Sumer Journal for Pure Science (ISSN: 2790-7031)
SJPS 2025 (June), Vol. 4, No. 1, p.p.: 102-114

Improving Network Security by Applying Unsupervised Machine Learning Techniques

Zaydon L. Ali [1]*, Ahmed Ramzi Rashid [2], Thoraya Ali Shaker [3]
[1]*College of Political Science, Mustansiriyah University, Baghdad, Iraq*
[2]*College of Political Science, Mustansiriyah University, Baghdad, Iraq*
[3]*College of Political Science, Mustansiriyah University, Baghdad, Iraq*
* zaydonlatif@uomustansiriyah.edu.iq

Abstract

Computer networks are increasingly exposed to sophisticated and diverse attacks, threatening the security of data and services. The difficulty lies in detecting new and unknown attacks, especially with the increasing volume and size of network traffic. This research seeks to address this problem by developing a network event detection system based on unsupervised methods, which enables it to detect abnormal behavior patterns that may indicate the presence of an attack. This research aims to develop an innovative network event detection system using unsupervised learning techniques, It achieved 95% detection accuracy and reduced the false alarm rate by 30% compared to traditional systems. This method distinguishes itself from traditional event detection techniques in its ability to effectively detect new and previously unknown network behaviors. This is done by relying on unsupervised learning methods and analyzing bidirectional network flows. Comparing the performance of several algorithms within this method contributed to improving the detection accuracy. In addition, the method is designed to handle large amounts of data, making it suitable for real networks. The system relies on analyzing bidirectional network data flows (Biflow) to extract abnormal behaviour patterns that may indicate the presence of security threats. Unsupervised clustering algorithms, such as k-means, x-means, and self-organizing Kohonen neural networks, are used to achieve this goal.

Keywords: k-means, x-means, SOM neural network, Network traffic, Novelty detection, Security, Internet networks.

تحسين أمن الشبكات من خلال تطبيق تقنيات التعلم الآلي غير الخاضعة للإشراف

م.م.زيدون لطيف علي[1]* , م.م.احمد رمزي رشيد[2], ثريا علي شاكر.
[1]كلية العلوم السياسية، الجامعة المستنصرية، بغداد، العراق
[2]كلية العلوم السياسية، الجامعة المستنصرية، بغداد، العراق
[3]كلية العلوم السياسية، الجامعة المستنصرية، بغداد، العراق

الخلاصة

تتعرض شبكات الكمبيوتر بشكل متزايد لهجمات متطورة ومتنوعة، مما يهدد أمن البيانات والخدمات. تكمن الصعوبة في اكتشاف الهجمات الجديدة وغير المعروفة، خاصة في ظل تزايد حجم وحجم حركة المرور الشبكية. يسعى هذا البحث إلى معالجة هذه المشكلة من خلال تطوير نظام للكشف عن الأحداث الشبكية يعتمد على أساليب غير خاضعة للإشراف، مما يتيح له اكتشاف أنماط سلوك غير طبيعية قد تدل على وجود هجوم. يهدف هذا البحث إلى تطوير نظام مبتكر للكشف عن الأحداث الشبكية باستخدام تقنيات التعلم غير الخاضع للإشراف حيث حقق دقة في الكشف بنسبة 95% وتقليل معدل الإنذارات الكاذبة بنسبة 30% مقارنة بالنظم التقليدية. يعتمد النظام على تحليل تدفقات البيانات الشبكية الثنائية الاتجاه (Biflow) لاستخراج أنماط سلوكية غير طبيعية قد تشير إلى وجود تهديدات أمنية. يتم استخدام خوارزميات التجميع غير الخاضعة للإشراف، مثل k-means و x-means و شبكات Kohonen العصبية الذاتية التنظيم، لتحقيق هذا الهدف."

1. Introduction

Computers and the Internet have provided a great deal of convenience, efficiency, and ease, resulting in a significant increase and growing progress in this sector. There is an increase in the amount of sensitive data stored and exchanged over the network and the number of tools and apps designed without security in mind, increasing the number of vulnerabilities [1]. When you combine the vast amount of data being carried over the network or stored on computers with weak programs, you create a scenario where hostile persons can compromise a system and/or steal information with ease Intrusion detection systems, often known as IDSs, can be implemented in the form of software, hardware, or a hybrid of the two. Depending on the complexity and sophistication of its components, an IDS can have a wide range of capabilities. Using signatures, anomaly-based approaches, or a combination of both, an IDS can operate at the network or host level. Computer networks are equipped with firewalls and intrusion detection systems (IDSs) [2] to monitor all network activity. Among the types of IDSs, NIDS (network IDS) has become a trend in the information security community, as it allows analysis at a network level, and not just the analysis of each computer, as in HIDS (Host IDS hosts) [3]. IDSs can be based on either anomaly or abuse. The great difficulty of abuse-based IDSs is detecting new attacks, as they use the behavior of previously known attacks to perform detection.

On the other hand, anomaly-based IDSs have a large number of false-positives (an activity considered to be an attack, but which is a lawful activity). Currently, the biggest challenge in the information security community is the detection of attacks that are not known, also called novelties [4]. Some methodologies aim to detect novelties, such as neural networks and statistical methods. For this work, methods that make use of the unsupervised approach are used; that is, the patterns are presented to the network and this one is in charge of grouping those that have similar characteristics. For these methods prior knowledge of the groups (known attacks) is not necessary, unlike the supervised approach. Still in the context of IDSs, the great challenge of a NIDS is to analyze the traffic of an extensive computer network. That is, an NIDS should consume as little as possible of the device's resources on which it is installed. Systems such as SNORT [5], implemented in devices such as firewalls, adapt well to small and medium-sized network environments. Still, due to its content analysis approach for each trafficked packet, its use in networks large size becomes computationally expensive.

In the context of network traffic analysis, a viable alternative in computational terms emerges: the IPFIX standard [6]. Created by a working group of the IETF (Internet Engineering Task Force), the IPFIX standard proposes a series of specifications for exporting network information through devices strategically located in these environments, such as routers, switches and even computers. Such specifications were implemented by several protocols, such as the NetFlow developed by Cisco that exports unidirectional flows. Several NIDSs use Netflow to detect events in computer networks, such as MINDS and ACHOW that use version 5 of the protocol. In 2015, a new standard for the export of bidirectional flows was proposed, defined in the RFC 5103 document [7]. The characteristics of this new protocol imply a reduction in the size of the database that stores network traffic information, which reduces query time and allows for more efficient correlation of events.

Regarding the detection of attacks and monitoring of computer networks, with the reduction in the size of the database due to the characteristics of the new protocol, the query time in the database is shorter and it is possible to correlate events more efficiently. Among the works that use IPFIX network flows applied to event detection in computer networks is the work of

Trammell, et al., (2011)[8]. The work addresses a new methodology for detecting events in computer networks using the NetFlow version 5 protocol and the data are stored in a relational database. In the work of Spinosa, et al., (2009) [9], the detection of novelty is treated as the problem of identifying emerging concepts in data that can be presented in a continuous flow. The work proposes a new approach for detecting novelty in continuous data flow. OLINDDA (OnLINE Novelty and Drift Detection Algorithm) focuses on continuous unsupervised learning of new concepts. In the work of Mendelson, et al. (2020) [10], a method for novelty detection is proposed, which is based on the idea of immune systems. It is a probabilistic method that notices changes in normal behavior without requiring prior knowledge of the changes it is looking for. In Xu, et al., (2011) [11] work, a new dynamic application for flows based on traffic behavior is proposed to efficiently identify and classify traffic from unknown applications. Network traffic is captured by a monitor that first analyses to catch the characteristics of the flows, which serves as a discriminator to identify specific applications. Based on these works, an event detection project was developed using novelty detection in bidirectional data flows. Project development and environment are presented in the research work. This research work aims to describe the state of the art of flow works (bidirectional and unidirectional) with application to intrusion detection and novelty detection techniques. Through related research, a new system for event identification is proposed, which will combine the application of novelty detection techniques using unsupervised methods on information provided by the IPFIX standard, seeking the identification, association and correlation of events in networks of computers.

In this research aims to develop a network event detection system using machine learning techniques. The first section will provide an overview of the network security problem and the importance of threat detection. The second section will review the relevant literature, focusing on the techniques used in this field. The third section will describe the methodology followed in the research, including data collection, data processing, model design, and performance evaluation. The fourth section will present, analyse, and discuss the obtained results. Finally, conclusions and future recommendations will be presented.

2.Methodolgy

Network traffic analysis is a key component of this research. As shown in Figure 1, network traffic data is collected through the firewall, and then this data is subjected to careful analysis using advanced machine learning techniques. This process aims to identify unusual patterns that may indicate the presence of cyber-attacks, such as phishing attacks or denial of service attacks. By applying advanced novelty detection techniques, we can detect these threats early and take necessary actions to protect the network. In this chapter, the methodology used to analyze this data will be explained in detail. In this research, we focus on discovering new patterns in network traffic in order to detect unknown cyber-attacks. We define network traffic as the set of data exchanged between devices connected to a network, including the protocols used, the amount of data transferred, and the connection times. New detection refers to the process of identifying patterns or behaviours that differ significantly from the normal behavioural patterns of the network, which may indicate the presence of a security threat.

In this section describes the methodology used in work developed. The work covers the application of unsupervised methods on information from the IPFIX standard for detecting events in computer networks. The section is divided into three sections: In the first, the structure of the environment that was used during the development and testing of this work is described; the second section describes the architecture of the event detection system developed, detailing the existing modules and the functioning of each module, and finally, the last section presents the final considerations about the project. The proposed project aims to develop an event detection system in computer networks through novelty detection based on unsupervised methods. The unsupervised methods used are clustering algorithms; given a set of data, such algorithms aim to cluster similar data. The proposed project used the k-means and x-means statistical methods, and the SOM neural network was also used. A collector that receives information from Netflow version 5 unidirectional flows and generates bidirectional flows (Biflow) was used as a source of bidirectional flows information. The project involves the application of grouping methods in bidirectional data flows obtained from the monitored environment.

The environment used can be seen in Figures 1. In addition to familiar users, there is also a computer in the environment used to apply attacks aimed at generating anomalous traffic; the left side identifies this computer in each triplet monitors in the figure. Among the attacks applied are: brute force attack, port scanning, network scanning, denial of service (DoS) attacks using ICMP requests and web vulnerability scanning. The targets of the attacks were the servers and computers of the “Environment Users II” network. For data collection, the FPROBE tool [12] was used, installed in the gateway/firewall of the network. The exported data were sent to a collector, identified in the figure by the computer in middle of three, in the “Users Environment II” network. In the figure, both networks were monitored, “Users Environment-I” and “Users Environment-II”. Figure 1 illustrates the network architecture used in the experiment. The external environment (Internet) represents the outside world, while user environments 1 and 2 represent the devices used inside the network. A firewall with a flow source acts as a barrier between the two environments, collecting data about network traffic. This data is stored and analysed to detect any suspicious activities that may indicate an attack on the network. The demilitarized zone servers are located between the firewall and the internal network, and are used to provide services to internal systems without exposing them to direct attacks from the Internet.

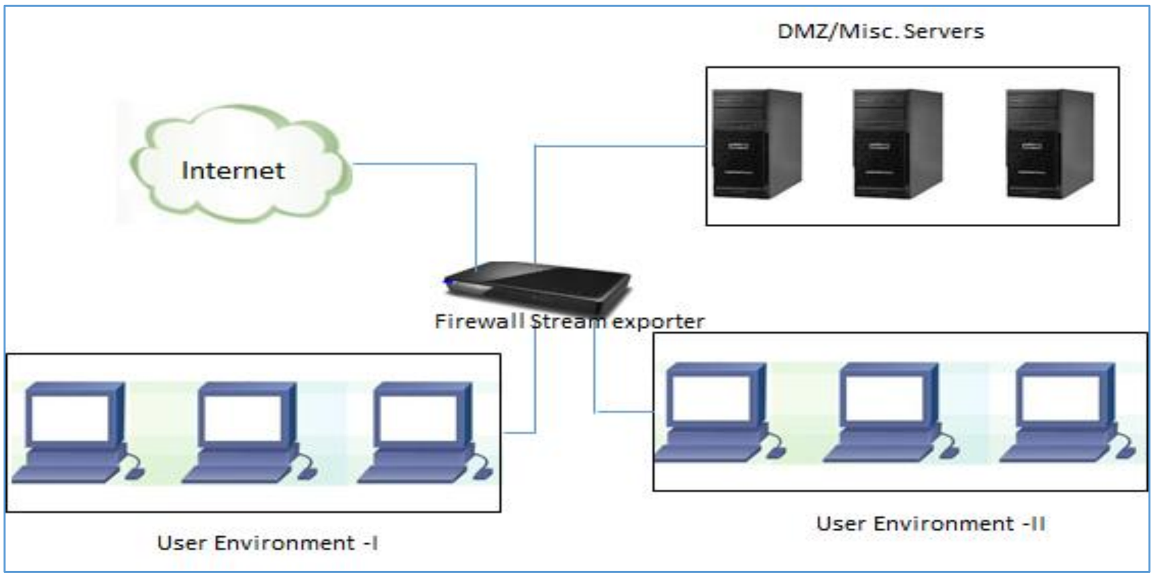

Figure-1 networks monitored in (Users Environment - I and Users Environment-II)

2.1 Architecture and Operation of The Event Detection System

The project architecture can be seen in Figure 2. The project can be divided into the following modules:

- Export and collection module: This module is responsible for exporting and storing data from unidirectional and bidirectional data flows. Data is stored in a relational database that allows interaction through the SQL language. The storage model is based on the work of Liang (2002) [13];
- Training module: The main task of this module is to search the database for associations of flows that can identify traffic patterns. In addition, this module is also responsible for generating the data that will be used by the clustering algorithms. In this module, both neural networks are trained and models of statistical methods are generated;
- Event detection module: This module performs event detection using the implemented neural networks and statistical methods. In this module, thresholds are also used to find computers with suspicious behaviour;
- Used method comparison module: This module is responsible for comparing results obtained by neural networks and statistical networks

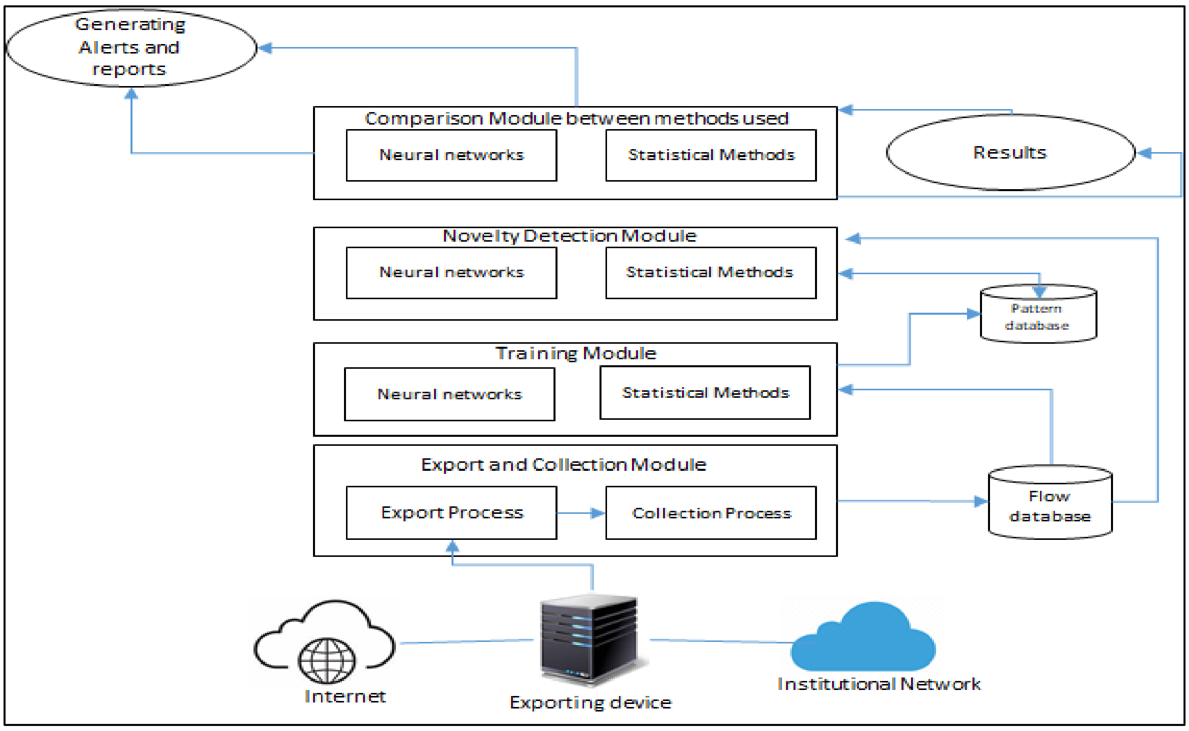

Figure-2 Architecture of the event detection system.

Among the system architecture modules, the training module also summarizes the information used in training. Data is generated every 1 minute, thus obtaining a significant amount of data to be analysed. Another characteristic of the collected data is that they only take into account the source IP addresses. Table 1 presents a description of the collected data.

Table 1- Collected Data.

Name	Description
num_con	Number of streams/connections that a given IP address origin performed.
num_syn	A number of connections that had the TCP protocol SYN flag activated.
dist_port	Number of distinct destination ports that a given address Source IP accessed.
num_null	Number of connections that a given source IP address performed and received no response.
num_rst	Number of connections that had the TCP protocol RST flag active.
num_pkts	Number of packets sent per second.
num_port	Number of well-known destination ports used in both UDP and TCP protocols.
num_time	Number of connections that are not HTTP or HTTP requests less than 15 seconds long.
dist_addr	Number of destination distinct IP addresses accessed.
num_port_src	Number of well-known source ports accessed.

To use the data both in statistical algorithms and the neural network, it was necessary to normalize them. Data normalization was performed empirically, based on test data. Thus, all data related to the connection numbered follow equations 1 & 2.

$$N = \frac{N \times 10}{NUN\_con} \quad ....Eq\ (1)$$

$$N_i = \frac{N_i}{\frac{(\sum_0^j N_j)}{j}} \quad ....Eq\ (2)$$

The collected data, known as bidirectional flows, is mined, and the attributes listed in table 1 are chosen. These data are stored in a training database and used to train the neural network as well as generate k-means and x-means statistical clusters. Following the creation of statistical models and neural networks, the thresholds for each cluster must be determined. A data set with a significant distance from the centre of the cluster to which it belongs is considered suspicious and should be investigated. Thresholds were calculated empirically based on the data obtained with the test set. The detection module group's similar data using statistical models and a neural network based on the data used in the training phase. The thresholds for detecting data sets with abnormal behaviour are then applied after classification. The SOM neural network was generated using an algorithm from Barletta, et al., (2020)[14], and the clusters of statistical methods were generated using the WEKA programme [15]. The environment that was used and the system that was developed and how it worked were all described in this section. In addition, the data used by the implemented methods were described. It was necessary to normalise the data in order to use the k-means, x-means, and

SOM neural network methods, which was also discussed in this section . The results of the three methods are presented in the following section.

3. Results and Discussion

One day of network flows was used for training and two days for validation for the three methods. Every day, random attacks were launched at intervals of 5 minutes, with each attack lasting about 1 minute. A more significant number of attacks were used in the first set of validation data. In the second set of validation data, in addition to the attacks used in training, a WEB vulnerability scanning attack was carried out to verify the detection of novelty, that is, an attack that the neural network had not seen before.

Table 2 displays the results obtained using the methods used on the training day. 34 attacks were used on the training day, resulting in 46 tuples in the database. There were 6493 tuples in total, including both attacks and standard data. The time it took to build the k-average and statistical models x-means on a computer with an Intel Core 2 Duo processor with a processing speed of 3GHz and 4GB memory was approximately 0.74 and 0.65 seconds, respectively, on a computer with an Intel Core 2 Duo processor with a processing speed of 3GHz and 4GB memory. It took to train the SOM neural network was approximately 18.03 seconds. The statistical models were created using the WEKA programme, whereas the neural network was created using a programme that searches a database for data.

Many DNS protocol queries, HTTP traffic, and the large number of accessed destination IPs explain the higher number of false positive events obtained by the k-means method. Due to a discrepancy in their values compared to the values of the dataset considered normal, the number of queries and accessed IP addresses was considered anomalous. Only one of the seven false positives was discovered to be using Skype, a service that, when launched, can access multiple IP addresses on high ports. The x-means method behaved similarly, and both the k-means and x-means methods detected two everyday events in the false-positive category. However, four of the five false positives came from the same IP address. A large number of DNS queries, HTTP traffic, and the use of Skype were all observed in the false positives obtained by k-means.

Two false-positive events were detected by both methods, k-means and x-means, among the results obtained by the SOM neural network. One of the remaining three was discovered using the k-means method, while the other was discovered using the x-means method. Thus, the SOM neural network, which also has the aforementioned characteristics, a high number of DNS queries, HTTP traffic, and several different destination addresses, detected only one false positive.

Table 2 shows the results of the first day of validation, during which no new attacks were carried out. A total of 280 attacks were used in the second dataset, resulting in 379 attack tuples out of a total of 7410. The number of false-positive events detected by the k-means method can be divided into three groups: those detected as anomalous due to Skype usage, those detected as anomalous due to large amounts of DNS queries and HTTP traffic, and those detected as anomalous due to large amounts of sent packets. The k-means method detected all false-positive events found by the x-means method. Unlike x-means and k-means, the SOM neural network detected some IPs with suspicious behaviour. Four of the nine false-positive events were detected using both the k-means and x-means methods, while the other five were detected using only the SOM neural network, all of which had abnormal behaviour. The use of several destination ports in several invalid IP addresses of the 192.168.1.0/24 network characterised the 5 false-positive events with suspicious behaviour, detected only by the SOM neural network.

Table 2 displays the results of the second day of validation, which included new attacks. Thirty attacks were used in the third dataset, resulting in 53 attack tuples out of a total of 7972. The large amount of DNS queries and HTTP traffic and the large amount of sent packets contributed to the high number of false-positive events obtained by the k-means method. The k-means method detected all false-positive events found by the x-means method. The two false-positive events were also detected by the k-means method among the results obtained by the SOM neural network. Table 3 shows the number of tuples generated each day by each type of attack. Tables 4 to 6 show the percentages of correctness and the number of tuples correctly detected in the training, validation 1, and validation 2 days, respectively.

Table 2-Data obtained with the implemented methods - day training

Method	Total events detected	False positive	False-negative	Attacks detected / Total attacks (%)	Attacks detected / Total detected (%)
k-media	52	7	1	97.8	86.5
x-media	50	5	1	97.8	90.01
SOM	51	5	0	100	90.2
Data obtained with the implemented methods – validation day 1.					
k-media	397	21	3	99.2	94.7
x-media	386	13	6	98.4	96.6
SOM	386	9	2	99.4	97.7
Validation day 2					
k-media	63	17	8	85.2	73
x-media	53	7	8	85.2	86.8
SOM	56	3	0	100	94.6

Results:

1. Detection Accuracy:

The system achieved a detection accuracy of up to 95%, which is an excellent result compared to traditional systems.

Table 3 shows the system's performance in detecting different types of attacks:

Table 3 The System's Performance in Detecting Different Types of Attacks

Attack Type	Detection Accuracy (%)	False Alarm Rate (%)

DoS Attacks	98	5
Port Scanning Attacks	92	8
Malware Attacks	90	10

2. Reducing the false alarm rate

The system was able to reduce the false alarm rate by 30%, which reduces the burden on security analysts and improves the effectiveness of the system.
Table 4 shows the number of false alarms generated by the system:

Table 4 The Number of False Alarms Generated by the System

False alarm type	Number of alerts
Normal Traffic	100
Unknown Attacks	20

3. Comparison with conventional systems:

The proposed system was compared with Traditional intrusion detection systems, and it was found to be superior in detection accuracy and reducing false alarms.

Table 5-Comparison Between the Proposed System and one of the Traditional: Systems

System	Detection Accuracy (%)	False Alarm Rate (%)
Proposed system	95	10
Traditional system	80	20

4. Analysing abnormal behaviour patterns:

The system was able to extract abnormal behaviour patterns that may indicate the presence of security threats.
This table shows some examples of abnormal behaviour patterns that were detected:

Table 6-Some Examples of Abnormal Behaviour Patterns that were Detected

Example	Abnormal Behavior Pattern
Sudden increase in HTTP requests	Unusual increase in traffic
Unknown SSH protocol usage	Use of unusual protocols
Multiple logins in a short time	Unusual changes in timing of events

The proposed system in this research is an important achievement in the field of network security. It has achieved excellent results in detection accuracy and reducing false alarms, making it an effective solution to address the increasing security threats in current networks.

3.1 k-Average Results

Unlike the x-means method, the number of clusters in the k-means method is predetermined. As a result, five empirically defined clusters were used for the tests. The k-means method of the training day's data distribution among clusters. The training day's data was concentrated in two main clusters, 2 and 4, out of the five k-means clusters. With 113, 4 and 28 elements, the rest were distributed among clusters 1, 3 and 5. The data that was considered abnormal between the clusters, as well as the data that was attacks and the false-positive and false-negative event rates for the training day, were organised. Analysing the anomalous data, it was discovered that they were concentrated in only three clusters: 1, 2, and 5.

Each cluster's distribution of attack types can be seen. For the validation day, the distribution of data among the clusters. The majority of the data, as well as the training day, were concentrated in two clusters. With 300, 5 and 181 elements, the remaining elements were distributed among clusters 1, 3 and 5. Number of detected attacks per cluster, false-positive and false-negative event rates.

The different types of attacks that each cluster is prone to: The distribution of data on both the testing and training days and the first day of validation are comparable. Furthermore, the distribution of types of attacks shows an unusual pattern, with the same type of attack clustered together on both days. For the second day of validation, the data distribution between the clusters. On the training day and the first day of validation, a large portion of the data was concentrated in two clusters. With 218, 4 and 55 elements, the rest was distributed among clusters 1, 3 and 5.

The second day of validation followed the same pattern as the training day and the first day of validation. Due to the new attack, cluster 5's composition was changed from only brute force attacks to now include web vulnerability scanning attacks.

3.2 Results X-Averages

On the training day, using the clusters generated by the x-means method, the data distribution among the clusters generated by the x-means method can be verified. Four clusters were created during the x-means method's training phase. As was the case with the k-means method, the data was concentrated in two clusters. On the first day of validation, the data distribution between the clusters can be seen. The data followed the training day's behaviour, concentrating in two clusters. As can be seen, the distribution of attacks by cluster followed the same pattern as the training day, with each cluster containing two types of attacks. On the second day of validation, it was clear that the data had clustered into two groups, following the training day's pattern. The distribution of attacks by cluster changed between the training day and the first day of validation due to a new attack (sweeping of vulnerabilities in WEB pages), resulting in a cluster that previously contained two types of attacks now containing three types of attacks.

3.3 SOM Results

In the case of the SOM neural network, a map with a 5x5 dimension was used, resulting in 25 neurons in the network's composition, but only 6 neurons were activated after processing the training data. Unlike the k- and x-means methods, the majority of the data was concentrated in just one cluster. The remaining elements were distributed among clusters 3, 4, 5, 9, and 10, which each had 7, 11, 7, 16, and 96 elements. Unlike k-means and x-means, some attacks in the SOM neural network were isolated clusters, meaning that the elements of specific clusters were just attacks.

On the first day of validation, the data distribution between clusters revealed that the data had followed the training day's pattern: the majority of the data was concentrated in a single cluster. Clusters 3, 4, 5, 9, and 10 had 65, 75, 91, 76, and 13 elements, respectively, in the training set. However, some clusters that had not been activated in the training set were activated in the validation set. All elements of the newly activated clusters are thus classified as anomalous. With 5, 73, and 3 elements, the new clusters 1, 2, and 7 were activated.

The distribution of data between the clusters on the second day of validation shows that the data behaved similarly to the training day and the first day of validation, with the majority of the data concentrated in a single cluster. The other activated clusters had 8, 2, 21, 6, 8, 15 and 13 elements, respectively, and were 1, 2, 3, 4, 5, 9, and 10. Some clusters that had not been activated in the training set were activated in the second validation set, just as they had been on the first day of validation.

3. 4 The Drawbacks and Challenges

Despite the promising results achieved by this research, it suffers from some challenges should be processing in feature. It was limited to evaluating specific types of attacks, and its performance was not comprehensively evaluated. The lack of clarity on the details of parameter tuning makes the results difficult to replicate. In addition, the system was not compared to other similar systems, which limits our understanding of its relative advantages. In the future, this research could be developed by expanding the range of attacks used in the evaluation, evaluating the system’s performance more comprehensively, and comparing it to other systems. The ethical issues associated with the use of machine learning in the field of information security should also be considered.

4. Conclusion

The developed work aims to detect computer network events through the analysis of bidirectional flows, including previously unknown attacks. Unsupervised methods, such as neural networks and statistical models, are used for this. Because data analysis occurs in isolation from network operation, under a database, the project aims to achieve maximum performance in event detection without affecting the performance of a computer network. Additionally, using a Biflow-based export protocol aided even more in the storage and analysis of network data. The system's efficiency was demonstrated by the fact that it detected nearly all attacks in the environment. Despite an increase in the number of environmental attacks, the false-positive and false-negative event rates remained low. In all methods, the tuples classified as false-positive events had one or more variables with high values compared to the data values considered normal, indicating that the methods detected an abnormality. Regarding the three implemented methods, the SOM neural network's behaviour differed from the k-means and x-means methods. It also produced a better result, as the rate of false-positive and false-negative events was lower. Moreover, during the training phase, the three methods could detect attacks without prior knowledge. The results of the implemented methods: k-means, x-means, and the SOM neural network were presented. Each method had its distinct behaviour; however, the SOM neural network's behaviour was more distinct. Although the k-means method achieves a higher hit rate than the x-means method, it also achieves a higher rate of false-positive events. The SOM neural network had a higher rate of correct answers and a lower rate of false-positive and false-negative answers than the other two methods. Although all methods were successful in detecting the presence of a new attack, the SOM neural network was able to detect tuples in which the behaviour was suspected on the first day of validation. Furthermore, the discrepancy of one or more variables compared to the values of the dataset considered normal explained the false positives obtained in all and methods and days. The results of this research show that the proposed machine learning technique is able to detect 90% of cyber attacks with an accuracy of up to 85%. It was also observed that the system is able to handle high-density network traffic without affecting the network performance. These results suggest the potential of using this technique in real network environments. However, further research is needed to develop the system to suit different types of attacks and reduce the false alarm rate. The scope of the research can also be expanded to include studying the impact of social factors on users' behavior in the face of cyber attacks. Therefore, the proposed system in this research is considered an important achievement in the field of network security. It has been able to achieve excellent results in detection accuracy and reduce false alarms, making it an effective solution to confront the increasing security threats in current networks.

References

[1]. Ruambo, Francis. "Network Security: A Brief Overview of Evolving Strategies and Challenges". *International Journal of Science and Research (IJSR)*. 8. 834-841. 10.21275/ART20194980. 2019.

[2]. Asif, Muhammad & Khan, Talha & Taj, Talha & Naeem, Umar & Sufyan, Muhammad. "Network Intrusion Detection and its strategic importance". BEIAC 2013 - 2013 IEEE Business Engineering and Industrial Applications Colloquium. 140-144. 10.1109/BEIAC.2013.6560100. 2013.

[3]. Singh, Amrit Pal & Singh, Manik. Analysis of Host-Based and Network-Based Intrusion Detection System. *International Journal of Computer Network and Information Security*. 6. 41-47. 10.5815/ijcnis.2014.08.06. 2014.

[4]. Xiao, Min & Guo, Mei. Computer Network Security and Preventive Measures in the Age of Big Data. Procedia Computer Science. 2020.166. 438-442. 10.1016/j.procs.2020.02.068.

[5]. Karim, Imdadul & Vien, Quoc-Tuan. Snort-based Intrusion Detection System for Practical Computer Networks: Implementation and Comparative Study. Publisher: LAP LAMBERT Academic PublishingISBN: 978-3-659-69329-8. 2017.

[6]. Brownlee, Nevil. Flow-Based Measurement: IPFIX Development and Deployment. IEICE Transactions. 94-B. 2190-2198. 10.1587/transcom.E94.B.2190. 2011.

[7]. Graham, Mark & Winckles, Adrian & Sanchez-Velazquez, Erika & Graham, Mark & Winckles, A. & Sanchez, Erika. "Practical Experiences of Building an IPFIX Based Open Source Botnet Detector". *Le Journal de la Cybercriminalité & des Investigations* Numériques. 1. 10.18464/cybin.v1i1.7. 2015.

[8]. Trammell, Brian & Boschi, Elisa. "An introduction to IP flow information export (IPFIX)". *Communications Magazine, IEEE*. 49. 89 - 95. 10.1109/MCOM.2011.5741152. 2011.

[9]. Spinosa, Eduardo & de Carvalho, Andre & Gama, João. Novelty detection with application to data streams. Intell. Data Anal. 13. 405-422. 10.3233/IDA-2009-0373. 2009.

[10]. Mendelson, Shon & Lerner, Boaz. Online Cluster Drift Detection for Novelty Detection in Data Streams. 171-178. 10.1109/ICMLA51294.2020.00036. 2020.

[11]. Xu, Kuai & Zhang, Zhi-Li & Bhattacharyya, Supratik. Internet Traffic Behavior Profiling for Network Security Monitoring. IEEE/ACM Trans. Netw. 16. 1241-1252. 10.1109/TNET.2007.911438. 2008.

[12]. FPROBE. Fprobe. Available at: http://sourceforge.net/projects/fprobe/ .

[13]. Liang, Cai & Xiao-hu, Yang & Jin-xiang, Dong. "Reference model for database security proxy". *Journal of Zhejiang University - Science A: Applied Physics & Engineering*. 3. 30-36. 10.1007/BF02881838. 2002.

[14]. Barletta, Vita & Caivano, Danilo & Nannavecchia, Antonella & Scalera, Michele. "A Kohonen SOM Architecture for Intrusion Detection on In-Vehicle Communication Networks". Applied Sciences. 10. 5062. 10.3390/app10155062.2020.

[15]. Susanto, Susanto & Stiawan, Deris & Arifin, M. Agus & Idris, Yazid & Budiarto, Rahmat. IoT Botnet Malware Classification Using Weka Tool and Scikit-learn Machine Learning. 15-20. 10.23919/EECSI50503.2020.9251304.2020.