IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 10, 2019, accepted December 21, 2019, date of publication January 1, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963407

Evaluation of Cybersecurity Data Set Characteristics for Their Applicability to Neural Networks Algorithms Detecting Cybersecurity Anomalies

XAVIER A. LARRIVA-NOVO, (Member, IEEE), MARIO VEGA-BARBAS, (Member, IEEE), VÍCTOR A. VILLAGRÁ, AND MARIO SANZ RODRIGO

Dpto. Ingeniería Sistemas Telemáticos, ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Corresponding author: Xavier A. Larriva-Novo (xlarriva@dit.upm.es)

This work supported in part by the Secretaría de Educación Superior de Ciencia y Tecnología del Ecuador (SENESCYT).

ABSTRACT Artificial intelligence algorithms have a leading role in the field of cybersecurity and attack detection, being able to present better results in some scenarios than classic intrusion detection systems such as Snort or Suricata. In this sense, this research focuses on the evaluation of characteristics for different well-established Machine Leaning algorithms commonly applied to IDS scenarios. To do this, a categorization for cybersecurity data sets that groups its records into several groups is first considered. Making use of this division, this work seeks to determine which neural network model (multilayer or recurrent), activation function, and learning algorithm yield higher accuracy values, depending on the group of data. Finally, the results are used to determine which group of data from a cybersecurity data set are more relevant and representative for the intrusion detection, and the most suitable configuration of Machine Learning algorithm to decrease the computational load of the system.

INDEX TERMS Cybersecurity, data analytics, data sets, machine learning, neural networks, intrusion detection.

I. INTRODUCTION

The increased complexity of new computer systems and the adaptability of new technological developments is leading to the progress in the application of new methods and techniques of artificial intelligence (AI) in the field of computer security. Specifically, AI has had a greater incidence in the detection of harmful software or anomalies and intrusions, generating new modules to support more efficient and robust decisions [1]. This aid, among other things, allows human interaction to focus on more abstract actions such as general monitoring of systems or the analysis of errors, i.e., false positives. In addition, AI techniques also help people responsible for IT security to manage and analyze the vast quantity of data that new information systems can generate.

One of the most common uses of AI is the generation of new models of intrusion detection systems (IDS). These systems handle large volumes of data that must be evaluated quickly while generating different types of alerts. In addition to the development of new and more efficient IDS, AI has been used as a basis for implementing IDS applying machine learning techniques for the categorization of patterns through explicit and implicit models [2]. These techniques offer high adaptability to the inclusion and processing of new information.

One of the main problems is the abundance of data in contemporary cybersecurity datasets which requires intelligent algorithms, such as machine learning algorithms, for extracting meaningful information. Specifically, its application to IDS involves the need of high amount of features with the objective to select the best approach and detect the possibility of an attack. The problem is important, because a high number of characteristics in a dataset leads to a model overfitting, consequently turning into poor results on the validation datasets [3].

Among all the available machine learning techniques, this research focuses on the study of neural network based computing models.

Thus, this work aims to determine which neural network model produces better analysis results for different types of

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

TABLE 1. Activation functions.

Function	Equation
Linear rectifier	$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$
Softsign	$f(x) = \frac{x}{1 + \lvert x \rvert}$
Hyperbolic tangent	$f(x) = \frac{2}{1 + e^{-2x}} - 1$

data specific to the context of information security. That is, considering that, depending on the work scenario, we will have a specific type of data, this research aims to show which set of parameters of a neural network favor the creation of detection mechanisms that provide an optimal response. Specifically, this work focuses on the study and comparison of multilayer and recurrent neural networks, with special attention to data of a temporary nature. Finally, the novelty of this work is that it presents a study on the behavior of different configurations of neural networks (multilayer and recurrent) based on the proposal of a categorization of a cybersecurity dataset. In this way it has been possible to determine which neural network configuration offers the best results, in terms of accuracy, for each category of data.

To achieve the objective of this research, this article has been organized as follows. First, Section II presents a theoretical perspective of the neural networks of interest for this work, i.e. the multilayer networks and the recurrent networks. Section III analyzes the most relevant works that have opted for the application of these specific types of neural networks for the detection of cyberattacks. Sections IV and V provide the study and justification of the choice of data used in this research. Finally, sections VI and VII present the results and conclusions of this work.

II. NEURAL NETWORKS

Artificial neural networks are complex systems constructed by simple computational units called neurons, analogous to the behavior of neurons in biological brains. These neurons are interconnected through links that manage the activation state of adjacent neurons.

Each neuron works according to an activation function, which relates its input to its output. The most common activation functions are detailed in Table 1. The connections or weights that connect the neurons are updated according to the learning algorithm used, explored in Section II-B, to reduce the error between the desired output and the obtained output.

A. NEURAL NETWORKS UNDERSTUDY

The most widespread neural networks are the so-called feedforward networks, where the signal travels in a single direction (from the input to the output), and there are no loops, which means that the output of one layer does not affect the same layer. They can be composed of one layer of neurons (monolayer) or several (multilayer). This study focuses only on multilayer networks; the basic structure is shown in Figure 1.

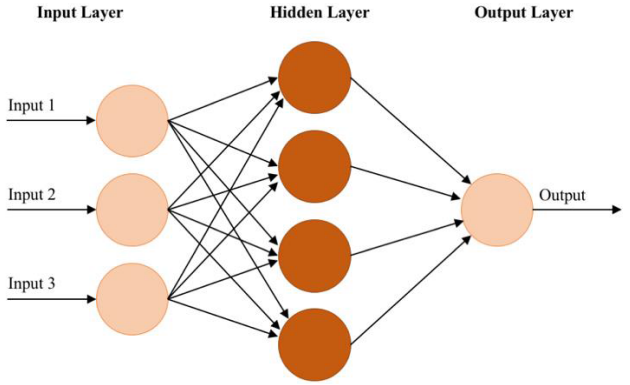

FIGURE 1. Basic structure of a multilayer neural network.

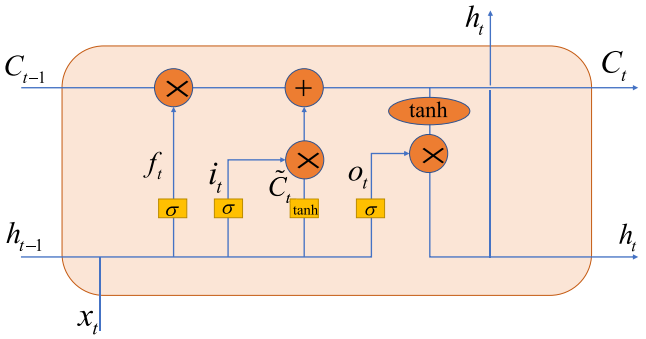

FIGURE 2. Basic scheme of an LSTM unit.

In contrast to the feedforward neural networks, there are recurrent neural networks (RNNs) where the signals can travel in both directions, introducing loops in the network, which results in the output of a layer affecting this same layer, so it can give the network the memory property. Therefore, this type of neural network is generally used for the modeling of time series or tasks [4]. The use of RNNs is lower compared to feedforward networks, partly because the learning algorithms are much less effective (to date). However, they are presented as a very interesting alternative [5].

One of the most commonly used RNN architectures is the long short-term memory (LSTM) network [6], which minimizes the problem of gradient descent. Figure 2 presents the basic scheme of a processing unit of this type of neural network.

The key to understanding the functioning of these networks are the values C_{t-1} and C_t, which represent the state of each cell. Thus, a cell can maintain its state in time (through the horizontal line that connects C_{t-1} and C_t) regulating the flow of information between the input and the output through nonlinear doors.

B. MACHINE LEARNING ALGORITHMS UNDERSTUDY

Learning is the process by which the degrees of freedom of a neural network are adapted through a process of stimulation,

the process by which the neural network modifies its weights (connections between neurons) in response to input information.

Learning methods can be divided into supervised learning or unsupervised learning. In a supervised learning model, the network receives a set of behavioral examples already tagged; in contrast, in the case of unsupervised learning, the inputs are the only source of information for learning, and the network learns to categorize the inputs.

For the problem addressed in this article, we only focus on supervised learning for error correction, which is intended to minimize a cost function based on the error signal, so that the response of each output neuron of the network is as close as possible to the objective response. The methods studied for this error correction are the *stochastic gradient descent (SGD)* [7], *the root mean square propagation (RMSProp)* [8], *Adam* [9], *Adaptive Gradient Algorithm* (adagrad) [10], *adadelta* [11], *adamax* [9] and *Nadam* [12]

The most important problems related to learning in recurrent neural networks are grouped around two concepts: the vanishing gradient and the exploding gradient [13], [14]. Problems related to exploding gradient refer to a large increase in the gradient norm during training due to the explosion of long-term components, which grow exponentially. The problems related to the vanishing gradient have an opposite behavior; that is, the long-term components decrease exponentially until reaching a norm close to zero. Both problems make it impossible for the model to learn the correlation between temporally distant events. In [15], several solutions were proposed to deal with these problems, such as scaling down gradients whenever a threshold is exceeded (exploding gradient) or using a regularization term that has a preference for some values, which implies that the gradients neither increase or decrease in magnitude (vanishing gradient).

III. THE USE OF NEURAL NETWORKS IN CYBERSECURITY

The application of artificial neural networks to the context of computer security is mainly focused on the detection of intrusions in a network since artificial neural networks are considered an efficient approach to pattern classification. The main problem with these algorithms consists of the high calculation requirements and the long training cycles they require, hindering their incorporation into commercial applications [16].

Even though different cybersecurity proposal are based on old public datasets, their results are not comparable due to different causes: different algorithms consider different features, implementation of pre-filtering operation and the use of different split between test and training data set [17]. For this, the current article makes an exhaustive comparation of different groups of characteristics applying different algorithms of feed forward neural networks (FFNN) and RNN setting and testing different configurations proposed as mentioned in section VI

Artificial neural networks, as discussed above, have been used in multiple and diverse problems related to IDSs. An example of the performance of a simple network can be found in [18], where an accuracy of 98.86% was obtained by making use of a three-layer neural network with backpropagation. This accuracy resembles that achieved by other algorithms (supported vector machine (SVM), naïve Bayes, and C4.5). The authors of [19] compared between models based on multilayer neural networks and SVM algorithms. The results presented in this work demonstrate a similar accuracy of approximately 99%.

Different algorithms such as DNN and non-DNN were presented, in which they compared different models using the data set KDD99 [20] obtaining an accuracy of 92.9% applying a DNN, approaching better results versus non-DNN models [21]. This results can be improved selecting different values for the configuration of the algorithm, such as the number of layers, normalization functions, number of nodes or activation functions [22].

In [23], a more complex network was presented, where a neural network with a time delay or time-delay neural network (TDNN) was used to develop an IDS capable of collecting the characteristics of the monitored network packets. These features were grouped and introduced to a neural network with a time delay that would classify them by setting an alarm if necessary. This work verified, by performing diverse tests, that the implemented system detects attacks more quickly than by using expert rules systems such as Snort.

Other works, such as those presented in [24], compared the use of a competitive enhanced learning network (improved competitive learning network (ICLN)) with the self-organizing map (SOM) model. ICLN networks are used in unsupervised learning, while SOM is a fully connected and single-layer model used in supervised learning. After performing experiments with both networks, a similar accuracy was obtained, although the SOM network requires longer processing time.

Another type of artificial neural network architecture, popular for computer security environments, is recurrent networks. In [25], an IDS architecture was presented where distributed-time deferred neural networks (DTDNN) were used, which provides a simple and efficient method for classifying data sets because of its high speed and fast convergence rates, with satisfactory results. Another type of recurrent neural network architecture widely used for the development of IDS is the so-called long short-term memory (LSTM), which was introduced in [7], [4], [26], [27]. In [7], an accuracy of 97.54% was presented, which is equal to other neural network architectures but had a false positive rate of 9.98%, quite high, although below most others architectures of neural networks with which it was compared. Additionally, [26] presented an architecture that yields an overall accuracy of 93.72%, although, for recognition attacks, which were addressed in this work, the accuracy was very low (56.4%). The work performed in [4] achieved high accuracy in DoS attacks and normal connections but low performance in reconnaissance attacks, R2L and U2R. Finally, [27] presented results that

showed that the use of recurrent networks for intrusion classification tasks is more accurate than with other learning algorithms. In addition to representing an independent functional unit, neural networks allow their combination with other machine learning algorithms to achieve better performance. This form of action can be found in [28], where spectral grouping and deep neural networks were used for the development of an IDS.

Undoubtedly, data are the key piece of any machine learning algorithm because they are the source of learning information to allow proper classifying of each new entry. For this reason, there are studies focused on categorizing patterns of a data set such as [29], either focused on their study or the reduction of features in the case of multidimensional data sets. However, these works lack a detailed analysis of the data used for learning as the categorization proposed in this research and whose objective is to improve the efficiency, performance, and reliability of the algorithm.

IV. CATEGORIZATION OF A CYBERSECURITY DATA SET

The analysis of some existing data sets (UNB-ISCX-2012 [30], CTU-13 [31], MACCDC [32] or UGR'16 [33]) allows us to observe that they have different formats and feature, so that we can say that cybersecurity data sets are highly heterogeneous.

The methodology proposed in this case aims to simplify multidimensional data sets, choosing only the relevant characteristics for the specific scenario and thus making the learning algorithm lighter. Specifically, the novelty of this work is reducing this multidimensionality by groups of characteristics, instead of using an individual approach, as an alternative of those presented in the current state of the art [34]–[37]. For this purpose, this research proposes three main feature groups: *basic connection characteristics, content characteristics, and traffic statistical characteristics*. Some of these characteristics will have more or less weight, depending on the type of attack being detected. For example, time-based traffic characteristics are especially useful for detecting high volumes of data in a small interval of time and, therefore, appropriate for possible denial-of-service (DoS) attacks. The following subsections describe each of them.

A. BASIC CONNECTION CHARACTERISTICS

This category includes the basic features that are usually found in a TCP header. They are intrinsic characteristics of a connection and can be useful for general-purpose network analysis, as well as being used for intrusion detection. Examples of these characteristics are the duration, the service, the protocol, or information about the origin and destination of the connection.

B. CONTENT CHARACTERISTICS

These characteristics refer to the content of the packets of the connection that is being analyzed. It is more specific information, so its use is more oriented to the detection of certain attacks instead of focusing on the detection of anomalies in a network. Characteristics that would be classified in this category are, for example, the number of unsuccessful authentication attempts, information about access to a root console or the number of file creation operations.

C. TRAFFIC STATISTICAL CHARACTERISTICS

This category includes characteristics that are not related to a single connection but statistical information related to a specific property [38]. That is, by selecting a particularity, such as the same host, these statistical properties can be the number of connections to that host or the percentage of connections to that host that have the same service. In general, they provide more information than previous groups of characteristics. This category is divided into subcategories depending on the characteristics studied.

1) TRAFFIC CHARACTERISTICS BASED ON TIME

These characteristics are obtained in a time window of 2 seconds, considering that recognition attacks are based on the generation of many connections in a short period. They can be, for example, the number of connections to the same host or the number of connections that have SYN errors during the defined time window.

2) TRAFFIC CHARACTERISTICS BASED ON THE SOURCE ADDRESS

These are characteristics referred to information relative to the same source host. Specifically, the analyzed data set uses a window of 100 connections to the same host in a certain period.

3) TRAFFIC CHARACTERISTICS BASED ON THE DESTINATION ADDRESS

These are identical characteristics to the previous case but grouping the information according to the destination host. For example, a possible feature for this category is the number of connections to the same service.

V. DATA SET UNDERSTUDY

For the problem addressed in this paper, the database used must contain information about different connections in a network together with a label that specifies whether the connection is an attack and its type or a normal connection. The algorithm used for detection will make use of supervised learning, and therefore, it is necessary that each type of data is labeled and classified.

In this case, the data set UNSW-NB15 [39], [40], which is widely used in cybersecurity [41]–[44] and considered as a benchmark data set [45], was chosen. The choice of this data set is motivated by several factors: the validity of the attacks the labeling of these, and the classification of the data, similar to that presented in the previous section.

The UNSW-NB15 data set is composed of 49 features, 47 of which are related to the attributes of the data; the last two features are related to the type of attack and the behavior in the data set (normal or attack). This data set contains

TABLE 2. Classification of Data set UNSW-NB15 features.

Group	Feature
Basic characteristics	srcip, sport, dstip, dsport, proto, state, dur, sbytes, dbytes, sttl, dttl, sloss, dloss, service, sload, dload, spkts, dpakts
Content characteristics	swin, dwin, stcpb, dtcpb, smeansz, dmeansz, trans_depth, res_bdy_len
Traffic characteristics based on time	sjit, djit, stime, ltime, intpkt, dintpkt, tcprtt, synack, ackdat
Traffic characteristics based on source address	ct_srv_src, ct_src_ltm, ct_src_dport_ltm, ct_src_dst_ltm
Traffic characteristics based on destination address	ct_srv_dest, ct_dst_ltm, ct_dst_dport_ltm, ct_dst_src_ltm

approximately 2.540.460 simple connections. Several efforts have been made to reduce the number of representative features of each of the connections without implying a reduction in the accuracy of the response. This is the case in [46], where a neural network was used to determine the accuracy obtained in the detection after the reduction in features using correlation or entropy techniques. In this article, we check the accuracy that is achieved using each group of data to infer which type of characteristics have more influence on the detection of an attack.

Modern attacks are found in this data set, which is divided into nine categories: *Fuzzers*, *Analysis*, *Backdoors*, *DoS*, *Exploits*, *Generic*, *Reconnaissance*, *Shellcode,* and *Worms*. This paper focuses on recognition attacks, partly because of the large number of records of this type present in the data set as well as their versatility. Generally, these attacks correspond to the first phase of a later attack of greater magnitude; therefore, it is an interesting starting point for the early detection of cyberattacks and consequently allows reacting as soon as possible when an organization faces a cyberattack.

As previously mentioned, each connection (registry) is defined by 47 features. For the grouping of these characteristics, the detailed scheme in Section IV was followed. Each of the proposed groups has been created for the experiment in the following way: Group 1 or basic characteristics, Group 2 or basic characteristics and content characteristics, Group 3 or basic characteristics, content characteristics and traffic characteristics based on time and Group 4 or basic characteristics, content characteristics, traffic characteristics based on time, traffic characteristics based on the source address and traffic characteristics based on the destination address. Table 2 shows a summary of this classification.

This division is useful for performing different tests with different groups and determining which variables influence the final result of the algorithm.

Certain features, such as *protocol*, *service* or *flag*, are not presented numerically, which is why one-hot coding was used [47]. In addition, some of the characteristics, such as *duration* or *src bytes (sbytes)*, present data with widely dispersed values over a wide numerical range, so they are normalized by both the min-max function

$$f\,(x) = \frac{x - Min}{Max - Min} \tag{1}$$

and the Z-score function

$$\sigma\alpha f(x) = \frac{x - \sigma}{\alpha} \tag{2}$$

where σ is the average and α is the standard deviation.

VI. EXPERIMENTATION AND DISCUSSION OF THE RESULTS

For the implementation of neural networks, Python was used as the programming language, and the TensorFlow library, an open-source library created by Google Brain Team. This library offers all the necessary tools to build, train and test the effectiveness of artificial neural networks.

Throughout this section, the results obtained after the tests were carried out with the two neural network models, the activation functions, the different learning algorithms, and the different groups of characteristics are presented. In the different experiments tested, both the activation function of the neurons and the optimizer were modified.

For experimentation and analysis, an *Adam* optimization algorithm was used. Once defined, the results related to the accuracy of the different activation functions were obtained. With these accuracy values, the best activation function was selected, and several experiments were carried out with the optimizers.

The variables that must be monitored to determine the performance of each network are the accuracy and the cost. To analyze the accuracy, we compared the test data for which connection labels were predicted, and the true values of these labels, obtaining the total number of predicted labels by the algorithm and obtaining the percentage of accuracy. Where appropriate, the cost focused on measuring the error between the test data for which connection labels were predicted, and the true value of the labels calculated the cross-entropy of the normalized exponential function. Once this error was obtained, it was averaged, and a value was obtained that was reduced in the next iteration of training. Finally, the weights of the neural network were initialized with random values.

A. ANALYSIS OF MULTILAYER NEURAL NETWORKS

The developed multilayer neural network consisted of three fully connected layers, an input layer, a hidden layer, and an output layer. The distribution of neurons of this network in each layer followed the set of rules defined in [48] and detailed in Table 3.

For the tests carried out in this research, the combinations of the activation function and the optimizers shown

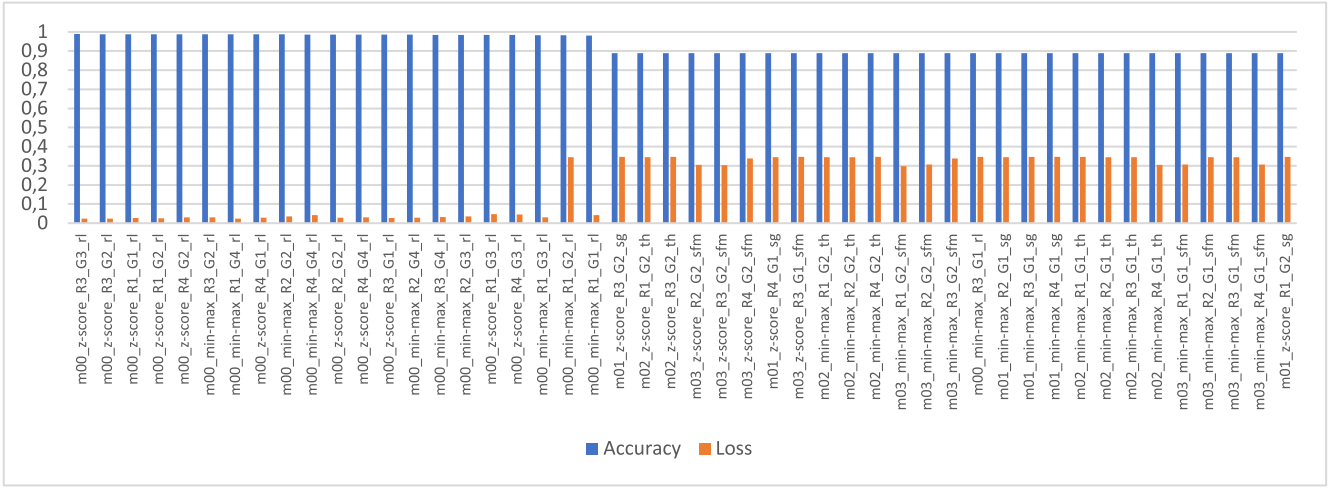

FIGURE 3. Feedforward neural network results in terms of loss and accuracy. The vertical axis shows a normalized value of accuracy and loss performance; while horizontal is to point out the most representative feed forward neural networks configurations of the proposed following the next pattern: *<configuration code>_<normalization function>_<rule selected for the number of nodes in the hidden layer>_<group of characteristics proposed>_<activation function>.*

TABLE 3. Rules of Calculation of Nodes in Hidden layers.

Rule code	Method
R_1	$H = 0.75\ x\ Input + Output$
R_2	$H = \frac{(Input + Outpt)}{2}$
R_3	$H = 0.70\ x\ Input$
R_4	$H = 0.90\ x\ Input$

TABLE 4. Testing performed for each group of characteristics using the multilayer neural network.

Configuration Code	Rule	Activation Function	Optimizer
m00	R_1,2,3,4	Linear rectifier	Adam
m01	R_1,2,3,4	Sigmoid	Adam
m02	R_1,2,3,4	Hyperbolic tangent	Adam
m03	R_1,2,3,4	Softsign	Adam
m04	Best rule	Best activation function	Gradient descent
m05	Best rule	Best activation function	RMSProp
m06	Best rule	Best activation function	Adagrad
m07	Best rule	Best activation function	Adadelta
m08	Best rule	Best activation function	Adamax
m09	Best rule	Best activation function	Nadam

in Table 4 were tested. For each configuration, both min-max and Z-score normalization were used. Additionally, each of these configurations was analyzed with each group of characteristics defined in Table 2, focusing on the best accuracy and determining the best configuration for each type of data.

First, to determine the activation function, m00, m01, m02, and m03 tests were performed, using the optimizer Adam and both normalization functions, min-max and Z-score. The results of this experiment are shown in Figure 3 in a descendent form, from the best performances in terms of accuracy to the worst. These results indicate that the best results (higher accuracy and earlier convergence) were obtained by the use of a linear rectifier for all the groups of characteristics, obtaining values of approximately 98% of accuracy using the R_1 rule and the Z-score normalization function for each of the groups of characteristics. Similarly, it can be observed the activation function, the corresponding rule to determine the number of specific nodes for the hidden layers, as well as the most appropriate normalization function.

Then, in Figure 4, the experiments performed once the linear rectifier function is set, show that the maximum accuracy was reached for each of the groups of characteristics of experiment m00, executed on each of the groups of characteristics and achieved an accuracy of 98.56%. The results of these tests are detailed in Table 5.

For the data belonging to Group 2, the activation function that achieved the best result was the linear rectifier, with an accuracy of 98.8%. Setting this as the activation function, the experiments related to the optimizers show that the highest accuracy was obtained with the Adam optimizer (value indicated previously), followed by RMSProp, with an accuracy of 98.18%. The main disadvantage presented RMSProp was that the accuracy did not remain stable,

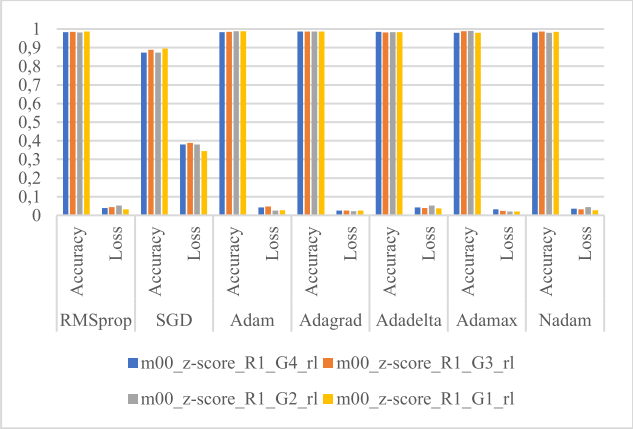

FIGURE 4. Results in terms of loss and accuracy concerning the optimizers.

TABLE 5. Results for each group of characteristics using a multilayer neural network.

Accuracy	Configuration Code /Accuracy			
	m00_z-score_R1_G1_rl	m00_z-score_R1G2_rl	m00_z-score_R1G3_rl	m00_z-score_R1G4_rl
RMSProp	0.987	0.9818	0.985	0.983
SGD	0.895	.08738	0.889	0.873
Adam	0.988	0.988	0.984	0.983
Adagrad	0.986	0.986	0.983	0.98
Adadelta	0.983	0.983	0.982	0.982
Adamax	0.98	0.988	0.983	0.981
Nadam	0.985	0.98	0.981	0.988

but fading occurred in its value throughout the rest of the experiment. For the gradient descent optimizer, accuracy values decreased by approximately 12%.

In the experiments performed with the Adam optimizer on the data belonging to Group 3, those data related to a time window indicated that the best performance was obtained using the linear rectifier function as the activation function (98.43% accuracy).

Finally, for the data belonging to Group 4, that is, traffic data characterized by traffic direction, the accuracy values were worse than for Groups 1, 2 and 3. The best combination was achieved with the linear rectifier as the activation function and the Adam optimizer. Thus, comparing the obtained data, we can affirm that the best results were obtained using the Adam optimizer and the linear rectifier function as an activation function, applying a normalization function of type Z-score and R_1.

B. ANALYSIS OF RECURRENT NEURAL NETWORKS

The recurrent neural network developed is based on an LSTM architecture. This type of network was chosen due to its learning capacity and the good results it has achieved in other similar projects. This neural network is composed of an LSTM neuron fed by a determined number of connections. These connections contain a distinct variable number of characteristics. The tests performed with this network are

TABLE 6. Testing Performed for each group of characteristics using the recurrent neural network.

Configuration Code	Activation Function Optimizer
r00	Adam
r01	SGD
r02	RMSProp

TABLE 7. Results for each group of characteristics using an LSTM neural network.

Configuration Code	Group 1	Group 2	Group 3	Group 4
r00	0.98	0.983	0.984	0.984
r01	0,793	0.81	0.8723	0.798
r02	0.973	0.955	0.964	0.959

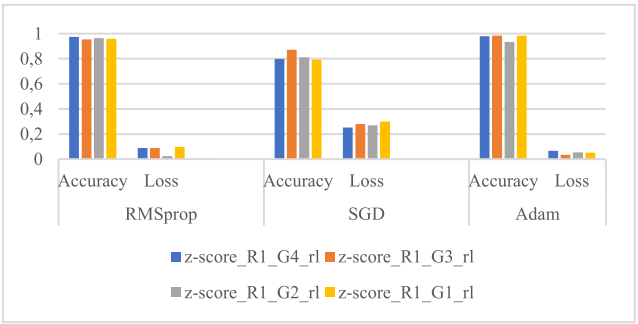

FIGURE 5. Results in terms of loss and accuracy referring to recurrent neuronal network optimizers.

detailed in Table 6, and the corresponding results are shown in Table 7. For Group 1, the maximum accuracy was 98% using the Adam optimizer. The optimizers that reduced the cost more quickly were Adam and RMSProp.

C. COMPARISON BETWEEN MULTILAYER AND RECURRENT NEURAL NETWORKS

Another objective of this work is to compare the performance of multilayer neural networks versus recurrent neural networks. To do this, the maximum accuracy obtained for each group of data were compared.

In general, there was no clear benefit that justified the use of a recurrent neural network since the accuracy obtained was very similar to those obtained through the use of a multilayer neural network. As explained in [10], this is due to the complexity inherent in the correct training of a recurrent neural network. However, the computing cost offered by recurrent networks is significantly greater (9 times higher) than that associated with multilayer networks.

The tests performed by these configurations are presented in Table 7 and Figure 5. In the case of Group 1, the maximum accuracy was achieved using the Adam optimizer (98%). Similar results were achieved for Groups 2, 3 and 4. The optimizers that reduced the cost faster were Adam and RMSProp.

TABLE 8. Comparison between multiples researches approaches.

Algorithm	Number of features	Accuracy
Decision Trees [45]	22	89.86%
Decision Trees [45]	13	89.76
Decision Trees[50]	47	85.41
J48 Classifier [42] (Worms attacks)	25	99.94
Our proposed FFNN	19	98.8%
Our proposed RNN	19	98%

D. RELEVANCE OF THE DATA GROUPS

After performing the described experiments with the multilayer neural network on different data groups, it was observed that when using only one set of characteristics, it was possible to obtain good prediction results. However, in the experiments performed with the complete data set of each group of characteristics, it was possible to observe better performance and accuracy with groups 1 and 2 with a value of 99%, while with groups 3 and 4, an accuracy was obtained of approximately 98%. Regarding the recurrent network, the maximum results obtained from accuracy in the experiments were similar and close to 98%.

E. ARCHITECTURE OF THE NEURAL NETWORK AND NORMALIZATION OF INPUT VALUES

Each neural network can offer different results depending on its configuration; that is, these results depend on the nodes of its architecture, the optimization functions and the normalization of input values. In this case, the rule that provided the greatest accuracy in each of the exposed data groups was R_1.

F. COMPARISON BETWEEN MULTIPLES RESEARCHES APPROACHES

Different researches have been presented related to our work. In [44] the authors analyzed the dataset UNSW-NB15 by several machine learning techniques offering a data preprocessing technique to reduce redundant data. The research explores an analysis with reduced features instead of 47 features presented in the current dataset. The best performance was with 22 features providing an accuracy of 89.86% followed by an accuracy of 89.76% using 13 features applying Decision Trees. Furthermore, some recent researches have studied the current datasets and most of them have done an evaluation of machine learning techniques such as [49], where the UNSW-NB15 was evaluated by different machine learning algorithms such as Decision Trees, Naïve Bayes and Support Vector Machine, obtaining the best accuracy by Decision Trees (C5.0) of 85.41%. Also, in [41] the authors present a feature selection for rare cyber-attacks, where they propose an evaluation of multiples algorithms with the objective to detect the best accuracy for multi class classification, obtaining an accuracy in the best case (for worms attacks) of 99.94%. Table 8 shows a representation of different related researches and their comparison with the results presented in this article.

VII. CONCLUSION

This work explored the application of neural networks to the detection of cybersecurity intrusions with two main objectives. First, the categorization of a data set (UNSW-NB15), dividing its characteristics into basic, content, traffic statistics and direction-based methods, to analyze which of these groups are the most relevant for the detection of anomalies, and to reduce training and reduce the loss of the models implemented. The second objective focused on determining which neural network can offer a better performance according to the data available for its training.

The experiments performed, using the data set and the proposed categorization, allowed several conclusions to be drawn. The optimal results for each group of data were identified according to the type of neural network, the activation function, the optimization function, and the network architecture, as detailed in sections VI-a and VI-b. Additionally, the results show that when using only one group of data, an accurate prediction of the attack can be obtained, independent of the neural network topology. Thus, the configuration proposed as m00_z-score_R1_G1_rl obtained an accuracy similar to the configuration m00_z-score_R1_G4_rl, decreasing the load of the algorithm in terms of performance, but with a smaller number of characteristics, as detailed in VI-a.

Regarding the comparison between the different neural network architectures analyzed, there was no substantial improvement when using recurrent networks instead of multilayer networks, which was most likely due to the difficulty of training a recurring network.

Also, this article makes a comparison between different works that uses the same dataset and propose a similar idea in terms of characterization of features, selecting the most appropriate to get the best performance as possible in terms of accuracy. The results have exposed that our proposed model and characteristics have obtained the best accuracy with the FFNN and the group one of characteristics with 19 features.

Finally, as future research drawn from this work, it is proposed, first, to extend the proposed methodology to a cybersecurity data set with more information, such as the one proposed by the Universidad de Granada in [26] ($\approx$ 240M flows of data and real traffic). Additionally, to improve the metric that estimates the performance of the algorithm, it is advisable to determine the types of predictions that are made and not only obtain the percentage of the accuracy.

REFERENCES

[1] B. Geluvaraj, P. M. Satwik, and T. A. Kumar, "The future of cybersecurity: Major role of artificial intelligence, machine learning, and deep learning in cyberspace," in *Proc. Int. Conf. Comput. Netw. Commun. Technol.*, 2019, pp. 739–747.

[2] S. Dilek, H. Çakır, and M. Aydın, "Applications of artificial intelligence techniques to combating cyber crimes: A review," 2015, *arXiv:1502.03552*. [Online]. Available: https://arxiv.org/abs/1502.03552

[3] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, 2015, pp. 1200–1205.

[4] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, 2016, pp. 1–5.

[5] D. Berman, A. Buczak, J. Chavis, and C. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, Apr. 2019.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[7] T.-T.-H. Le, J. Kim, and H. Kim, "An effective intrusion detection classifier using long short–term memory with gradient descent optimization," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2017, pp. 1–6.

[8] T. Tieleman and G. Hinton, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[10] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[11] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: https://arxiv.org/abs/1212.5701

[12] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.

[13] D. Nikolov, I. Kordev, and S. Stefanova, "Concept for network intrusion detection system based on recurrent neural network classifier," in *Proc. IEEE 27th Int. Sci. Conf. Electron. (ET)*, Sep. 2018, pp. 1–4.

[14] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[15] V. Garcia-Font, C. Garrigues, and H. Rifà-Pous, "Difficulties and challenges of anomaly detection in smart cities: A laboratory analysis," *Sensors*, vol. 18, no. 10, p. 3198, Sep. 2018.

[16] N. Papernot, P. Mcdaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 49–54.

[17] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *Proc. 10th Int. Conf. Cyber Conflict (CyCon)*, May 2018, pp. 371–390.

[18] B. Subba, S. Biswas, and S. Karmakar, "A neural network based system for intrusion detection and attack classification," in *Proc. 32nd Nat. Conf. Commun. (NCC)*, Mar. 2016, pp. 1–6.

[19] M. Moradi and M. Zulkernine, "A neural network based system for intrusion detection and classification of attacks," in *Proc. IEEE Int. Conf. Adv. Intell. Syst.-Theory Appl.*, 2004, pp. 15–18.

[20] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

[21] R. K. Vigneswaran, R. Vinayakumar, K. Soman, and P. Poornachandran, "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," in *Proc. 9th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2018, pp. 1–6.

[22] R. Vinayakumar, H. B. B. Ganesh, P. Poornachandran, M. A. Kumar, and K. P. Soman, "Deep-Net: Deep neural network for cyber security use cases," 2018, *arXiv:1812.03519*. [Online]. Available: https://arxiv.org/abs/1812.03519

[23] O. Al-Jarrah and A. Arafat, "Network intrusion detection system using neural network classification of attack behavior," *J. Adv. Inf. Technol.*, vol., 6, no. 1, pp. 1–8, 2015.

[24] C. Obimbo, K. Ali, and K. Mohamed, "Using IDS to prevent XSS attacks," in *Proc. Int. Conf. Secur. Manage. (SAM)*, 2017, pp. 233–239.

[25] J. Deny and M. Sundhararajan, "Neural networks and machine learning techniques for intrusion detection system," *Int. J. Digit. Commun. Netw.*, vol. 2, no. 1, pp. 5–8, 2015.

[26] R. C. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection," *South Afr. Comput. J.*, vol. 56, no. 1, Sep. 2015.

[27] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[28] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, p. 1701, Oct. 2016.

[29] P. Suyal, J. Pant, A. Dwivedi, and M. C. Lohani, "Performance evaluation of rough set based classification models to intrusion detection system," in *Proc. 2nd Int. Conf. Adv. Comput., Commun., Autom. (ICACCA)*, Sep. 2016, pp. 1–6.

[30] K. Kato and V. Klyuev, "Development of a network intrusion detection system using apache Hadoop and spark," in *Proc. IEEE Conf. Dependable Secure Comput.*, Aug. 2017, pp. 416–423.

[31] D. S. Terzi, R. Terzi, and S. Sagiroglu, "Big data analytics for network anomaly detection from netflow data," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 592–597.

[32] D. Krovich, A. Cottrill, and D. J. Mancini, "A cloud based entitlement granting engine," in *National Cyber Summit*. Cham, Switzerland: Springer, 2019, pp. 220–231.

[33] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Theron, "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs," *Comput. Secur.*, vol. 73, pp. 411–424, 2018.

[34] B. B. Zarpelão, R. S Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[35] P. Ravi Kiran Varma, V. Valli Kumari, and S. Srinivas Kumar, "A survey of feature selection techniques in intrusion detection system: A soft computing perspective," in *Progress in Computing, Analytics and Networking*. Singapore: Springer, 2018, pp. 785–793.

[36] R. Thomas and D. Pavithran, "A survey of intrusion detection models based on NSL–KDD data set," in *Proc. 5th HCT Inf. Technol. Trends (ITT)*, Nov. 2018, pp. 286–291.

[37] R. Zuech and T. M. Khoshgoftaar, "A survey on feature selection for intrusion detection," in *Proc. 21st ISSAT Int. Conf. Rel. Qual. Design*, 2015, pp. 150–155.

[38] Z. Wang, "The applications of deep learning on traffic identification," BlackHat USA, San Francisco, CA, USA, Tech. Rep., 2015, vol. 24.

[39] N. Moustafa and J. Slay, "UNSW–NB15: A comprehensive data set for network intrusion detection systems (UNSW–NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[40] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW–NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016.

[41] S. Bagui, E. Kalaimannan, S. Bagui, D. Nandi, and A. Pinto, "Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset," *Secur. Privacy*, vol. 2, no. 6, pp. 1–13, Nov./Dec. 2019.

[42] T. Janarthanan and S. Zargari, "Feature selection in UNSW–NB15 and KDDCUP'99 datasets," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2017, pp. 1881–1886.

[43] L. Zhiqiang, G. Mohi-Ud-Din, L. Bing, L. Jianchao, Z. Ye, and L. Zhijun, "Modeling network intrusion detection system using feed–forward neural network using UNSW–NB15 dataset," in *Proc. IEEE 7th Int. Conf. Smart Energy Grid Eng. (SEGE)*, Aug. 2019, pp. 299–303.

[44] V. Kumar, A. K. Das, and D. Sinha, "Statistical analysis of the UNSW-NB15 dataset for intrusion detection," in *Computational Intelligence in Pattern Recognition*. Singapore: Springer, 2020, pp. 279–294.

[45] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Comput.*, pp. 1–22, 2019.

[46] Akashdeep, I. Manzoor, and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert Syst. Appl.*, vol. 88, pp. 249–257, Dec. 2017.

[47] M. Cassel and F. Kastensmidt, "Evaluating one–hot encoding finite state machines for SEU reliability in SRAM-based FPGAs," in *Proc. 12th IEEE Int. On-Line Test. Symp. (IOLTS)*, Aug. 2006, p. 6.

[48] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection," *Comput. Secur.*, vol. 75, pp. 36–58, Jun. 2018.

[49] S. Meftah, T. Rachidi, and N. Assem, "Network based intrusion detection using the UNSW-NB15 dataset," *Int. J. Comput. Digit. Syst.*, vol. 8, no. 5, pp. 478–487, 2019.

XAVIER A. LARRIVA-NOVO (Member, IEEE) received the M.Sc. degree in cybersecurity from the Universidad Politécnica de Madrid (UPM), Spain, in 2018, where he is currently pursuing the Ph.D. degree in telecommunications engineering. He is currently a Researcher of telematics engineering with UPM. He has been involved in European research projects related with network management, security design in services, network security, machine learning, and high-performance computing as well as different national projects. He is an active IEEE Computational Intelligence Society Member.

MARIO VEGA-BARBAS (Member, IEEE) was born in Guadalajara, Spain, in 1984. He received the B.S. and M.S. degrees in computer science from the University of Alcala, Spain, in 2009, and the Ph.D. degrees in telematics and in applied medical technology from the Universidad Politécnica de Madrid (UPM), Spain, and the KTH-Royal Institute of Technology, Sweden, in 2016. He is currently an Assistant Professor and a Senior Researcher within the research group on Telecommunication and the Internet Networks, and Services with UPM. Previously, he has been a Postdoctoral Researcher with the Institute of Environmental Medicine, Karolinska Institute, Sweden. His research interests include data analysis and visualization, ubicomp, pervasive sensitive services, the design of smart spaces, ambient intelligence, user-oriented security, and the development of the IoT solutions in healthcare environments.

VÍCTOR A. VILLAGRÁ received the Ph.D. degree in computer science from the Universidad Politécnica de Madrid (UPM), Spain, in 1994. He has been an Associate Professor of telematics engineering with UPM, since 1992. He has been involved in several international research projects related with network management, advanced services design and network security, as well as different national projects. He is author or coauthor of more than 60 scientific articles. He is also the author of a textbook about *Security in Telecommunication Networks*.

MARIO SANZ RODRIGO was born in Madrid, Spain, in 1989. He received the degree in communication systems engineering from University Carlos III (UC3M), Spain, in 2017, and the M.Sc. in cybersecurity from the Universidad Politécnica de Madrid (UPM), Spain, in 2019. He is currently a Researcher of telematics engineering with UPM. He has been involved in European and national projects related with video coding, network monitoring, microservice virtualization, telemetry in electrical network with PLC using PRIME protocol, security in industrial environments, and SCADA systems.