metrics is to complement the data analysis. Therefore, it is possible to prevent errors and help define the type of the coming attack with this information.

D. Signals centralization

Synchronizing the received signals is the first challenge for the intelligence center. The COOPRED DDoS system needs to synchronize them because the latency between the agents can be different during the classification of the signals. The intelligence center waits to receive signals from all agents for three capture cycles to prevent errors. The intelligence center marks the agent offline if the agent does not send the signals. Therefore, the intelligence center will not delay the prediction (lines 2 to 11 of Algorithm 1). The three-cycle timeout is the system default; however, this setting is changeable.

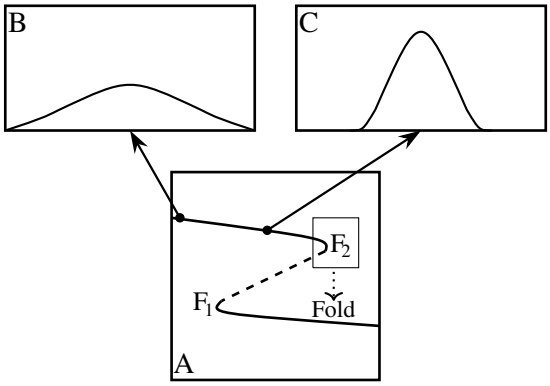


Fig. 2. Variation of data distribution in a critical transition [36].

Missing signals and different scales are inconsistencies that degrade system performance. The system preprocesses the missing signals in two ways. The system removes signals with missing inputs or adds a value based on the median, mean, or a constant, i.e., "-1" (line 12 of Algorithm 1). The COOPRED DDoS system uses standardization or Min-Max rescaling to handle data at different scales. Standardization rescales attributes by removing the mean of previous observations divided by the standard deviation of previous samples. Eq. 9 defines the standardization, where the $X(j)$ term represents the value of each sample of the analyzed attribute, the $\bar{x}$ term represents the mean, and the s term is the standard deviation. Min-Max is the default strategy of the COOPRED DDoS system for data scaling. The Min-Max strategy scales the attribute values within a predefined range. The range value can be between zero and one or utilize the real values of the base. Eq. 10 defines the Min-Max strategy. The $X(j)$ represents the value of each sample of the analyzed attribute; the min_A and max_A are respectively the smallest and the biggest observed value for the attribute. The min'_A and max'_A represent the new range [39], for example, zero and one (line 13 of Algorithm 1).

$$Standardization = \frac{(X(j)-\bar{x})}{s} \tag{9}$$

$$Min-Max = \frac{X(j)-min_A}{max_A-min_A}(max'_A-min'_A)+min'_A \tag{10}$$

After preprocessing, the intelligence center performs the classification of signals (line 14 of Algorithm 1). Classification guarantees cooperation between all agents to generate evidence to predict DDoS attacks because the intelligence center utilizes the EWSs of all agents in the same capture cycle to predict attacks. The classification analyzes the received signals searching for changes in the data distribution that might anticipate a critical transition. This work relates critical transitions with possible DDoS attacks. Therefore, the COOPRED DDoS system correlates the presence of these changes with signals of possible attacks. The challenge of classification is to

identify the changes present in the signals. For this, this work uses machine learning techniques. At each network traffic collection cycle, the intelligence center classifies the union of signals to identify precursor variations to DDoS attacks.

Algorithm 1 Intelligence Center Pseudocode

```
while True do
  cycle_time, waiting_time, ml_model ← get_definitions()
  wait(waiting_time)
  signal, complete_cycle ← get_signals()
  if not complete_cycle then
    wait(cycle_time * waiting_time)
    signal, complete_cycle ← get_signals()
    if not complete_cycle then
      set_agents_offline(signals)
    end if
  end if
  signals ← process_missing_signals(signals)
  signals ← rescale_signals(signals)
  has_attack_warning ← ml_model.predict(signals)
  if has_attack_warning then
    notify_administrators()
  end if
  set_signals_as_processed(signals)
end while
```

E. Admin notification

The COOPRED DDoS system sends notifications to network administrators when the intelligence center output corresponds to a possible attack (lines 15 to 17, Algorithm 1). Notifying a possible attack occurs when the intelligence center identifies changes in data distribution. There are multiple options for this notification, such as sending text messages or push notifications. One option to automate incident response is to send a message of the type JavaScript Object Notation. It serves as an input to automating a firewall or alerting systems. The default action is to notify network administrators by email.

IV. Performance Evaluation

This section describes the performance evaluation of early warning engineering signals and the COOPRED DDoS system. This work conducted four experiments, Experiments 1 and 2 aimed to verify if the COOPRED DDoS system, utilizing EWS engineering, performs the prediction of DDoS attacks. Experiments 3 and 4 aim to verify if the COOPRED DDoS system equipped with a deep learning technique can improve the results obtained in Experiments 1 and 2. Experiments 3 and 4 improve the results in two manners. Firstly, it is possible to reduce the number of errors obtained in Experiments 1 and 2. Furthermore, increasing the prediction time of attacks in Experiments 3 and 4 concerning the previous experiments can improve the results. All experiments were run on a computer with an i5 processor, 240-GB solid-state drive, and 8-GB random-access memory.

This work uses four agents to process network traffic and evaluate distribution and cooperation. This number of agents was chosen only to evaluate the cooperation between the agents in the system; however, the COOPRED DDoS system accepts other numbers of agents. The symbolic name of each agent is AG-1, AG-64, AG-128, and AG-192. AG-1 was responsible for processing packets originating from devices with addresses from 0.0.0.1 to 63.255.255.254. The AG-64 processed traffic from devices in the range of 64.0.0.1 to 127.255.255.25. AG-128 was responsible for the range from 128.0.0.1 to 191.255.255.254. The AG-192 analyzes traffic generated by devices with addresses from 192.0.0.1 to 255.255.255.254. This segmentation ensures that all the evaluation does not benefit any segment. Therefore, because the topology of the evaluated networks is not available, this segmentation provides impartiality in data processing because all agents have the same number of possible addresses. However, in real environments, the division of network traffic must respect the network topology and hardware limitations.

It is necessary to define the network attributes and the length of the capture cycle to evaluate the proposal. Subsection III-A mentions that the system can utilize ten network traffic attributes. However, the number of packets sent by nodes in the network is the only network attribute utilized in the experiments. Therefore, each agent collects the number of packets sent by all devices on the network segment that the agent analyzes. This work chose this attribute because it is the most relevant presented by [25]. Furthermore, this work utilized the default value for the capture cycle as one second to evaluate EWS engineering in different scenarios. Therefore, the agents applied EWS engineering in the collected network traffic every second. Finally, the literature supports utilizing the number of packets sent per second for DDoS attack prediction [21].

The last configuration defines the machine learning technique that will integrate the intelligence center. For Experiments 1 and 2, this work utilized Adaboost as a machine learning technique to classify the signals, and for Experiments 3 and 4, the technique chosen was the Multilayer perceptron (MLP). Experiments 1 and 2 tested Adaboost utilizing the default configuration defined by the library developers [40], except for the number of estimators. Adaboost's default configuration utilizes the Decision Tree as the estimator to create the various instances present in the model. These two experiments utilized just five estimators when the default is 50. This action aims to utilize the same configuration for Experiments 1 and 2. Experiments 3 and 4 tested the MLP utilizing the default settings, except for the optimization approach and the number of hidden layers. The optimization approach chosen was L-BFGS because the library authors recommend utilizing it when the analyzed dataset does not have several thousand records, as in experiments [40]. Experiments 3 and 4 utilized three hidden layers, 33, 17, and 11 were the number of neurons for the first, second, and third hidden layers, respectively.

This work utilizes accuracy, precision, and recall to evaluate the system's performance. It is necessary to measure the total number of true positive (TP) and true negative (TN), the total false positive (FP) and false negative (FN), and the total of observations (N) to calculate the metrics. The accuracy evaluates the classifications of the system (Eq. 11), and it is a standard metric that helps understand the results. However, as attackers hide their actions, few signs of attack preparedness

exist as attackers hide their actions. Therefore, accuracy above 90% may present a false sense of good results because the system can correctly classify all observations of the majority class and misclassify all observations of the minority class and have accuracy above 90%. This work utilizes precision and recall to complete the analysis of the results. Precision indicates the relationship between the observations labeled by the system for a specific type and how many were of the assumed type (Eq. 12). The recall presents the relationship between all the expected observations of the specific type and how many observations of this type the system correctly classified (Eq. 13).

$$Accuracy = \frac{TP + TN}{N} \quad (11) \qquad Precision = \frac{TP}{TP + FP} \quad (12)$$

$$Recall = \frac{TP}{TP + FN} \quad (13)$$

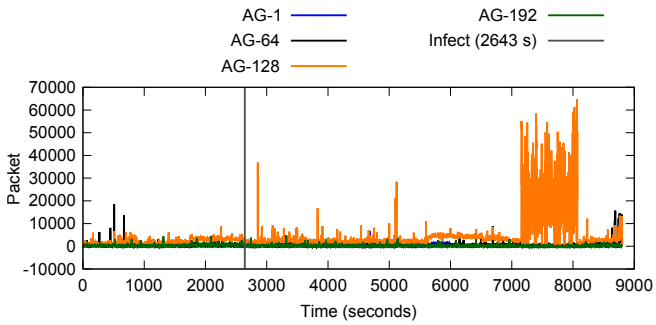
A. Experiment 1

Experiment 1 utilizes two scenarios from the CTU-13 dataset. This dataset has 13 scenarios of botnets' actions. The scenarios have web pages with the bot infection moment and when the DDoS attacks started. Among the available scenarios, the network traffic captured in Scenario 10 preceded the traffic captured in Scenario 11. Therefore, this work utilized Scenario 10 as a training set and Scenario 11 as a test set. Finally, the sets have 8803 and 972 seconds for training and testing.

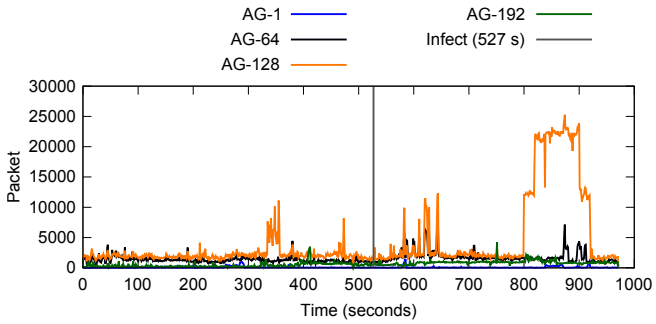
The following action was to segment the network traffic and count the number of packets sent per second in all network segments. Fig. 3 shows the packet count in Experiment 1. Throughout the training set (Fig. 3(a)), the AG-128 has the highest peaks in the number of packets sent. This behavior can be justified because all the attack bots had IP addresses in the AG-128 range. The test set presents similar behavior (Fig. 3(b)), where the AG-128 is the one that has the most packet peaks. Again, the bots are present in the IPs range analyzed by the AG-128, which justifies this agent having more traffic peaks. Finally, in the training and testing set (Fig. 3), the number of packets sent collected by AG-128 increases after bot infection. This increase is due to the action of bots that began to exchange packets on the network.

Each agent utilized the number of packets sent per second to perform EWS engineering. Fig. 4 shows the variation of kurtosis processed by the AG-128 in the training and test sets to simplify the presentation of the results; however, all data are available online[1]. In Fig. 4(a), the red lines indicate the start and end of the attack. Moreover, this figure presents three intense variation points before the attack. In the second 2854, the kurtosis value varies from 2.6 to 304.9. The figure shows variations in seconds 4614 and 5100; however, the variation is smaller compared to the second 2854. Fig. 4(b) presents the variation of kurtosis for the test set. This figure shows three intense variation points before the attack. In the second 336, the kurtosis value varies from 7.2 to 56.9. The figure shows variations in seconds 549 and 576; however, the variation is smaller than in the second 336.

[1]https://github.com/andersonneira/tnsm-data



(a) Packet count per second in training.

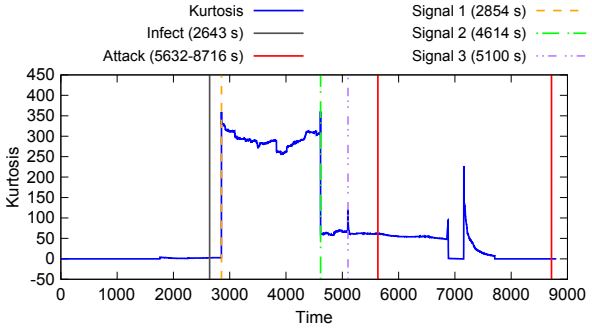


(b) Packet count per second in the test.

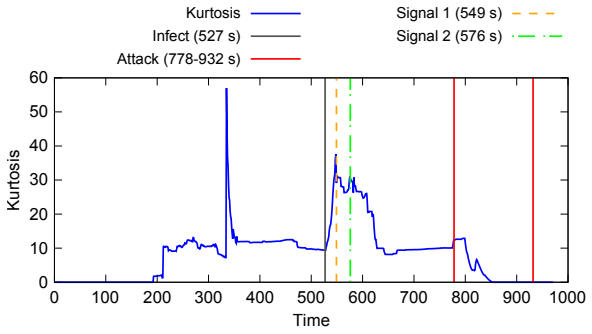
Fig. 3. Packet count in Experiment 1.

This work has realized the missing data imputation with the constant -1, the Min-Max rescaling, and the labeling of the training and test sets to execute the training of the intelligence center. This work has utilized the extreme values presented before, jointly with the moment of bot infection and malicious traffic, to label the training and test sets and make it possible to use supervised machine learning. Labeling the training and test sets in the seconds shown in Fig. 4 enables the machine learning technique to identify changes in data distributions. For the training set (Fig. 4(a)), the set documentation indicates that the bots' infection occurred at second 2643. Therefore, the bots influence the extreme values obtained in seconds 2854, 4614, and 5100. Therefore, this work labels these seconds as DDoS attack warnings. For the test set (Fig. 4(b)), the infection starts at 527 seconds, which means that the bots did not influence the peak at the second 336. Therefore, this work labeled the second 336 as a normal second. The bots influence the extreme values obtained in seconds 549 and 576. Therefore, this work has defined these two seconds as attack warnings.

Fig. 4 shows the imbalance present in the training and test sets. This imbalance is because the training set has 8803 seconds of data, and only three of those seconds are attack warnings. The test set has 972 seconds and only two seconds are attack warnings. The problem of data imbalance is relevant because some techniques may favor the majority class, misclassifying attack warnings [41]. For example, if the system classifies every second as a normal second in the test set, the system achieves 99,79% of accuracy; however, the system identified no attack warning. Therefore the COOPRED DDOS system would not achieve its function. Despite this challenge, this work does not reduce the imbalance because this work aims to evaluate if the system works with the imbalanced data.



(a) Kurtosis per second in training.



(b) Kurtosis per second in the test.

Fig. 4. EWS engineering in Experiment 1.

The intelligence center classified every second of the test set as a normal second or attack warning. The normal seconds are the second ones with no extreme values in the EWS. Attack warnings are the seconds with extreme values in the EWS. Table II shows that the system correctly classified 968 seconds as normal seconds. The system correctly classified the second 576 as an attack warning. It wrongly classified the attack warning labeled on the second 549 as the normal second (FN). The system classified two normal seconds as attack warnings (FP). The accuracy is 99.69%, and for the attack warnings, the precision is 33%; this was because the system classified two normal seconds as attack warnings. The precision of normal seconds was 99.89%. The recall for the attack warnings class is 50% as one of the two attack warnings has been identified, while the recall for the normal second is 99.79%.

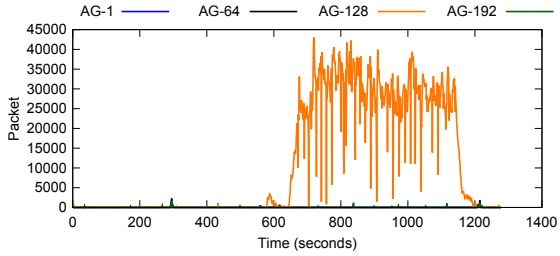
TABLE II
RESULTS IN EXPERIMENT 1.

Confusion matrix		Real class	
		Attack warning	normal second
Hypothetical class	Attack warning	1	2
	Normal second	1	968

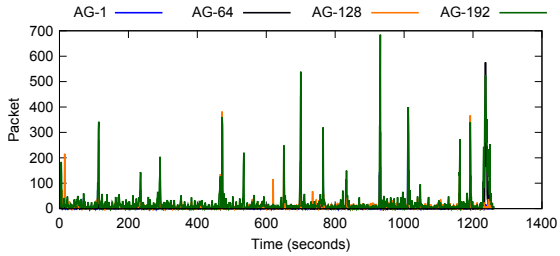
B. Experiment 2

Experiment 2 uses two scenarios from the DDoS evaluation dataset (CICDDoS2019) [42]. This dataset has 19 DDoS attack scenarios, and the documentation shows the start and end times of attacks. The "UDP Attack" and "UDP-Lag Attack" scenarios are sequential. For this reason, this work utilized the "UDP Attack" scenario as a training set and the "UDP-Lag Attack" scenario as a test base. The set sizes are 1277 and 1260 seconds for training and testing, respectively. Finally, this work selected these attacks because they are different types of attacks, making prediction difficult.

The following action was to segment the network traffic and count the number of packets sent per second. Fig 5 shows the packet count in Experiment 2. The AG-128 has the biggest peaks in the number of packets sent in the training set (Fig. 5(a)). This behavior can be justified because all the bots carrying the attack had IP addresses in the AG-128 range. The test set behaves differently; the AG-128 has some traffic peaks; however, the agent with the most peaks is the AG-192 (Fig. 5(b)). The total packet processed by AG-128 is 15020 packets and by AG-192 is 28593. A possible explanation for this behavior is the possibility that the attack failed.



(a) Packet count per second in training.



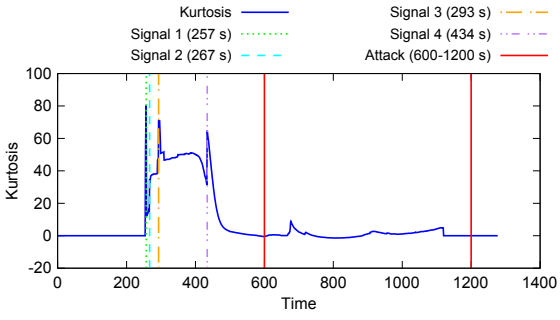
(b) Packet count per second in the test.

Fig. 5. Packet count in Experiment 2.

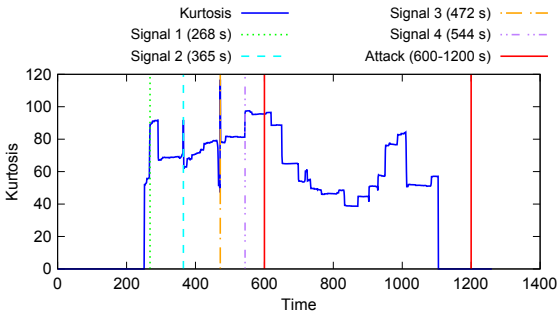
The following action was to apply EWS engineering to the network traffic. Fig. 6 shows the result of the kurtosis processed by the AG-128 to simplify the presentation of the results; however, all processed data are available online[2]. Fig. 6(a) presents the variation of kurtosis in training. The red lines indicate when the attack began and finished. It is worth mentioning that in the second 434, the kurtosis value varies from 31.3 to 64.7. In seconds 257, 267, and 293, the figure shows variations; however, the variation is smaller when compared to the second 434. Fig. 6(b) shows the variation of the kurtosis signal for the test set. In the second 472, the kurtosis value varies from 47.2 to 116.7. The test set shows other peaks at seconds 268, 365, and 544. Finally, Fig. 6 shows imbalance data; however, as in Experiment 1, no action was taken to resolve the imbalance data.

This work realized the imputation of missing data with the constant -1, the Min-Max rescaling, and the labeling of the training and test sets to execute the training of the intelligence center. This work utilized the extreme values presented previously to label the training and test sets. Labeling the bases in the seconds shown in Fig. 6 enables the machine learning technique to identify changes in data distributions.

[2]https://github.com/andersonneira/tnsm-data



(a) Kurtosis per second in training.



(b) Kurtosis per second in the test.

Fig. 6. EWS engineering in Experiment 2.

The official documentation does not specify when bots are infected; however, the training and test sets have bot traffic from the beginning. In this way, the bot traffic influenced analysis even before the attack began. In the case of training (Fig. 6(a)), extreme values obtained in seconds 257, 267, 293, and 434 are the labels that warn of a possible DDoS attack. For the test set (Fig. 6(b)), labels representing DDoS attacks were added at seconds 268, 365, 472, and 544.

The intelligence center classifies every second of the test set as a normal second or attack warning. As shown in Table III, the system correctly classified 1256 seconds as normal seconds. The system correctly classified the second 472 as an attack warning, and it does not show FPs. Accuracy is 99.76%, and precision for the class of warnings is 100%, as there are no FPs. Precision was 99.76% for normal seconds because the system has classified three attack warnings as normal seconds. The recall for the warnings is 25% because the system correctly classified one of the warnings, while the recall for the class of normal seconds is 100%.

TABLE III
RESULTS IN EXPERIMENT 2.

Confusion matrix		Real class	
		Attack warning	normal second
Hypothetical class	Attack warning	1	0
	Normal second	3	1256

C. Experiment 3

Experiments 1 and 3 differ in the machine learning technique present in the intelligence center. In Experiment 1, the intelligence center uses Adaboost, a shallow learning-type machine learning technique. In Experiment 3, the intelligence center uses MLP, a deep learning-type machine learning technique. Therefore, Experiment 3 uses the same training and testing sets and labels as Experiment 1. Fig. 4 shows the training and testing labels. The objective of replicating Experiment 1 by modifying the intelligence center to use deep learning is to verify if deep learning techniques can improve the results obtained previously.

Table IV presents the results obtained in Experiment 3. The first cell on the main diagonal indicates that the intelligence center successfully identified an attack warning. The second correctly classified as an attack warning was 549. The other component of the main diagonal indicates that the vast majority of normal seconds were correctly classified. The negative point of these results is the total number of errors obtained. The intelligence center misclassified 11 seconds, ten normal seconds misclassified as a warning attack, and an attack warning erroneously classified as a normal second. The ten seconds wrongly classified as an attack were seconds 548, 551 to 558, and 582. The accuracy of the intelligence center is 98.87%. The precision for the warnings class is 9.09%; this is because, of the 11 times that the intelligence center classified a second as an attack warning, only one classification was correct. The precision for the normal second class is 99.89%. The recall for the class of warnings is 50%, and the recall for the class of normal seconds is 98.96%.

TABLE IV
RESULTS IN EXPERIMENT 3.

Confusion matrix		Real class	
		Attack warning	normal second
Hypothetical class	Attack warning	1	10
	Normal second	1	960

D. Experiment 4

Experiments 2 and 4 differ in the machine learning technique present in the intelligence center. In Experiment 2, the intelligence center uses Adaboost; in Experiment 4, the intelligence center uses MLP. Therefore, Experiment 4 uses the same data and labels as Experiment 2. Fig. 6 shows the training and testing labels. The objective of replicating Experiment 2 by modifying the intelligence center to use deep learning is to verify if deep learning techniques can improve the results obtained previously.

Table V presents the results obtained in Experiment 4. Table V indicates that the experiment has successfully identified an attack warning. The second correctly classified as an attack warning was 365. The results indicate that the vast majority of normal seconds were correctly classified. As with Experiment 2, the intelligence center has missed three out of four attack warnings. In addition, the intelligence center equipped with MLP classified two normal seconds as an attack warning. The two seconds wrongly classified as an attack were seconds 537 and 542. The intelligence center's accuracy is 99.60%. The precision for the class of warnings is 33%; this is because, of the three times that the intelligence center classified a second as an attack warning, only one classification was correct. The precision for the normal second class is 99.76%. The recall for the class of warnings is 25%, and the recall for the class of normal seconds is 99.84%.

TABLE V
RESULTS IN EXPERIMENT 4.

Confusion matrix		Real class	
		Attack warning	normal second
Hypothetical class	Attack warning	1	2
	Normal second	3	1254

E. Discussion

In Experiment 1, the proposal correctly classified one second as an attack warning (Table II). The second 576 in the test set was labeled an anomaly capable of indicating a possible DDoS attack. As shown in Fig. 7, the second 576 precedes the attack by 3 minutes and 22 seconds. This result is significant because the system notified the possibility of a DDoS attack only 49 seconds after the infection. Furthermore, the peak of packets during the attack happens in the second 875. Therefore, the COOPRED DDoS system gives network administrators 4 minutes and 59 seconds before the peak of the attack (Fig. 7). In order to further emphasize the importance of these results, this work presents a hypothetical comparison. Assuming that a hypothetical DDoS attack detection solution identified the attack in the exact second after the attack began, network administrators would have 1 minute and 37 seconds until the attack reached its peak. In contrast, the COOPRED DDoS system would provide 4 minutes and 59 seconds until the peak of the attack. Moreover, it is appropriate to discuss the number of alerts generated by the COOPRED DDoS system. The system would be more confident to network administrators if it correctly identified other EWS. Therefore, administrators would receive two alerts about the occurrence of a possible DDoS attack, not just one of how it happened.

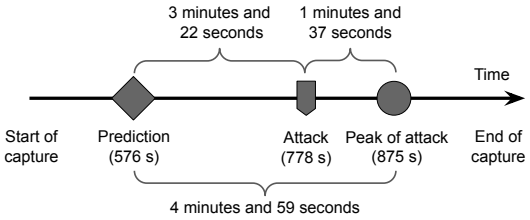


Fig. 7. Prediction time in experiment 1.

Also, in Experiment 1, the COOPRED DDoS system classified seconds 577 and 578 as attack warnings. This incorrect classification means that the system wrongly classified two normal seconds as an attack signal. This misclassification is serious because this moment does not represent an anomaly in EWS that would justify an attack alert. Therefore, in a scenario where no attacks occur, the COOPRED DDoS system would predict a non-existent attack, encouraging the security team to take undue actions, wasting time and money. In Experiment 1, seconds 577 and 578 follow the EWS (second 576); this reduces the magnitude of the error because they reinforce the attack prediction. However, they are still wrong predictions.

In Experiment 2, the COOPRED DDoS system utilizing the EWS engineering identified one second as an anomaly indicating a possible DDoS attack. This second is the 472 of the test set. Thus, the system predicts the attack 2 minutes and 8 seconds before the launch of the attack. This result is significant because it gives network administrators an advantage over the attack. This advantage would be more significant if the system identified the attack at second 268 or 365 because it precedes the second 472. In addition, classifying other attack warnings would increase confidence in the system.

Comparing the results obtained in Experiments 1 and 3 (Tables II and IV, respectively), Experiment 1 is more accurate than Experiment 3. If this work analyzes the results utilizing the total errors and correct classifications, Experiment 1 shows fewer errors. When the intelligence center operated with Adaboost, the total errors were only three (two FPs and one FN). When the intelligence center operated with MLP, the total errors were 11 (10 FPs and one FN). However, if the analysis focuses on when the prediction occurred, the intelligence center utilizing MLP has an advantage because the attack warning identified in Experiment 3 is prior to the attack warning identified in Experiment 1. While in Experiment 3, the identification of preparation signals for a DDoS attack would occur in the 549th second, in Experiment 1, the attack prediction would occur in the second 576. Hence, utilizing MLP in the intelligence center, the COOPRED DDoS system could provide 27 seconds longer than the intelligence center operated with Adaboost. Therefore, network administrators received a notification about the possibility of a DDoS attack occurring 3 minutes and 49 seconds before the start of the attack. The result is more relevant when comparing it with the peak of the attack; the COOPRED DDoS system anticipated the attack's peak by 5 minutes and 26 seconds. As mentioned, that vantage of 27 seconds cost eight FP errors.

Comparing the results of Experiments 2 and 4 (Tables III and V, respectively), the difference is that Experiment 4 has two FPs while Experiment 2 has zero. As discussed, the COOPRED DDoS system needs to avoid FP errors. Following this prism, the intelligence center results utilizing Adaboost are superior. However, it is opportune to analyze the moment the system performs the prediction. In Experiment 2, the prediction occurred at second 472, while in Experiment 4, the prediction occurred at second 365. This difference between the results shows an increase of 1 minute and 47 seconds concerning the prediction performed in Experiment 2. Therefore, the network administrators received a notification about a DDoS attack 3 minutes and 55 seconds before the attack started. As in Experiment 3, the cost of increasing the prediction time is the increase in FPs errors. However, the increase in FPs in Experiment 4 is significantly smaller than in Experiment 3.

The results indicate that the intelligence center equipped with Adaboost was more accurate than the MLP in predicting DDoS attacks; however, the MLP predicted the attack earlier than Adaboost. This work analyzed the attributes that most impact system decisions using the SHAP [43] tool to hypothesize the reasons for this behavior. One reason for the different results obtained in the experiments is how machine learning techniques utilized the data. In Experiment 1, kurtosis of AG-64, skewness and kurtosis of AG-128, and kurtosis of AG-192 are the attributes that most influenced the predictions. In Experiment 2, the kurtosis of AG-128 and AG-192 are the attributes that most influenced Adaboost decisions. In Experiment 3, skewness and kurtosis of AG-1, skewness and kurtosis

of AG-64, skewness and kurtosis of AG-192, and kurtosis of AG-128 influenced the MLP decisions. The S_{fold} of AG-1, S_{fold} of AG-64, skewness, kurtosis, S_{max}, and S_{fold} of AG-128, S_{max} and S_{null} AG-192 influenced MLP decisions. Therefore, fewer attributes influenced Adaboost, which may have made the model more accurate than MLP. Furthermore, through the cooperation of the signals collected by all agents, the MLP can predict the attack before Adaboost, even though the diversity of attributes may have made it challenging to identify the preparation signals for DDoS attacks. However, future works can confirm this hypothesis.

This work compares the COOPRED DDoS system with the ANTE system [44] to emphasize the relevance of the results. This work compared these systems because the authors verified the possibility of predicting attacks in [44]. Furthermore, the COOPRED DDoS system develops the future work of [44], specializing in DDoS attack prediction and making the architecture distributed and cooperative. The comparison shown in Table VI utilizes the total time that the solutions anticipate the attack, the manner the data is processed, the versatility in machine learning use, and the detail of the results. The purpose of the ANTE system is to identify botnets in advance. For this, the ANTE system analyzes network traffic to identify bots to notify network administrators before the consequences of the attack are irremediable. Among the evaluations of the ANTE system, one occurred in Scenario 11 of CTU-13, as well as in Experiments 1 and 3 (Subsections IV-A and IV-C). The ANTE system predicted the attack 30 seconds before the attack, and the COOPRED DDoS system predicted the attack 3 minutes and 22 seconds before the attack. Therefore, the COOPRED DDoS system gives administrators more time to stop the attack. Another advantage of the COOPRED DDoS system is related to data processing. While the COOPRED DDoS system processes the data distributed, the ANTE system centralizes the data processing. The centralization of processing can cause bottlenecks and degrade data processing. One common characteristic between the COOPRED DDoS systems and the ANTE system is the versatility in utilizing different machine learning techniques. Both systems are capable of utilizing different techniques to perform data processing. Finally, the COOPRED DDoS system notifies of the possibility of the attack, while the ANTE system identifies possible bots.

TABLE VI
COMPARISON BETWEEN COOPRED DDOS AND ANTE SYSTEMS.

System	Prediction time	Distributed processing	Different ML	Details
COOPRED DDoS (Adaboost)	3 min 22 s	✓	✓	✗
COOPRED DDoS (MLP)	**3 min 49 s**	✓	✓	✗
ANTE [44]	0 min 30 s	✗	✓	✓

This work compares the results of [45] to highlight the relevance of the results obtained in this work. In [45], the proposed solution predicted the attacks in Scenarios 10 and 11 of the CTU-13 dataset. Unfortunately, the authors did not present the results for Scenario 11, the same scenario that this work utilized in Experiments 1 and 3. However, the authors present that the solution predicted a DDoS attack in Scenario 10 with 5 minutes and 41 seconds before launching the attack; this means that the solution predicted the attack 44 minutes and 08 seconds after the infection started. The COOPRED DDoS evaluations utilized Scenario 10 as a training set. In this case, the first attack preparation signal occurred 46 minutes and 18 seconds before the launch of the attack; this represents 3 minutes and 31 seconds after the start of the infection (see Fig. 4(a)). Therefore, COOPRED DDoS signal engineering increases the prediction time compared to [45].

This work presents and evaluates EWS engineering and the COOPRED DDoS system; however, some limitations guide future work to evolve the DDoS attack predictions. The agents distribute the processing of EWS engineering, and the intelligence center centralizes the analysis of signals. Future work addresses the possibility of agents performing distributed predictions to prevent the intelligence center from being a point of failure. Therefore, network administrators would receive notifications of possible DDoS attacks even with the intelligence center inoperative. Utilizing the solution in real environments can be challenging if it needs labeled data. Therefore, new research aims to reduce the dependency on labeled data with unsupervised machine learning. One limitation of the evaluation is the attributes utilized by the agents. The evaluation is a valid proposal because the preparation and execution of attacks increase the number of packets sent on the network. However, additional evaluations will test other attributes and evaluate the best attributes to perform EWS engineering. Furthermore, future works will analyze the impact of the distributed approach on precision, prediction time, and network data processing. Finally, this work does not address these future works because this work aims to evaluate EWS engineering and the COOPRED DDoS system. Furthermore, these future works are not trivial. Therefore, it is vital to report the evolution of this work and define the improvements for the new versions of the system.

V. CONCLUSION

Attack prediction, unlike attack detection, is a proactive defense mechanism that has been drawing attention in the literature and can assist in network security management. Predicting attacks is essential because threats are constantly evolving, and attack detection is not enough. Prediction aims to identify attack preparation signals and provide administrators more time to stop the attack. Given this goal, this work presented the COOPRED DDoS system. The system utilizes the distribution of network traffic processing, EWS engineering, cooperation between agents, and machine learning. Analysis of network traffic through EWS engineering makes the system independent of prior knowledge of attack signatures or IDS alerts. The processing distribution aims to avoid bottlenecks, and cooperation prevents that processing distribution from becoming inaccurate. The intelligence center ensures cooperation between agents, and machine learning automates the prediction of attack signals in its preparation stage. This work evaluated the system in four different experiments. The results indicate that EWS engineering can highlight the preparation signals, allowing the system to predict attacks in four scenarios. However, future works will guide improvements in the system.

REFERENCES

[1] Adobe, "Adobe digital economy index: COVID-19 report." blog.adobe.com/en/publish/2021/03/15/adobe-digital-economy-index-covid-19-report.html (accessed August 2021), Adobe Inc., 2021.

[2] J. M. Biju, N. Gopal, and A. J. Prakash, "Cyber attacks and its different types," *IRJET*, vol. 6, no. 3, pp. 4849–4852, 2019.

[3] 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.

[4] R. Hummel, C. Hildebrand, H. Modi, C. Conrad, S. B. Roland Dobbins, J. Belanger, G. Sockrider, P. Alcoy, and T. Bienkowski, "Netscout threat intelligence report." www.netscout.com/sites/default/files/2021-04/ThreatReport_2H2020_FINAL_0.pdf (accessed August 2021), Netscout, 2021.

[5] DDoS-Stress, "DDoS-as-aService." https://ddos.services (accessed July 2021), DDoS Stress, 2021.

[6] B. B. Gupta and A. Dahiya, *Distributed Denial of Service (DDoS) Attacks: Classification, Attacks, Challenges, and Countermeasures*. USA: CRC Press, 2021.

[7] R. M. A. Ujjan, Z. Pervez, K. Dahal, W. A. Khan, A. M. Khattak, and B. Hayat, "Entropy based features distribution for Anti-DDoS model in SDN," *Sustainability*, vol. 13, no. 3, pp. 1–27, 2021.

[8] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: a hybrid deep learning approach for DDoS detection and classification," *IEEE Access*, vol. 9, pp. 146 810–146 821, 2021.

[9] Y. Zhou, G. Cheng, Y. Zhao, Z. Chen, and S. Jiang, "Toward proactive and efficient DDoS mitigation in IIoT Systems: A moving target defense approach," *IEEE TII*, vol. 18, no. 4, pp. 2734–2744, 2022.

[10] A. A. Santos, M. Nogueira, and J. M. F. Moura, "A stochastic adaptive model to explore mobile botnet dynamics," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 753–756, 2017.

[11] A. Marrow and G. Stolyarov, "Russia's Yandex says it repelled biggest DDoS attack in history." reuters.com/technology/russias-yandex-says-it-repelled-biggest-ddos-attack-history-2021-09-09 (accessed October 2021), Reuters, News agency company, 2021.

[12] A. Toh, A. Vij, and S. Pasha, "Azure DDoS protection—2021 Q3 and Q4 DDoS attack trends." https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends/ (accessed January 2022), Microsoft, 2022.

[13] B. B. Gupta and O. P. Badve, "Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment," *NCA*, vol. 28, no. 12, pp. 3655–3682, 2017.

[14] I. Cvitić, D. Perakovic, B. B. Gupta, and K.-K. R. Choo, "Boosting-based ddos detection in internet of things systems," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2109–2123, 2022.

[15] A. Mishra, N. Gupta, and B. B. Gupta, "Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller," *Telecommun. Syst.*, vol. 77, no. 1, pp. 47–62, Jan. 2021.

[16] A. Olabelurin, S. Veluru, A. Healing, and M. Rajarajan, "Entropy clustering approach for improving forecasting in DDoS attacks," in *ICNSC*. Taiwan: IEEE, 2015, pp. 315–320.

[17] Z. Wang and Y. Zhang, "DDoS event forecasting using twitter data," in *IJCAI*. Australia: AAAI Press, 2017, p. 4151–4157.

[18] M. Pelloso, A. Vergutz, A. Santos, and M. Nogueira, "A self-adaptable system for DDoS attack prediction based on the metastability theory," in *GLOBECOM*. UAE: IEEE, 2018, pp. 1–6.

[19] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comp. Secur.*, vol. 45, p. 23, 2014.

[20] A. Muhammad, M. Asad, and A. R. Javed, "Robust early stage botnet detection using machine learning," in *ICCWS*. Pakistan: IEEE, 2020, pp. 1–6.

[21] H. Salemi, H. Rostami, S. Talatian-Azad, and M. R. Khosravi, "Leaesn: Predicting DDoS attack in healthcare systems based on lyapunov exponent analysis and echo state neural networks," *MTA*, vol. -, pp. 1–22, 2021.

[22] A. Singh and B. B. Gupta, "Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms," *IJSWIS*, vol. 18, no. 1, pp. 1–43, 2022.

[23] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 4, pp. 2046–2069, 2013.

[24] M. Jog, M. Natu, and S. Shelke, "Distributed and predictive-preventive defense against ddos attacks," in *ICDCN*. India: ACM, 2015, pp. 1–4.

[25] 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.

[26] A. Shahraki, M. Abbasi, and Øystein Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of real adaboost, gentle adaboost and modest adaboost," *Eng. Appl. Artif. Intell.*, vol. 94, p. 103770, 2020.

[27] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. USA: John Wiley & Sons, 2015.

[28] V. Dakos, S. R. Carpenter, W. A. Brock, A. M. Ellison, V. Guttal, A. R. Ives, S. Kéfi, V. Livina, D. A. Seekell, E. H. van Nes, and M. Scheffer, "Methods for detecting early warnings of critical transitions in time series illustrated using simulated ecological data," *PLOS ONE*, vol. 7, no. 7, pp. 1–20, 2012.

[29] M. Scheffer, *Critical Transitions in Nature and Society*. USA: PUP, 2009.

[30] O. Ibitoye, R. A. Khamis, A. Matrawy, and M. O. Shafiq, "The threat of adversarial attacks on machine learning in network security - A survey," *CoRR*, vol. abs/1911.02621, pp. 1–27, 2019.

[31] L. An and S. E. Ahmed, "Improving the performance of kurtosis estimator," *CSDA*, vol. 52, no. 5, pp. 2669–2681, 2008.

[32] H. Oja, "On location, scale, skewness and kurtosis of univariate distributions," *SJS*, vol. 8, no. 3, pp. 154–168, 1981.

[33] P. H. Westfall, "Kurtosis as peakedness, 1905–2014. r.i.p." *The American Statistician*, vol. 68, no. 3, pp. 191–195, 2014.

[34] R. Biggs, S. R. Carpenter, and W. A. Brock, "Turning back from the brink: Detecting an impending regime shift in time to avert it," *PNAS*, vol. 106, no. 3, pp. 826–831, 2009.

[35] J. D. Salas, J. W. Delleur, V. Yevjevich, and W. L. Lane, *Applied modeling of hydrologic time series.* USA: WRP, 1980.

[36] V. Guttal and C. Jayaprakash, "Changing skewness: an early warning signal of regime shifts in ecosystems," *Ecology Letters*, vol. 11, no. 5, pp. 450–460, 2008.

[37] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE TAU*, vol. 15, no. 2, pp. 70–73, 1967.

[38] T. M. Bury, C. T. Bauch, and M. Anand, "Detecting and distinguishing tipping points using spectral early warning signals," *J. R. Soc.*, vol. 17, no. 170, p. 20200482, 2020.

[39] X. H. Cao and Z. Obradovic, "A robust data scaling algorithm for gene expression classification," in *BIBE*. Serbia: IEEE, 2015, pp. 1–4.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *JMLR*, vol. 12, pp. 2825–2830, 2011.

[41] A. Telikani and A. H. Gandomi, "Cost-sensitive stacked auto-encoders for intrusion detection in the internet of things," *Internet of Things*, vol. 14, p. 100122, 2021.

[42] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *ICCST*. India: IEEE, 2019, pp. 1–8.

[43] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *NIPS*. USA: Curran Associates, 2017, pp. 4765–4774.

[44] A. B. de Neira, A. M. Araujo, and M. Nogueira, "Early botnet detection for the internet and the internet of things by autonomous machine learning," in *MSN*. Japan: IEEE, 2020, pp. 516–523.

[45] B. M. Rahal, A. Santos, and M. Nogueira, "A distributed architecture for ddos prediction and bot detection," *IEEE Access*, vol. 8, pp. 159 756–159 772, 2020.

Anderson Bergamini de Neira is a Ph.D. candidate at Federal University of Paraná (UFPR), Brazil. His main research interest includes security in computer networks, especially in solutions that use machine learning to reduce the impacts of DDoS attacks. He is a member of the Center for Computational Security Science research team.

Alex Medeiros de Araujo is M.Sc student at Federal University of Paraná (UFPR). His research interest includes data science to network security, machine learning engineering, and distributed systems. He is a member of the Center for Computational Security sCience research team.

Michele Nogueira is an associate professor in the computer science department of the Federal University of Minas Gerais. She holds a Doctorate degree in Computer Science from the UPMC-Sorbonne University, Laboratoire d'Informatique de Paris VI (LIP6). She was in a Sabbatical Leave at Carnegie Mellon University in 2016-2017. Her research interests include wireless networks, network security, and resilience. She was an Associate Technical Editor for the IEEE Communications Magazine and the Journal of Network and Systems Management.