An Autonomous System for Predicting DDoS Attacks on Local Area Networks and the Internet

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

2023 IEEE Latin-American Conference on Communications (LATINCOM) | 979-8-3503-2687-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/LATINCOM59467.2023.10361878

Abstract—**Distributed denial of service (DDoS) attacks continuously evolve, causing losses and increasing service costs. The high volume and fast scaling make it difficult to defend from them. DDoS attack detection is not sufficient to protect the services from attacks. Thus, it is necessary to design prediction strategies to confront these attacks. When it is possible to identify the preparation for attacks, the time to combat them increases. This paper proposes a self-adaptable system to identify DDoS attack preparation and predict them. The system automatically determines the most appropriate neural network architecture for predicting attacks in different scenarios. The system predicts a DDoS attack with an accuracy of 97.89%, higher than the literature, and the prediction occurs 29 minutes before it starts.**

Index Terms—**Security Management, DDoS attack prediction, Machine Learning, AutoML.**

I. Introduction

Malicious individuals create elaborate attacks by exploiting weaknesses in systems and networks to target their victims. An example is the Distributed Denial of Service (DDoS) attack, one of the most harmful existing cyber threats [1]. The constant evolution and rapid DDoS attack escalation make detection solutions obsolete. In November 2022, a DDoS attack achieved 60 million requests in less than 30 seconds [2]. These attacks can cause losses of up to millions of dollars [3]. Therefore, only detecting DDoS attacks is insufficient to prevent the damages caused by them.

Deep Learning (DL) has been gaining attention in the cybersecurity community because of the diverse architectures capable of adapting to different attacks. However, successfully implementing Machine Learning (ML) algorithms requires significant effort [4]. Selecting a candidate among all DL architectures for the deployment environment is challenging. Also, adjusting the solution parameters for optimal performance takes time [5]. This process is even more complex in environments with class imbalance. Cyber attack scenarios are inherently imbalanced, as they have more attacks at one moment and more legitimate instances at another.

The existing solutions for DDoS attack prediction require proper parameterization. In [6], the authors used an Artificial Neural Network (ANN) to predict DDoS attacks. However, identifying the correct ANN architecture is challenging and requires adjustments based on the context. In addition to the time spent evaluating all architectures, the one found may not be ideal for other scenarios. Hence, in [7], the authors employ Markov chains to predict the next state of the network based on the current state as input to DDoS attack prediction. However, correctly configuring the Markov chain for each scenario is not trivial. This work addresses the problem of how to effectively detect DDoS attacks with minimal human intervention.

This paper proposes a self-adaptive system to predict DDoS attacks. The system is based on DL and Automated ML - AutoML (*i.e.*, a study field that aims to democratize ML use). The AutoML automates a part of the work by selecting techniques and hyperparameters that reduce classification errors [4]. Thus, the system employs AutoML techniques to configure neural networks for DDoS attack prediction in different network models. This saves time and outperforms existing literature in both processing and DDoS attack prediction time.

The proposal was evaluated through two experiments. The first experiment considered the CTU-13 dataset, which represents the local network of the Czech University [8]. In this experiment, the proposal predicted the attack 29 minutes and 51 seconds before its launch, with an accuracy of 97.89%. The second one utilized the DDoS Evaluation Dataset (CIC-DDoS2019) [9], where the victim and attackers are in separate networks connected via the Internet. The proposal anticipated the attack 15 minutes and 21 seconds before the attacker launched it, with an accuracy of 85.61%. The analysis demonstrated that the system autonomously adapts to the datasets and surpasses the accuracy results in the literature [10].

This article contributes to the prediction of DDoS attacks and demonstrates the applicability of the AutoML process. The results show that DDoS attacks can be predicted by autonomously configured models without the dependence on labeled data for algorithm training. The use of labeled data is costly and limits the generalization of the solution, given the abundance of DDoS attacks. The results also highlight the self-adaptation capacity of the system in different scenarios. This work is pioneering in automating DL architecture selection to predict DDoS attacks on local networks and the Internet.

This paper proceeds as follows. Section II presents related works. Section III details the proposal. Section IV discusses 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.

979-8-3503-2687-1/23/$31,00 ©2023 IEEE

II. Related Works

AutoML techniques support cybersecurity solutions adaptation to different types of attacks. In [11], the authors utilized the AutoML process to combat blackhole routing attacks. The proposed solution found different DL architectures for different datasets and achieved an accuracy of 97.91% in classifying normal and malicious traffic to detect blackhole attacks. However, the authors do not provide complementary metrics (*i.e.*, precision and recall) for evaluating whether the proposed solution adequately addresses the problem.

In [12], the authors proposed an AutoML framework to detect DDoS attacks. The framework receives data to evaluate six algorithms. The AutoML selects the algorithm that achieves the highest accuracy. However, accuracy is not ideal when the data classes are imbalanced. Accuracy favors models that correctly classify only normal traffic, failing to identify attack traffic. Moreover, the solution only detects the attack after it is started. The time to stop it may not be enough.

Few works address DDoS attack prediction in the literature. In [13], a solution was proposed to monitor social media texts to identify potential attack alerts. The solution filters posts related to DDoS attacks. However, attackers must signal the attack, such as posting a message containing hate speech against the victim. In [14], the authors manually compared three algorithms for predicting DDoS attacks: Logistic Regression (LGR) and Support Vector Regression. In order to perform the comparison, the authors manually increased the number of packets at certain times to introduce attack traffic. LGR achieved the best results predicting DDoS attacks 15 minutes in advance, with an accuracy of 98.60%.

The literature focuses on predicting attacks by creating solutions based on algorithms that do not adapt to new contexts (*i.e.*, changes in the distribution of analyzed data). The different DDoS attacks cause premature obsolescence of the cybersecurity solutions, as the attackers changes the behaviors pattern for which the model was created and calibrated to detect. Moreover, correctly selecting and configuring the architecture of DL algorithms is a challenge. Thus, this work proposes a DDoS attack prediction approach that autonomously defines and configures the employed neural network architectures.

III. Proposed System

This section describes the proposed system following its five steps as illustrated in Fig. 1. The system uses the autonomously selected and configured DL to predict DDoS attacks: (1) network traffic capture; (2) feature engineering; (3) Autonomous hyperparameter tuning and configuration; (4) attack prediction; and (5) attack notification.

A. Network Traffic Capture

The collected data is input to train the DL model and to predict attacks. The traffic is captured by a tool integrated into a firewall that intercepts inbound and outbound traffic and extracts the headers from transmitted packets. The system may not collect data from a specific protocol to save computational resources. For example, the system ignores the other protocols if the user defines that the system should only use network traffic attributes (Subsection III-B) based on User Datagram Protocol (UDP). Finally, the collection of network traffic is independent of the network topology (Subsection IV-A).

The collected data are exported to Packet Capture (PCAP) files to enable the extraction of network attributes on the packet headers (Subsection III-B). The selected network attributes of the packet headers are collected and then stored on a dedicated server by a tool integrated into the firewall. The system defines the volume of the capture by the number of packets or by the capture duration. Thus, each PCAP file contains a fixed amount of packets or all the packets collected within a time window. The defined window is 50K packets in this work, and the default value for the time is one second. However, these parameters are adjustable by the network administrator. Each PCAP file is saved with the capture date. Then, the system filters the newest ones. The oldest data is removed when the storage limit is reached to avoid server overload. The user sets the limit storage value based on hardware availability. The default action is removing old data when the remaining space is less than 20%.

B. Features Engineering

The feature engineering step starts defining and extracting network traffic attributes. The system monitors the dedicated file server (Step 1), searching for new captures and extracting attributes based on packet headers (*e.g.*, the sum of packet size in bytes and a total of Internet Protocol - IP addresses). The selected network traffic attributes can be customized on the fly. The imperative for the system to work is that the network attributes selected must be affected by attack.

The attacker may perform tests before launching the attack [15]. Thus, the prelude to the attack can impact a few network traffic attributes, such as the number of devices exchanging packets and the number of packets. Command and Control (C&C) communication, *i.e.*, messages sent by the attacker to the BotMaster and from BotMaster to infected devices (bots), can represent the prelude to DDoS attacks. In [16], the authors have identified 40 representative attributes for detecting C&C communications. These attributes can predict attacks since the C&C communication occurs before the attack. This work does not carry out attribute selection, as the DL does not require manual selection.

The system aggregates capture by a time unit (*e.g.*, minutes or seconds) to extract network traffic attributes. Aggregation is required because the system captures the network traffic by packet window (*e.g.*, 50k packets, as in Subsection III-A). Defining the appropriate capture aggregation time unit is essential. Small aggregation time units are unsuitable for predicting the attack due to the limited information available. Large values result in longer processing time, impairing the attack prediction. The default value adopted by the system is one second, which the user can change. Aggregation keeps network traffic attributes ordered sequentially over time, composing a time series for each attribute.

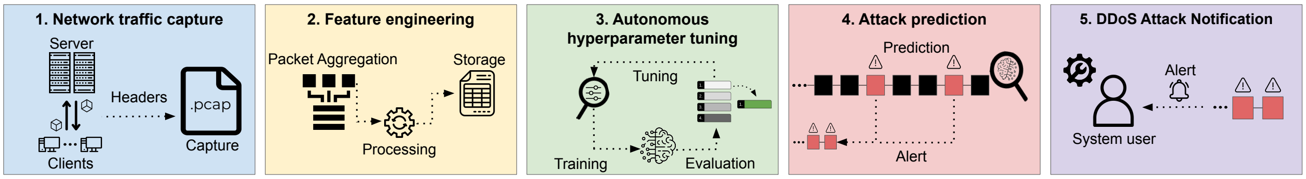


Fig. 1. Proposed system architecture

The system transforms user-defined network traffic attributes (*e.g.*, sum of packet size in bytes and total of IP addresses) into early warning signals. Thus, the statistical measures of Skewness, Kurtosis, and coefficient of variation are applied. These statistical measures are based on the concept of time series. Kurtosis (Eq. 1) is a measure that characterizes the flatness of the curve of a distribution. This metric shows how much a variable is in the distribution tails.

$$Kurt = \frac{(T-1)}{(T-2)(T-3)}(T-1)\hat{y}+6, \tag{1}$$

where

$$\hat{y} = \frac{T\sum(x_t-\mu)^4}{[\sum(x_t-\mu)^2]^2}. \tag{2}$$

The term T is the total number of items observed in the time series, μ is the simple mean of the entire time series, and x_t refers to each observed item in the time series with its index.

The skewness estimates data asymmetry in time series. This metric indicates how much the probability distribution of a random variable deviates from its normal distribution (Eq. 3).

$$Skewness = \frac{T\sum_{t=1}^{T}(x_t-\mu)^3}{(T-1)(T-2)s^3}, \tag{3}$$

where T represents the total amount of observed items. The term x_t refers to each observed item in a time series. The μ refers to the simple arithmetic mean of the set, s represents the standard deviation of the time series.

The Coefficient of Variation (CV) analyzes the dispersion relative to its mean value. The CV makes it possible to compare a series of values with different measurement units. The CV indicates the diversity of the mean of the analyzed datasets. CV formula divides the standard deviation (s) by the mean of the analyzed time series(μ), where $\mu \neq 0$.

Oscillations and perturbations affect the data distribution and generate variation. An example is the critical transition that occurs when the system transits between equilibrium points. The statistical calculation on time series features helps to identify signals that precede critical transitions [17], [18]. Thereby, the system transforms network traffic attributes into signals through Skewness, Kurtosis, and CV to identify critical transitions during the orchestration of attacks and thus predict their launch. In the state transition, it is possible to observe an increase or decrease in the Skewness of a time series.

Similarly, oscillations cause the state of a system to reach extreme values close to a transition, leading to an increase in the Kurtosis of a pre-transition time series. The CV can also be an early warning signal, as increases in the CV point out the occurrence of a critical transition [17]. Thus, the proposal predicts attacks based on network traffic. The future event of attack occurrence is considered when the statistical features vary, forcing the system to present errors in the reconstruction.

C. Autonomous Hyperparameter Tuning

In Step 3, the system autonomously trains a neural network for DDoS attack prediction. There is a vast diversity of ML algorithms and ML architecture, with unique characteristics that adapt to different data volumes. However, selecting and configuring the proper architecture for the problem is challenging. Aiming to democratize the use of machine learning, specialists gathered efforts to develop the AutoML process. AutoML aims to find and configure an ML algorithm to reduce classification errors for the database selected by the user of the framework [5]. Given AutoML automation potential, the system uses this resource to avoid errors in the model selection, configuration, and training process able to predict DDoS attacks without human interaction. AutoML suggests the most suitable algorithm for the network context, determining the most suitable set of ML algorithms or neural network architecture. Thus, pre-processed data are used for training the neural network with layer selection.

The AutoML framework usually follows four stages. Stage 1 defines the search space with a set of neural network architectures [19]. Candidate architectures can vary between different structures (*i.e.*, numbers of hidden layers, weights, and number of neurons). Each AutoML framework defines its search space. In Stage 2, a subset of all candidates ML algorithms is configured. This stage uses some optimization processes for the AutoML framework to evaluate algorithm configuration and architecture combinations. In Stage 3, the AutoML framework trains and tests configured algorithms using the dataset that the framework user selects. Accuracy is a metric for evaluating ML algorithms. However, choosing other metrics such as precision, recall, or F1-score, to privilege more balanced models is expected. In environments with imbalanced classes, these metrics favor models that correctly classify normal traffic and attack preparation traffic, enabling the prediction of DDoS attacks (minority class).

After the Stage 3, the AutoML framework returns to Stage 2, then ML algorithms receive new configurations to maximize the evaluation metrics (results). The cycle between Stage 2 and 3 repeats until the AutoML framework encounters a stopping value (*e.g.*, the execution time or the number of iterations). In Stage 4, the AutoML framework chooses the ML architecture that maximizes the evaluation metric. The proposed system evaluates only neural networks of the Long

Short-Term Memory (LSTM) Autoencoder type to restrict the search space and reduce the time to select and configure the neural network architecture. However, the system is able to use other types of neural networks based on Autoencoders.

LSTM is a neural network that differs from other algorithms because it reduces the problem of gradient dissipation. The backpropagation algorithm is used in LSTM for training neural networks by updating the neuron weights based on the error of the model output. However, over the training time, the backpropagation algorithm dissipates the error value and thus cannot update the layer weights, especially the initial ones. LSTM avoids gradient dissipation through the constant error carousel (CECs). CECs assist in propagating the error to the initial layers by creating a constant error stream.

The autoencoder comprises a neural network composed of input and output layers, a latent space, an encoding neural network, and a decoding neural network. This neural network aims to learn how to encode and decode data the network user selects. Hence, the Autoencoder encode the data in the latent space and decodes the output data of the model. Then, the Autoencoder compares the processed output with the original user data. The error between the predicted and the real is used to update the network weights [20].

This work assumes an LSTM Autoencoder network. Thus, both encoder and decoder are the LSTM network [20]. LSTM network training can be conducted without using labeled data. This feature enables the system application in real network environments by reducing the cost of acquiring data labels. Based on the network data collected, the system identifies changes in the behavior of the signals that correspond to the occurrence of a future DDoS attack. Hence, the goal of Step 4 of the system is to automate the configuration of every aspect of the LSTM Autoencoder architecture to adapt to different DDoS attacks without requiring human interaction.

D. DDoS Attack Prediction

In Step 4, the autonomously configured neural network processes the data from the new network collections and tries reconstructing it. Then, the neural network compresses the signals (*i.e.*, Skewness, Kurtosis, and CV) in the first layers of the LSTM Autoencoder. Next, the neural network decodes these signals to reconstruct them. The system compares the values of Skewness, Kurtosis, and CV over the network traffic with the values of the reconstructed signals. It uses the absolute difference between them to obtain the value of the reconstruction error. The error is calculated for Skewness, Kurtosis, and CV individually. The system calculates the arithmetic mean of the errors to obtain a total error indicator and repeats this process whenever there are new network collections.

The DDoS attack prediction occurs when the total reconstruction error exceeds a previously defined threshold. The system sets the threshold based on the percentile of the training reconstruction error. The default value for the percentile is 97.9%. This means that the threshold is set based on the error value of the smallest sample outside the 97.9 percentile group. This default value was empirically defined based on several tests performed before the evaluation (Section IV). However, the user has an option to modify the percentile for finer control of the predictions. A lower percentile can be applied for more system flexibility and alertness, on the other hand for a tighter system, a higher percentile must be used.

E. DDoS Attack Notification

The system uses the output of Step 4 to issue notifications of the occurrence of possible DDoS attacks (Step 5). A web application programming interface (API) can transfer the system data as input to automate another cybersecurity solution. Besides, whenever the system predicts an attack, it can notify the users via e-mail or messages.

IV. Evaluation

The evaluation of the proposed system follows two experiments. The datasets used in the experiment have labeled the bots and the start of a DDoS attack. This work uses them to check whether the proposed system identifies signals of attack preparation. For the evaluation, the datasets show some communication from the bots or some action taken before the attack, such as infection of the devices or some attack test. Identifying the beginning of the attack allows the use of network traffic from before the start of the attack. This work used Capture 51 of the CTU-13 and CIC-DDoS2019 datasets.

The experiments group the packets from the datasets every second to extract the attributes. This work used one-second intervals to obtain accurate results. For every second, the system collected three network traffic attributes: (i) the number of packets, (ii) the number of source IP addresses, and (iii) the number of destination IP addresses. The number of packets is one of the most relevant attributes identified in [16]. Also, when attackers test attacks, the number of packets varies. In order to define the number of source IP addresses, the system measured how many unique IP addresses sent packets using the source address field of the IP packet. The number of destination IP addresses depends on the count of unique IP addresses in the destination field of the IP packet. These attributes avoid IP address spoofing, a common practice in DDoS attacks [1]. The number of IPs that send packets before the attack has the potential to be an attribute because the preparation of the attack causes variations in this attribute.

The system calculates the values for each early warning signal: kurtosis, skewness, and CV (Subsection III-B) using a fixed-sized sliding window. The system calculates one signal for each attribute of the network traffic for the ML selection to be performed on time. Therefore, the system calculates kurtosis for the number of source IP address attributes, skewness for the number of destination IP addresses, and CV for the total packets. All system results are available online[1]. Combining early warning signals with network traffic attributes was responsible for maximizing DDoS attack prediction results on multiple tests. The system uses the fixed-size sliding window concept to eliminate inaccurate trends and evaluate

[1] github.com/daviembrito/pred-ddos-automl

the proposed solution over time [18]. The literature is not unanimous about this value. For example, [18] used 40% of the dataset for the window size. This work employs 5% to maximize the DDoS prediction time. The prediction is delayed when using a high window size value since the system spends more time to conduct the analyses.

After computing the signals, the system used Autokeras[2] to identify the best LSTM neural network Autoencoder for each dataset (*i.e.*, for the two experiments). Autokeras uses Bayesian optimization to find the best neural network architecture based on input data. Autokeras implements the steps presented in Subsection III-C through a tree-structured acquisition function optimization algorithm and a neural network kernel. The goal is that Autokeras analyzes the search space efficiently and finds the best architecture for different cases.

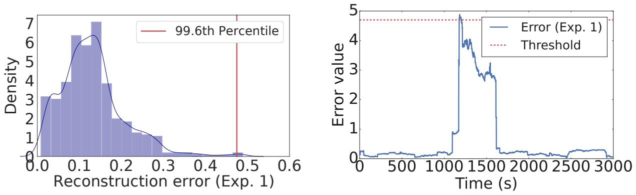
The metrics accuracy, precision, and recall evaluate the system performance. These metrics require the following values: (i) total true positives (TP); (ii) total false positives (FP); (iii) total false negatives (FN); (iv) total true negatives (TN) and (v) total of samples (N). Accuracy evaluates the system classifications (Eq. 4). As there are few signals of preparedness for attack as attackers hide their actions, it is necessary to supplement analysis with accuracy and recall. Precision indicates the relationship between samples labeled by the system for a specific type and how many are of the assumed type (Eq. 5). The recall presents the relationship between all expected samples of the specific type and how many samples of that system are correctly classified (Eq. 6). Since it is possible to get precision and recall for the positive and negative classes, and the number of samples varies a lot because of the data imbalance in a DDoS attack, the average precision and recall weighted by the number of samples for each class is considered.

$$a = \frac{TP + TN}{N} \quad (4) \qquad p = \frac{TP}{TP + FP} \quad (5) \qquad r = \frac{TP}{TP + FN} \quad (6)$$

A. Experiments

Experiment 1 has employed a network traffic collected from a local network university and made available on the CTU-13 dataset. This capture is 8803 seconds long, with ten bots, 41 GB, 46,997,342 packets, and Internet Control Message Protocol (ICMP) and UDP flood-type attacks. In order to create the dataset, the researchers have captured real data from the university, infecting the bots in the second 2643 and launching the attacks in the second 5632 of the capture. The system has employed 30% of the total network traffic to run Autokeras and configure the LSTM Autoencoder network. This traffic contains only pre-attack traffic (*i.e.*, from second 0 to second 2642). The test set has used the remaining traffic before the attack started (*i.e.*, from the second 2643 to the second 5631). In order to demonstrate the customization power of the system, Experiment 1 uses a high percentile set at 99.6%. Therefore, this work set the threshold to the smallest sample error value out of the 99.6% of the smallest samples.

Fig. 2(a) shows the histogram of the error in the training set. Following the approach defined in Subsection III-D, the threshold defined for this experiment is 0.47. However, it is worth mentioning that the user can determine how the system will issue the alert. The threshold has been defined multiplying it by 10 to show the customization ability. This results in a severe and more customized threshold than the default threshold of the system. The system has misclassified only 63 of the 2990 seconds of the test set using the threshold equal to 4.7 (red dotted line in Fig. 2(b)). This guarantees an accuracy of 97.89%, a weighted average accuracy of 97.4%, and a weighted average recall of 97.9% for identifying seconds with more than two packets sent by bots (Table I). The proposed system has predicted the DDoS attack 29 minutes and 51 seconds before its start.



(a) Error in training set

(b) Error in test set

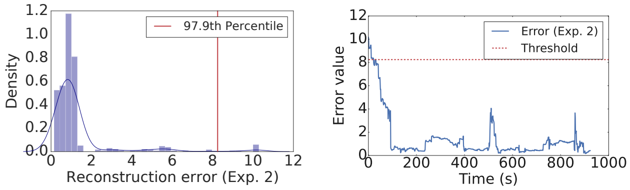
Fig. 2. Results in the Experiment 1

Experiment 2 has employed the CIC-DDoS2019 dataset, where the Internet connects the victim to the attacker. This capture has 19 DDoS attacks launched by the researchers in two days. The dataset has 61,407,883 packets, and 27 GB of data referring to attacks and real data. System evaluation has focused on predicting the first DDoS attack performed. The attack starts at the second 1484 of capture and lasts 540 seconds. The experiment has used 28% of the total capture traffic to perform model training (*i.e.*, from second 0 to second 559) and the remainder of pre-attack traffic for the test (*i.e.*, from second 560 to second 1483). Fig. 3(a) shows the histogram of the error in the training set. Experiment 2 uses the default threshold setting defined in Subsection III-D (97.9% percentile). In this case, the threshold value is 8.3; this implies a lower and less customized threshold than the threshold value used in Experiment 1. Using this threshold (red dotted line in Fig. 3(b)), the system has predicted the DDoS attack 15 minutes and 21 seconds in advance, correctly classifying 791 out of 924 seconds with more than two packets sent by bots. Accuracy was 85.39%, average precision was 78.7%, and average recall was 85.4% (Table I).

B. Discussion

The proposed system has predicted the DDoS attack with an accuracy greater than 85%. The solution notifies the DDoS attack occurrence minutes before the attacker launches it, increasing the time to proceed with the necessary countermeasures to avoid losses caused by attacks. The results show that the system autonomously selects and configures the best LSTM Autoencoder neural network architecture for each

[2] https://autokeras.com/



(a) Error in training set (b) Error in test set

Fig. 3. Results in the Experiment 2

experiment performed. This autonomy reduces the time spent in the manual neural network configuration process. Even with the significant imbalance of classes with more normal traffic than attack traffic, the system has obtained metrics that surpassed works in the literature.

The proposed system brings four advances to the literature. The first is the increase in the prediction time compared to the literature. The proposal has predicted the attack 29 minutes and 51 seconds in advance in CTU-13 capture 51. The work of [21] has predicted the same attack only 5 minutes and 41 seconds in advance. The second one is the reduction of prediction errors. The work of [10] has misclassified 257 seconds, and the work of [22] misclassified 77 seconds of the network traffic from capture 51 of the CTU-13 dataset. While this work misclassified 63 seconds in the same context (Table I). The error reduction is related to the autonomous configuration of the LSTM Autoencoder performed by the proposed system. A specialized configuration for each context allows the system to extract the best from ML.

TABLE I
RESULTS AND COMPARISON WITH THE LITERATURE

Experiments	VP	FP	FN	VN	Accuracy	Total error
Experiment 1	2926	24	39	1	**97.89%**	**63**
Experiment 2	783	18	117	6	85.39%	133
[10]	2737	1	256	2	91.42%	257
[22]	413	32	45	10	84.60%	77

The third advance is related to adaptation. The system is adaptable to different scenarios, and the results support this generalization. The system predicted attacks on different datasets. All datasets have distinct characteristics, such as attack types, botnets, topology, and size, differently from the previous works that focused mainly on Capture 51 of the CTU-13 [10] dataset. Finally, all these advances are independent of labeled data. Thus, the results are not limited to a botnet family, specific DDoS attacks, or a specific statistical profile.

Future work will focus on optimizing time and resource consumption. The system used 42 and 28 minutes to prepare the LSTM Autoencoder neural networks in Experiments 1 and 2. The literature indicates that this process can take longer, between 3 and 24 hours [19]. The proposal consumes less time than the literature because it uses few features.

V. Conclusion

This paper presented a pioneering system for automating the prediction of DDoS attacks on local networks and the Internet. Using AutoML, the system configures one LSTM neural network autoencoder architecture for different scenarios, providing attack prediction without human interaction. Further, the system does not use labeled data to predict the DDoS attack. The results indicated that the system can predict DDoS attacks 29 minutes and 51 seconds before launch, with an accuracy of 97.89%. Future works will cover other neural networks and their performance.

References

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

[2] R. Lakshmanan, "Massive HTTP DDoS attack hits record high of 71 million requests/second," https://thehackernews.com/2023/02/massive-http-ddos-attack-hits-record.html, 2023.

[3] Nokia, "Ddos security," https://www.nokia.com/networks/security/ddos-security/, Nokia Corporation, 2023.

[4] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "DoS and donts of machine learning in computer security," 2020.

[5] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *NIPS*. USA: MIT Press, 2015, p. 2755–2763.

[6] S. Anuar, N. A. Ahmad, S. Sahibuddin, A. Ariffin, A. Saupi, N. A. Zamani, Y. Jeffry, and F. Efendy, "Modeling malware prediction using artificial neural network," in *SOMET*, vol. 303. IOS Press, 2018, pp. 240–248.

[7] M. Q. Ali and E. Al-Shaer, "Configuration-based IDS for advanced metering infrastructure," in *SIGSAC*. USA: ACM, 2013, p. 451–462.

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

[9] 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.

[10] G. L. F. M. E Silva, A. Neira, and M. Nogueira, "A deep learning-based system for DDoS attack anticipation," in *LATINCOM*, 2022, pp. 1–6.

[11] P. Ioulianou, V. Vasilakis, and S. F. Shahandashti, "ML-based detection of blackhole and rank attacks in RPL networks," PRT, 2022.

[12] E. Horsanali, Y. Yigit, G. Secinti, A. Karameseoglu, and B. Canberk, "Network-aware AutoML framework for software-defined sensor networks," in *DCOSS*. IEEE, 2021, pp. 451–457.

[13] L. A. F. Santos, R. Campiolo, M. A. Gerosa, and D. M. Batista, "Extração de alertas de segurança postados em mensagens de redes sociais." in *SBRC*, Brasil, 2013, pp. 791–804.

[14] P. Machaka, O. Ajayi, H. Maluleke, F. Kahenga, A. Bagula, and K. Kyamakya, "Modelling DDoS attacks in IoT networks using machine learning," 2021.

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

[16] 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.

[17] S. R. Carpenter and W. A. Brock, "Rising variance: a leading indicator of ecological transition," *Ecology Letters*, vol. 9, no. 3, p. 8, 2006.

[18] 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, 2020.

[19] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5G networks," *arXiv*, vol. -, no. -, p. 12, 2020.

[20] H. Nguyen, K. Tran, S. Thomassey, and M. Hamad, "Forecasting and anomaly detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management," *IJIM*, vol. 57, pp. 1–38, 2021.

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

[22] A. Neira, L. Borges, A. Araújo, and M. Nogueira, "Engenharia de sinais precoces de alerta para a predição de ataques ddos," in *WGRS 2023*. Brasil: SBC, 2023, pp. 139–152.