Vol 26 (3)

Ingeniería y
Competitividad
Vol 26 (3) 2024; doi: 10.25100/iyc.v26i3.13800

ISSN
0123-3033
e- 2027-8284

Original Article

Validation of the Intelligence Technique in the detection of cyber attacks

Validación de la Técnica de Inteligencia en la detección de ciberataques

Santiago Ordoñez Tumbo[1] Katerine Márceles Villalba[2] Siler Amador Donado[3]

[1]Institución Universitaria Colegio Mayor del Cauca, Faculty of Engineering, Computing Engineering Program, I+D in Computing Group, Popayán-Colombia.
[2]Universidad de Antioquía, Faculty of Engineering, System Engineering Program, In2lab Group, Medellín-Colombia.
[3]Universidad del Cauca, Faculty of Electronic Engineering and Telecommunicatios, System Engineering Program, Information Technology Research and Development Group (GTI), Popayán-Colombia.

Abstract

This article presents the process carried out to evaluate the most suitable intelligence technique for the identification of malicious traffic in order to minimize the risk of a cyberattack. This was accomplished through four phases using an action research methodology articulated to a systematic literature review, and through proposed scenarios that allowed for the validation of this approach.

Resumen

Este artículo presenta el proceso realizado para la evaluación de la técnica de inteligencia más adecuada que permita la identificación de tráfico malicioso con el fin de minimizar el riesgo a un ciberataque. Esto fue realizado a través de cuatro fases empleando la metodología de investigación-acción articulada a una revisión sistemática de literatura y a través de escenarios propuestos permitieron validar ésta.

Keywords: cyberattack detection, Intelligence technique. engineering.

Palabras clave: detección de ciberataque, ciberataque, inteligencia técnica, ingeniería.

How to cite?

Ordoñez, S., Márceles, K., Amador, S. Validation of the Intelligence Technique in the detection of cyber attacks. Ingeniería y Competitividad, 2024, 26(3)e-20213800

https://doi.org/10.25100/iyc.v26i3.13800

Recibido: 26-05-24
Aceptado: 12-08-24

Correspondence:
Katerine.marceles@udea.edu.co

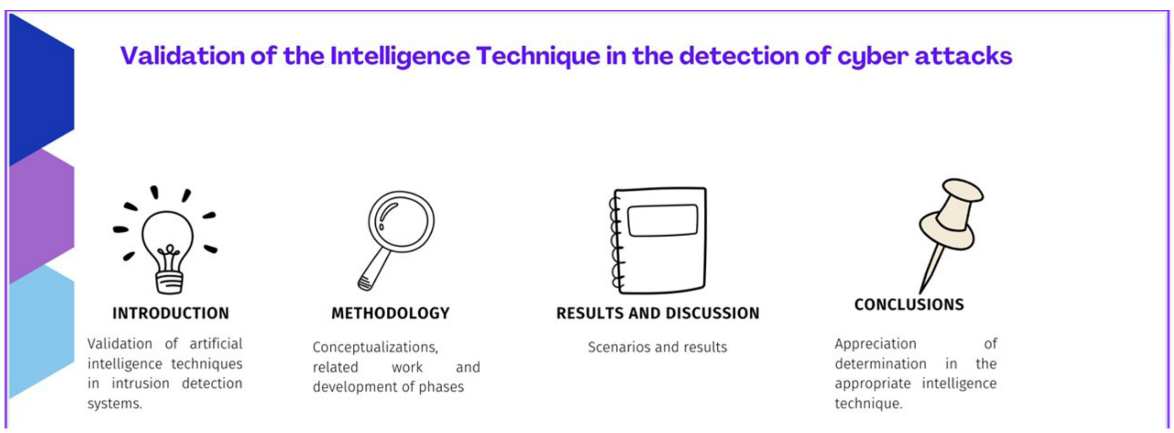
Conflict of interest: none declared

Why was it conducted?:
The research was motivated by the need to develop a cybersecurity framework that integrated artificial intelligence to effectively respond to cyberattacks. This requirement arose from the increasing sophistication and frequency of digital threats. To address this, multiple scenarios were designed and evaluated, employing various artificial intelligence techniques, with the goal of optimizing threat detection and reducing errors in threat identification.

What were the most relevant results?
The most significant results emerged after multiple tests based on simulated attack scenarios. The most effective technique was identified, which not only improves the detection of cyberattacks but also minimizes the incidence of false positives. This technique proved to be superior compared to traditional methods and other new approaches evaluated during the research.

What do these results contribute?
The results of this study are particularly valuable because they offer a viable and efficient solution for medium and small-sized companies, providing a crucial tool for network administrators in anomaly detection. This advancement represents significant support in enhancing cybersecurity, allowing for more precise and proactive risk management.

Graphical Abstract



doi: 10.25100/iyc.v26i1.13800

Introduction

Currently, there are various ways to maintain the security of our machines or networks. The digital age has seen a tremendous impact due to innovation and new technologies that enable organizations to carry out their activities more efficiently, largely thanks to the adoption of the Internet of Things (IoT). However, the constant incorporation of these technologies also requires a robust approach to information assurance. Considering that the data stored on these devices are critical assets, it is essential to recognize that cyberattacks evolve daily, making system security an ongoing challenge.

On the other hand, organizations implement security measures to mitigate risks, but often without considering that attack methods are constantly evolving, which could pose a significant problem for those responsible for system security. In this context, Collaborative Intrusion Detection Systems (CIDS) are used, which allow information sharing between detectors to expand the knowledge base and improve response capabilities to potential threats. However, one of the main challenges of CIDS lies in their ability to adapt to new threats. Updating rules often requires manual intervention, which can be a daunting task for organizations needing to maintain an up-to-date knowledge base and process large volumes of data to anticipate new attacks, although this is never 100% effective.

This is where artificial intelligence becomes crucial, as it offers the possibility of generating anomaly detections intelligently and based on real data, achieving a 95% accuracy rate in detecting abnormal behaviors in devices connected to an organization's network.

For the development of this research, it was essential to follow a series of structured steps. The research-action (R-A) methodology was employed in this project, allowing results aligned with the need for validating the artificial intelligence technique suitable for adaptation to an IDS (Intrusion Detection System).

Methodology

Before starting with the methodological development, it is important to consider some concepts and background information that were important references for the development of this work:

Cybersecurity: Cybersecurity safeguards computing equipment, mobile devices, electronic services, and data networks from malicious attacks (1). Electronic devices are characterized by inter-relating with each other, either directly or indirectly, through data networks or external storage devices.

Intrusion Detection System (IDS): An IDS is a device or software application that monitors the network for malicious activity (2). The intrusion detection system continuously analyzes the traffic passing through the data network to identify anomalies based on patterns and heuristics.

IoT: Refers to devices being networked to identify, monitor, and control the physical world (3). The function of IoT is to interconnect and transfer data.

Netflow: It is a network protocol developed by Cisco Systems (4) that collects specific information about network traffic through IP addresses and allows selecting only certain packets, turning them into a dataset containing a series of information fields.

Deep Learning: Currently, artificial intelligence has gained significant momentum and is being applied in many fields of computer science, one of its fundamental components being deep learning. Deep learning can be defined as a class of machine learning algorithms (2). The main idea of deep learning is to solve problems using deep neural networks that seek to mimic the way the brain makes decisions. In this case, neural networks have a large number of hidden layers compared to traditional neural networks, and this technology aims to obtain simple patterns or features from complex inputs.

doi: 10.25100/iyc.v26i1.13800

Threat: Characterized by being an unwanted incident that can harm a system or an organization (5) by exploiting a vulnerability to attack the security of an information system.

Vulnerability: A weakness or flaw in an information system that can be exploited by a threat (5), thus jeopardizing the security of information in terms of its integrity, availability, or confidentiality.

Industrial Internet of Things (IIoT): With the emergence of the Internet of Things, the industry identified that this technology could be leveraged in its operations. IIoT integrates computing, networks, and physical objects for the industry, where devices are networked to detect, monitor, and control the physical world (6).

Supervised Learning: Characterized by requiring initial labels. These labels refer to the final value of a data sequence that already has its target value (7), allowing the algorithm to learn from its errors and successes based on these previously labeled results, generally used when a numerical or categorical result is requested.

Unsupervised Learning: Its learning is based on unlabeled data, and its experience depends almost entirely on the clustering of data called "clusters," which allow the learning process to group enough data so that in new iterations, it better understands the training data. These clustering methods are divided into two branches: hierarchical, which is based on the hierarchical score set by the model, and non-hierarchical, which are generated by any type of flow.

Semi-Supervised Learning: To talk about semi-supervised algorithms, one must first understand the structures of supervised and unsupervised learning, as this learning uses parts of both (8). For data recognition, labeled data from supervised learning is used, and for final decision-making and learning, an unsupervised system based on clusters generated by unsupervised learning is utilized.

Among the references that were taken into account are:

Internet of Things: A survey on machine learning-based intrusion detection approaches (9). This research focuses on rigorous and cutting-edge investigations on the topics of machine learning applied to the Internet of Things and intrusion detection for network security. The objective of the work is to provide a recent and in-depth investigation of relevant works that address various intelligent techniques and their intrusion detection architectures applied to computer networks, with an emphasis on the Internet of Things and machine learning. This article contributes to the work by providing rigorous results on deep learning and the most suitable techniques.

Detecting Internet of Things attacks using distributed deep learning (8). In this document, a cloud-based distributed deep learning framework is proposed for the detection and mitigation of phishing and botnet attacks. The model comprises two key security mechanisms that operate cooperatively: (1) a distributed convolutional neural network (DCNN) model integrated as a micro security complement for IoT devices, designed to detect phishing and DDoS attacks at the application layer; and (2) a long-short-term memory network (LSTM) hosted in the back-end cloud to detect botnet attacks and ingest CNN (convolutional neural network) embeddings to detect phishing attacks distributed across multiple IoT devices. The distributed CNN model, integrated into a machine learning engine on the client's IoT device, enables the detection and defense of the IoT device against phishing attacks at the point of origin. A dataset consisting of phishing and non-phishing URLs is created to train the complementary CNN security models, and the N_BaIoT dataset is selected to train the back-end LSTM model. This article provides a wealth of information on neural networks and their implementation. Additionally, the model implemented with two neural networks is noteworthy, as it will be very useful to demonstrate their functioning, offering another perspective to consider before proceeding with the implementation.

Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things (10). This paper presents an intrusion detection system based on distributed machine learning in the Internet of Things (IoT) that uses Blockchain technology. Specifically, spectral partitioning is proposed to divide the IoT network into autonomous systems (AS) that allow traffic monitoring for intrusion detection (ID) by the border area nodes of the AS, selected in a distributed

doi: 10.25100/iyc.v26i1.13800

manner. The identification system is based on machine learning, in which a machine algorithm is trained on the support vector using prominent IoT datasets to detect attackers. Additionally, the integrity of the attacker list is provided using Blockchain technology, which enables the distribution of attacker information to the nodes.

The contribution to the present proposal is the specification of presented vulnerabilities and the use of blockchain in IoT technologies, as well as the method of training the machine learning algorithm.

For the construction of any proposal, a series of steps must be followed. In this sense, for the development of this project, the research-action (R-A) methodology (11) is used. This methodology combines theory with practice in such a way that the researcher can draw accurate conclusions about the practices performed. Since this type of methodology aims to solve specific problems by continuously understanding and interpreting them to improve, the phases are described as follows:

Phase 1: Selection of the Intelligence Technique.

Initially, a review was conducted of possible Deep Learning techniques that meet the requirements for detecting cyberattacks and/or anomalies in network traffic. This review considered the articles selected as primary sources after a characterization, resulting in an initial selection of approximately 100 articles in Phase 1. This approach provides insight into which characteristics should be considered for subsequent selection.

Table 1 shows the list of articles that include techniques related to artificial intelligence and were reviewed in detail to obtain the technologies used or referenced in the construction of each one.

Table 1. List of Machine Learning Articles

#	Publication Year	Title of the Article or Study
1	2020	CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques (12)
2	2018	Machine Learning Based Mobile Malware Detection Using Highly Imbalanced Network Traffic (13)
3	2019	EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques (14)
4	2019	Security Analysis of Network Traffic Using Data Mining (15)
5	2020	Probabilistic Modeling Based on Deep Learning for Anomaly Detection in Network Traffic (16)
6	2018	Detecting and classifying malicious TLS network traffic using machine learning (17)
7	2019	Indicators for Ransomware Attack Detection (18)
8	2019	Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection (19)
9	2020	Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers (20)
10	2020	Machine Learning for Traffic Analysis: A Review (21)
11	2019	Machine Learning Algorithms on Botnet Traffic: Ensemble and Simple Algorithms (22)
12	2020	Cyber forensics framework for big data analytics in IoT environment using machine learning (23)
13	2019	Android Malware Identification Based on Traffic Analysis (24)
14	2020	Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city (25)

Source: own by the authors.

doi: 10.25100/iyc.v26i1.13800

Subsequently, each of the articles listed in Table 1 was reviewed, from which the techniques described in the following table were obtained.

Table 2. Description of Techniques

Techniques	Description
Bilstm- bidirectional	It is a bidirectional short-term and long-term memory algorithm, which presents an extended version of RNNs (Recurrent Neural Networks). It is characterized by its inability to learn contextual information over an extended period due to the vanishing gradient problem. LSTMs (Long Short-Term Memory networks) use gates related to units, which address the vanishing gradient problem; thus, they preserve information for longer periods, enabling it to be retained for analysis (26).
Swarm(pso)	Particle Swarm Optimization (PSO) is an Evolutionary Computation (EC) algorithm that can be used to solve optimization problems without domain knowledge (27).
Deep learning of mlp	The Multilayer Perceptron (MLP) algorithm is often an extension of the logistic regression algorithm (28).
Restricted boltzmann machines (rbm)	It is defined as a neural network that places a probability distribution over its input set within Deep Learning. Restricted Boltzmann Machines (RBM) are related to Boltzmann Machines, where the restriction of neurons forms a bipartite graph, with nodes commonly referred to as visible and hidden, respectively (29).
Multilayer perceptron (mlp)	The algorithm consists of multiple layers, providing a good alternative for solving problems that are not linearly separable, which is a limitation when applying Deep Learning. This is why it is referred to as a simple perceptron (30).

Source: own by the authors.

After identifying the different learning techniques, a comparison was made considering the basic criteria of deep learning. To determine the deep learning technique that best adapts to identifying malicious traffic or any anomalies, characteristics were established to score them with quantitative values. For this reason, the selection criteria present in Table 3 were used.

Table 3. Selection Criteria for Deep Learning Techniques

Description		Characteristic	Qualitative Value	Quantitative Value
1	Works under data prediction	Prediction	Yes	1
			No	0
2	Accuracy level in relation to final training	Precision	High	1
			Medium	0.5
			Low	0
3	Permite identificar si la tecnología es de código libre	Open source	Si	1
			No	0
4	Allows determining if the technology is open-source	Learning	Si	1
			No	0
5	The ability to produce results with previously unseen situations	Trend recognition	High	1
			Medium	0.5
			Low	0

Source: Own by the authors.

doi: 10.25100/iyc.v26i1.13800

Based on the above, Table 4 was created, specifying the essential characteristics to consider in order to select the most suitable technique.

Table 4. Deep Learning Techniques with Qualitative Values.

Deep Learning Technique	Prediction	Precision	Open source	Learning	Trend Recognition
Bidirectional Long Short-Term Memory (BiLSTM) Deep Learning	Yes	High	Yes	Yes	High
Deep Neural Network (DNN)	Yes	High	Yes	Yes	High
Swarm(PSO)	Yes	High	Yes	No	High
MLP Deep Learning	Yes	Medium	Yes	Yes	Medium
Restricted Boltzmann Machines (RBM)	No	Medium	Yes	Yes	Medium
Multilayer Perceptron (MLP)	Yes	Low	No	No	Medium

Source: Own by the authors.

In Table 4, the qualitative characterization of deep learning techniques (8) is shown, which will provide a more focused position based on the results obtained for each characteristic.

Table 5. Deep Learning Techniques with Quantitative Values.

Deep Learning Technique	Prediction	Precision	Open source	Learning	Trend Recognition	Average
Bidirectional Long Short-Term Memory (BiLSTM) Deep Learning	1	1	1	1	1	1
Deep Neural Network (DNN)	1	1	1	1	1	1
Swarm(PSO)	1	1	1	0	1	0.8
MLP Deep Learning	1	0.5	1	1	0.5	0.8
Restricted Boltzmann Machines (RBM)	0	0.5	1	1	0.5	0.6
Multilayer Perceptron (MLP)	1	1	0	0	0.5	0.5

Source: Own by the authors.

Based on the results obtained in Table 5, the classification and the highest-rated techniques can be observed, allowing us to deduce which ones are most suitable for the needs of this proposal.

doi: 10.25100/iyc.v26i1.13800

Upon completing this categorization of results based on the deep learning techniques associated with this process, it was concluded that the Deep Neural Network (DNN) (31) technique is the most suitable, considering its high accuracy in response and its solution to the gradient problem, which could present a future inconvenience. This issue is addressed with access to these nodes. Its good performance and low resource cost make it the definitive choice for the appropriate deep learning technique.

After conducting the research and considering the rapid technological advancements and integration of new proposals, the implementation of the deep learning algorithm was initiated. However, several issues arose, including not only the high hardware resource consumption, which is common in such technologies but also the high costs associated with the pre- and post-processing of data. This is due to the fact that the primary data source would quickly become saturated.

The characterization is specific to the final identification process, taking into account characteristics such as being a semi-supervised algorithm. This approach facilitates the process by dividing the analysis into two phases: supervised for training and anomaly identification, and unsupervised for creating clusters to achieve better data management (32).

Phase 2: Dataset Characterization and Construction.

Each context in a project has a specific purpose for directing its artificial intelligence algorithm, considering certain characteristics and needs that should be prioritized. In this case, the priority is the accuracy in analyzing traffic flow to detect potential attacks. After this phase, it is important to understand the data for initial training, as variables of interest need to be defined and subsequently divided into training and test datasets. Therefore, it is essential to have a good input dataset that meets all the aforementioned requirements.

It is important to mention that for the algorithm implementation, a good dataset is required to achieve a precision greater than 90%. This can only be achieved by finding a good balance between benign and malicious traffic. For optimization purposes, it is suggested that the training dataset contain at least 20% attacks and 80% normal or benign traffic, as proposed in (33). This approach ensures that neurons can more easily learn to identify benign traffic, leaving new and unidentified data as anomalous to be compared by unsupervised clusters. In a supervised manner, the algorithm learns to differentiate between benign and malicious data to identify necessary patterns for creating sufficiently optimal neurons for decision-making.

The dataset consists of stored and tabulated data, where each column represents a variable and the rows contain data that identify the same information. This part is crucial for the proper implementation of the learning algorithm because a complete dataset with well-defined data makes the normalization process simpler. This enables the algorithm to learn more optimally and achieve the best results in subsequent tests.

The type of traffic selected for generating input data for both the learning model and the prediction model was netflow, which uses UDP or SCTP-based data sent to a data collector server. In this case, a Raspberry Pi 3 was used to collect the maximum number of netflow packets over a specified period, so that the IDS Suricata could obtain information from these packets. Once the information was collected, it was sent to the proposed algorithm for traffic analysis and to generate a prediction to identify if the traffic is anomalous.

The dataset used for the training phase was a netflow traffic dataset organized based on a test dataset retrieved from the Queensland repository (34) in Australia, which contains various versions of datasets. Subsequently, the dataset was adapted to the test requirements, including both labeled malicious and benign traffic. It is also important to note that the required information was numeric, so a script was generated within the learning model to achieve this transformation and normalization with the training dataset.

doi: 10.25100/iyc.v26i1.13800

For data normalization, certain rules must be considered, such as ensuring that variables are numeric or of type float. This is essential to prevent issues when creating the learning model and to ensure that the model runs smoothly. The dataset contains a total of one million data entries distributed across 12 columns, which include both anomalous and benign data. These will serve as the initial training data, allowing the algorithm to learn to identify potential attacks. These attacks will be transformed into new learning variables and inputs for the training dataset. Likewise, the implemented neurons will be able to identify these new cases, which is the expected outcome from a Deep Learning algorithm.

Phase 3: Evaluation Scenario of Intelligence Technique for Cyberattack Detection.

In this activity, some controlled scenarios are proposed, where the main idea is to test the functionality of the model and provide feedback to the Deep Learning algorithm.

This section addresses how the neural network achieves the expected results based on the training dataset and all the structure and changes adopted throughout the research. To determine the accuracy of the algorithm, it was evaluated using a confusion matrix, which is useful for model evaluation. In this case, it operates based on the principles of neural networks, which require input data divided into training and testing, thus using the model test to determine the variables to be considered by the confusion matrix. Figure 1 shows a representation of the 2*2 matrix that will be used for analyzing the algorithm's performance..

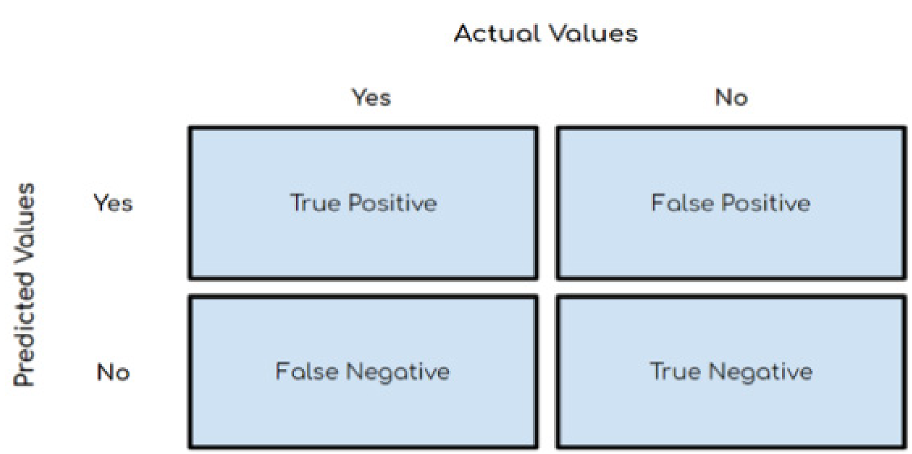


Figure 1. Confusion Matrix. Source: Samhain Labs | samhain. Samhain Labs. https://www.la-samhna.de/samhain/ (accessed on May 15, 2022)

Considering Figure 1, you can see how the accuracy of the algorithm's predictions will be calculated. Similarly, Figure 2 contains the formula that allows for determining accuracy mathematically, where the sum of correct predictions is divided by the total predictions.

$$Accuracy = \frac{\#\ of\ correct\ predictions}{total\ \#\ of\ predictions} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure. 2. Precision Formula. Source: Samhain Labs | samhain. Samhain Labs. https://www.

doi: 10.25100/iyc.v26i1.13800

la-samhna.de/samhain/ (accessed on May 15, 2022)

Based on the formula described in Figure 2, it is adapted to the results of the prediction process, as shown in Figure 3, where a total of favorable hits is observed, resulting in a 95.44% effectiveness rate.

```
In [39]: v=pd.Series(v)
    ...: confusion_matrix(Y,y)
Out[39]:
array([[413511,      0],
       [     0,  20495]], dtype=int64)

In [40]:
```

Figure. 3. Confusion Matrix of the Algorithm. Source: Own by the authors.

Results and discussion

Considering the results shown in Figure 3, it is important to clarify the meaning of the variables used in the confusion matrix, which are as follows:

Y: These are the labeled data that the matrix will compare with the algorithm's prediction results.

y: These are the prediction results or outcomes from the algorithm and its prediction model.

From this, it can be deduced that there were a total of 413,511 true positives, 0 false positives, 0 false negatives, and 20,495 true negatives.

Similarly, the results shown in Figure 4 are produced by the model, which, as mentioned earlier, performs the prediction on a set of data already prepared for the prediction tests, resulting in an accuracy of 95.44% in correct predictions. It can be deduced that the algorithm performed well with the training data and that the trained neurons are ready to be tested with real traffic.

```
Epoch 5/5
13563/13563 [==============================] - 17s 1ms/step - loss: 0.0393 -
binary_accuracy: 0.9549
13563/13563 [==============================] - 12s 846us/step - loss: 0.0393 -
binary_accuracy: 0.9544

binary_accuracy: 95.44%
```

Figure. 4. Algorithm Accuracy. Source: Own by the authors.

This provides a perspective on the algorithm's performance, which, due to its good initial training and neural network algorithm structure, could further improve this percentage as new traffic and more attacks are introduced, making its detection arsenal even more robust. Consequently, the detection time of the algorithm is evidenced through three test scenarios, where the algorithm detected all three attacks during its execution time. See Table 6.

doi: 10.25100/iyc.v26i1.13800

Table 6. Attack Detection Results by Machine Learning Algorithm

Scenario	Attack	Process	Detected	Execution Time
1	Backdoor	Successful	Yes	3 Min
2	DDOS	Successful	Yes	8 Min
3	ARP poisoning	Successful	Yes	5 Min

Source: Own by the authors.

Here are the scenarios that were designed for testing the algorithm's detection:

Scenario 1. In this scenario, a Backdoor attack generates a large amount of malicious and anomalous traffic, where the algorithm was challenged to identify this malicious connection.

Backdoor. When referring to a Backdoor attack, it can be concluded that it is a type of cyber attack designed to provide remote access to the attacker. For this scenario, a direct connection was set up between the attacker and the victim, with the attacker using a virtual machine running Kali Linux and the victim using a virtual machine with Windows 7. It is important to mention that the scenario was not conducted with Windows 10 or 11 because most small and micro businesses use machines with limited resources for their daily activities, and few update their operating systems due to performance concerns. Therefore, Windows 7 was selected for this scenario.

Scenario 2. In this scenario, a DDOS attack was carried out, which, like the Backdoor attack, generated a large amount of malicious and anomalous traffic. This posed a challenge for the algorithm, as it had to identify anomalies caused by service denial. To address this, the algorithm was trained in advance, as with some other attacks.

A DDOS attack (Distributed Denial of Service) is essentially a way to disrupt services on servers, websites, etc., by overwhelming a specific IP address with an excessive number of requests. This causes servers to be unable to process the requests, resulting in errors and restarts. Only one victim is needed for this attack. For this scenario, the ports of a virtual machine were considered, which was prepped for the attack. The algorithm successfully identified the denial-of-service attack patterns.

Scenario 3. For this scenario, an ARP (Address Resolution Protocol) poisoning attack was conducted. ARP is known to be one of the fundamental protocols for IPv4 networks. During this attack, a challenge arose for the algorithm as the traffic pattern closely resembled IPv4 protocols.

ARP Poisoning Attack. This attack allows for intercepting a conversation or data transmission between two connected machines, which results in the information being sent to the attacker.

Figure. 5. Confusion Matrix for Real Traffic. Source: Own by the authors.

doi: 10.25100/iyc.v26i1.13800

In Figure 5, the evaluation results are graphically presented using a confusion matrix, comparing the outcomes of executed attacks. It is evident that all three executed attacks were detected; however, two attacks were false positives. The overall effectiveness of the algorithm proved to be efficient in real-world scenarios and demonstrated the capability to adapt to new challenges.

Conclusions

Considering the results obtained from the classification of machine learning techniques, it can be determined that whenever selecting a technique or algorithm, the type of data to be handled as inputs at all times must be taken into account, as this will determine the percentage of accuracy and false positives.

The use of new technologies and the multitude of new frameworks emerging can facilitate the application of Machine Learning, making the programming and training of models more effective and easier to implement. For example, sklearn contains most of the libraries ready to use in Machine Learning projects among other technologies.

Finally, as a recommendation, it is important to specify the software and hardware resources that might be necessary for the proper execution of the algorithm.

Acknowledgments

Thanks to the University of Cauca, especially the GTI research group, the Colegio Mayor del Cauca University Institution, and the University of Antioquia, in particular the in2lab group, for their support in the development of this proposal.

References

(1) Kaspersky, «¿Qué es la Ciberseguridad?,» Latam Kaspersky, (En línea). Available: https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security. (Último acceso: 12 01 2021).

(2) J. P. Sifre, «IDS de red para la detección de ataques sobre SSH y FTP,» Universidad de Alicante, España, 2020.

(3) B. Y. Navarro, «Blockchain y sus aplicaciones,» 2017. (En línea). Available: https://docplayer.es/74398078-Blockchain-y-sus-aplicaciones.html. (Último acceso: 1 2021).

(4) T. ©. 2021, «TELCO manager,» Telcomanager, 2021. (En línea). Available: https://www.telcomanager.com/es/blog/que-es-el-netflow/. (Último acceso: 12 1 2021).

(5) J. S. A. Enrique Javier Santiago, «Riesgos de Cibersguridad en las Empresas,» Tenología y Desarrollo, vol. 15, pp. 3-33, 2017.

(6) P. P. Angie Valencia, «Internet Industrial de las Cosas (IIOT): Nueva Forma de Fabricación Inteligente,» Fundación Universitaria de Popayán - FUP, Popayán, 2020.

(7) H. G. W. W. Y. G. Yi Zeng, «A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework,» IEEE Xplore, vol. 7, nº Doi:10.1109/ACCESS.2019.2908225, pp. 45182 - 45190, 01 04 2019.

(8) P. R. K.-K. R. C. N. B. Gonzalo De La Torre Parra, «Detecting Internet of Things attacks using distributed Deep Learning,» Journal of Network and Computer Applications, vol. 163, nº https://doi.org/10.1016/j.jnca.2020.102662, 01 08 2020.

(9) J. P. P. C. d. O. L. R. M. V. H. C. d. A. Kelton Pontara Augusto da Costa, «Internet of Things: A survey on machine learning-based intrusion detection approaches,» Computer Networks, vol. 151, nº https://doi.org/10.1016/j.comnet.2019.01.023Get rights and content , pp. 147-157, 2019.

doi: 10.25100/iyc.v26i1.13800

(10) H. K. Q. C. C. M. L. Muhammad Asaad Cheema, «Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things,» IEEE Xplore, vol. 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), nº Doi: 10.1109/DCOSS49796.2020.00074, 2020.

(11) J. E. L. Emilio Berrocal de Luna, «El proceso de investigación educativa II: Investigación - Acción,» Universidad de Granada, España.

(12) Z. T. A. K. B. X. D. M. G. Muhammad Shafiq, «CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques,» IEEE Xplore.

(13) Q. Y. H. H. S. W. L. P. L. W. B. Y. Zhenxiang Chen, «Machine learning based mobile malware detection using highly imbalanced network traffic,» Information Sciences, nº https://doi.org/10.1016/j.ins.2017.04.044, pp. 346-364, 2018.

(14) T. J. L. Ayush Kumar, «EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques,» IEEE Xplore, 22 07 2019.

(15) F. U. R. A. Y. A. T. F. Toshiro Nagata Bolivar, «Análisis de seguridad en tráfico de redes empleando minería de datos,» Revista Ibérica de Sistemas e Tecnologias de Informação, vol. 21, pp. 314-326, 2019.

(16) C. C. J. G. Santiago Eguren, «Modelado probabilístico basado en aprendizaje profundo para la detección de anomalías en el tráfico de red,» XXI Workshop de Investigadores en Ciencias de la Computación, nº http://sedici.unlp.edu.ar/handle/10915/77280, pp. 1-4, 2019.

(17) G. A. G. Montes, «Detecting and classifying malicious TLS network traffic using machine learning,» E.T.S. de Ingenieros Informáticos (UPM), Madrid, 2018.

(18) L. I. B. L. Á. L. V. C. M. B. H. Á. Freddy Daniel Bazante Veloz, «Indicadores para la detección de ataques ransomware,» Revista Ibérica de Sistemas e Tecnologias de Informação, nº 19, pp. 493-506, 2019.

(19) J. B. A. D. Omar M. K. Alhawi, «Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection,» Cyber Threat Intelligence. Advances in information Security, vol. 70, nº https://doi.org/10.1007/978-3-319-73951-9_5, pp. 1-11, 24 04 2018.

(20) P. K. R. Sunil Kumar Singh, «Detecting Malicious DNS over HTTPS Traffic Using Machine Learning,» IEEE Xplore, nº Doi:10.1109/3ICT51146.2020.9312004, 08 01 2021.

(21) Q. Y. Nour Alqudah, «Machine Learning for Traffic Analysis: A Review,» Procedia Computer Science, vol. 170, pp. 911-916, 2020.

(22) B. P. J. B. B. N. Robert Ian McKay, «Machine Learning Algorithms on Botnet Traffic: Ensemble and Simple Algorithms,» ACM Digital Library, nº https://doi.org/10.1145/3314545.3314569, pp. 31-35, 03 2019.

(23) V. P. S. M. S. Gurpal Singh Chhabra, «Cyber forensics framework for big data analytics in IoT environment using machine learning,» Multimed Tools Appl , vol. 79, nº https://doi.org/10.1007/s11042-018-6338-1, p. 15881–15900, 2020.

(24) Y. L. W. F. Rong Chen, «Android Malware Identification Based on Traffic Analysis,» Artificial Intelligence and Security. ICAIS 2019. Lecture Notes in Computer Science(), vol. 11632, nº https://doi.org/10.1007/978-3-030-24274-9_26, p. 293–303, 2019.

(25) Z. S. X. D. M. G. Muhammad Shafiq, «Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city,» Future Generation Computer Systems, vol. 107, nº https://doi.org/10.1016/j.future.2020.02.017, pp. 433-442, 06 2020.

doi: 10.25100/iyc.v26i1.13800

(26) H. Y. D. W. J. D. M. Z. Wanting Yu, «SL-BiLSTM: A Signal-Based Bidirectional LSTM Network for Over-the-Horizon Target Localization,» Journals Hindawi, vol. 2021, nº https://doi.org/10.1155/2021/9992120, 2021.

(27) N. N. y. A. B. Miloud Besnassi, «Face detection based on evolutionary Haar filter,» Pattern Analysis and Applications, vol. 23, nº https://doi.org/10.1007/s10044-019-00784-5, pp. 309-330, 12 02 2020.

(28) R. D. Gómez, «Introducción y optimización estocástica de redes neuronales profundas MLP,» Universitat De Barcelona, Barcelona, 2020.

(29) E. D. d. l. R. Montero, «Máquinas restringidas de Boltzmann para el modelado de sistemas no lineales,» Centro de Investigación y de Estudios avanzados del Instituto Politécnico Nacional, Tesis para Doctorado en Ciencias, 2018.

(30) M. F. I. M. Gilbert Pla Martinez, «Clasificador automático de imágenes de muestras de sangre basado en redes neuronales profundas,» Revista Ingeniería Electrónica, Automática y Comunicaciones, vol. 40, nº 1, pp. 18-30, 2019.

(31) B. S. S. A. M. J. F. D. B. G. N. Chao Liang, «Intrusion Detection System for Internet of Things based on a Machine Learning approach,» IEEE Xplore, nº Doi: 10.1109/ViTECoN.2019.8899448, 2019.

(32) L. A. A. M. T. S. V. S. M. Cristian Cardellino, «Convolutional Ladder Networks for Legal NERC and the impact of Unsupervised Data in Better Generalizations,» The Thirty-Second International Florida Artificial Intelligence Research Conference (Flairs - 32), pp. 155-160, 2019.

(33) E. d. d. m. d. a. p. e. Azure, «Microsoft Azure,» 2020. (En línea). Available: https://docs.microsoft.com/es-es/azure/architecture/reference-architectures/ai/training-deep-learning. (Último acceso: 02 02 2021).

(34) N. f. Cybersecurity, «Cisco Press,» 03 10 2017. (En línea). Available: https://www.ciscopress.com/articles/article.asp?p=2812391&seqNum=5. (Último acceso: 02 02 2021).

doi: 10.25100/iyc.v26i1.13800