Towards a user network profiling for internal security top-k rankings similarity measures using k-means

Cristian Salvador Najar Juárez

Abstract. Effective traffic control generally uses a variety of techniques for the classification, prediction, and monitoring of network traffic. This article proposes a method supported by the application of a technique to identify whether or not a network user is having normal behavior by analyzing host traffic using k-classification similarity measures. This article proposes an improvement of the estimation methodology of the K-means algorithm as an efficient solution oriented to cases with numerous groups and sizes [1]. The results of this study showed high accuracy using the joint approach with a group of 90 students from the Autonomous University of Guadalajara who were monitored for 30 minutes each using a proxy, demonstrating that the support vector classifier algorithm achieved 100% accuracy in this dataset. Once this information is obtained, the k-means algorithm is used to group users together and identify outliers that exceed the upper limit of the k range.

Keywords: machine learning, top-k, information security.

1. Introduction

Internet traffic has increased significantly in the last year, and there is no prognosis that this predisposition will change in the coming years [2]. This increase is almost fivefold due to the proliferation of new access network technologies. But the engine of this change is Internet users, whose habits and consumption behaviors of APPs have advanced over the years, directly affecting the demand for access to network traffic. This increase is almost fivefold due to the proliferation of new access network technologies. But the driver of this change is Internet users, whose application consumption habits and behaviors have evolved over the years, directly affecting the demand for access to network traffic [3].

This information is grouped and compared with other users to confirm that the information generated is as intended. Similarly, 90 students were analyzed and monitored for 30 minutes by each user with a proxy to stop any problems. This group was divided into 20 networks, 10 programmers, and 60 mechatronics students from the network operations center, and their network traffic was captured to create different user profiles, web pages, applications, and connections.

Network monitoring has become a critical issue in today's digital age, where organizations and individuals face a variety of threats from malicious actors looking to exploit vulnerabilities in networks and systems. Traditional methods of network monitoring, as well as retention of incoming traffic on the private network and intrusion detection systems (IDS), have been used for decades to protect networks from these threats, but they have some limitations. In recent years [2], machine learning (ML) has become a promising alternative to cybersecurity, capable of detecting and responding to threats more effectively than traditional methods.

The installation of the above tools can be done from a computer or from a cell phone and has several utilities according to the purpose pursued by the person who will manipulate the intelligence of this application. It is versatile because it has several versions according to the operating system on which it will be installed. The installation of it has certain specific requirements and configurations that must be made so that it can work accurately before the requisitions are monitored using its efficiency.

Once this information is obtained, several methods are used to select the best algorithm that will show if a user is generating unusual traffic. At the end of a user's session, all cases of highly unusual traffic will be displayed visually for easy identification and appropriate corrective actions taken, such as verifying the true identity of the user associated with that device or detecting malware or other content. In previous studies [3], the authors relied on analyzing the behavior of the traffic generated and comparing it with certain rules, such as firewalls or access lists.

The objective of this article is to apply a machine learning algorithm (k-means) using the top-k to accumulate data for user monitoring, One of its main tools is focused on checking the correct use of text on a web page or application, identifying URLs, and comparing with nearby or local domains to determine similarities, allow checking of similar names and APIs, making changes, monitoring and more.

For this article, the authors relied on Charles Proxy [4], a dedicated tool for web debugging and monitoring network traffic on some Windows Android and IOS devices. By using it, call network traffic can be monitored and decrypted. The data can be read when sent from different servers, which can also be read from different devices with Windows, Android, and iOS systems. The app requires installation, configuration, and payment after a limited free trial period.

With it, users can access URLs and even different links with the proper and custom download settings that allow them to use them for these monitoring features. The application is in English, so it is necessary to learn the language or translate it to understand the root, the origin of the configuration of the process to be carried out. It is necessary to create some filters to perform readings or activate the characteristics of the machine itself or the computer [4]. To configure it, users need to know some buttons that are necessary to activate the functions of the program: it has a button to delete, start, enable SSL Proxy, Internet connection bandwidth network limit, breakpoint to allow the visit or send the user back to the previous navigation point.

The Charles Proxy was chosen because the ability to read each request in plain text uses the free version, which imposes a usage time limit of 30 minutes. This is a popular tool used by developers and security professionals to debug and analyze network traffic. It allows users to intercept and inspect HTTP and HTTPS traffic between clients and servers, making it an invaluable tool for debugging and optimizing web applications, highlighting its ability to improve the performance of APPs, identify vulnerabilities, and improve the overall development process [4].

Once user traffic is captured and invalid auxiliary connections are avoided, the method used is data cleansing, as it is the main stage of data preprocessing aimed at identifying and correcting or eliminating errors and inconsistencies in the dataset [4]. It involves several steps, including identifying and handling missing data, identifying and handling outliers, transforming data to a standard format, identifying and correcting errors and inconsistencies, and identifying and removing duplicates.

The specific methods used in data cleansing will depend on the nature of the dataset and the research question being addressed. Proper data cleansing is essential to ensure that subsequent analysis is accurate and unbiased, such as ad pages, response codes, protocols, and TSL handshakes, but if speed, latency, bandwidth, etc. are required, users can add information to the dataset. This study will use the following data to group profiles within each application:

2 Method

For the collection of the data of the selected group, the Top-k technique was used, which implies the selection of the most relevant elements of the data set according to certain criteria. In other words, it is the process of identifying the most important or critical ones in a data set. This technique is commonly used in data analytics and machine learning applications that aim to identify and determine the most important features or data points [5]. The selection of top-k elements can be based on several criteria, such as the number of visits, likes, or other relevant metrics. This approach is invaluable in improving the accuracy and efficiency of data analysis by focusing on the most relevant and informative data points.

Unsupervised learning methods model or structure data without clear instructions or characteristics, which differ from supervised learning algorithms in that they require labeled training data to make predictions; unsupervised algorithms use unlabeled data and seek to discover hidden patterns, groups, or relationships between data [5]. It simply makes data models clearer so that user behavior can be collected before using learning techniques in conjunction to select the algorithm most accurately for their participation.

Data grouping is also used, whose technique is based on dividing a set of objects into groups where objects belonging to one group are very similar to each other but different from objects belonging to other groups. There are different kinds of algorithms for grouping data [6]. However, this paper focuses on one class of partitioning algorithms, specifically the standard K-means algorithm. K-means are widely used for many reasons, and their popularity lies in their ease of implementation for the current case of data that is getting bigger and is made more sequentially, causing the repetition and scalability challenge of K-means [6].

During this study, 90 users belonging to three different categories "Student", "Programmers" and "Networks" of the Autonomous University of Guadalajara were selected and each user generated 30 minutes of network traffic. Where certain similarities were identified between the traffic generated by each user based on the profile of each, the main objective of this study was to group the profiles and make an analysis of the traffic generated by each proxy due to the installation and group them according to the similarity. The research shows that some users may fall into different categories; for example, a developer profile may generate traffic similar to that of a student, indicating that the user may be working and studying.

With respect to the corporate environment, employees are typically restricted to a specific set of sites for work-related activities due to rules imposed by various firewalls. However, in the context of this study, students' learning takes place in a controlled network environment where they are free to generate different network requests. After being classified according to the most popular profiles, any user who deviates from the center may experience unexpected behaviors such as not logging in, getting infected with a virus, etc.

In the same order of ideas, the compilation of the flow generated by the students of the Autonomous University of Guadalajara was carried out with the participation of students from various areas, including software engineering, mechatronics, and biomedical. With an IP address of 24 masks, 254 users can be connected to VLAN 62 to capture traffic. On the other hand, the survey of web users and developers was conducted at a private banking firm, and IP address information could not be shared.

User	Equipment	Visual Studio
Networks 01	5	0
Programmer 01	0	3
Student 01	0	1
Networks 01	5	0
Programmer 01	0	3
Student 01	0	1
Networks 01	5	0
Programmer 01	0	3
Student 01	0	1

Table 1 Example for data collection. For the data set users can download the full data: https://pastebin.com/d08inJz3, students, developers, and networks are the users, while the teams and Visual Studio are the numbers of requests or interactions of the user had for 30 minutes. In the above file, the dataset can be downloaded where requests were made to services such as computers, overflow stack**,** SevOne, Packet Tracer, Spectrum, Service Now, Visual Studio**,** Udemy, Xcode, and GitHub.

The technique used was ensemble learning [7], which combines a multiple-learning model to optimize the accuracy and stability of the prediction. The result is as follows:

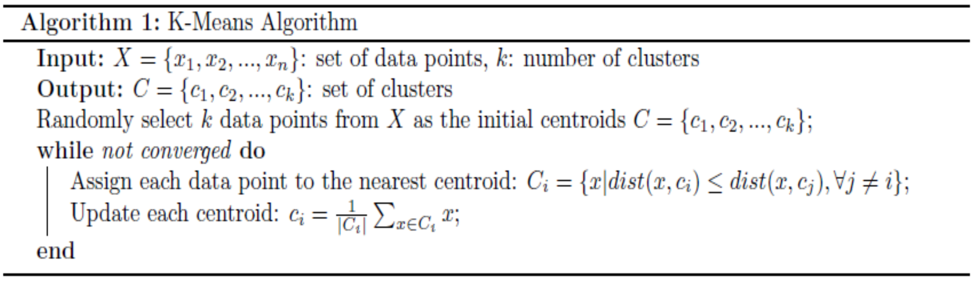
Acuraccy
Logistic regression 92%
Random forest classifier 92%
Decision tree classifier 92%
Multilayer perceptron 85%
Support Vector Classifier 100%

Tabla 2. Support Vector Classifier

This study uses a fusion learning approach, a machine learning technique that combines a multiple learning model to optimize prediction accuracy and performance. Co-learning has been shown to be effective when it is necessary to classify a data set into several classes. Because it improves the accuracy of the model as well as predicts values rather than categories, co-learning is advantageous when dealing with missing or incomplete data, as it reduces its impact on it. Also, if the dataset is bulky and difficult to process with a single model, co-learning helps break down the work into smaller models [7].

Another method used during the process was logistic regression, which is used to analyze and model the relationship between the dependent variable and one or more independent variables. This algorithm estimated the probability of an event based on the values of the independent variables. After performing the set method, it is decided to use the SVC algorithm, that is, a supervised learning algorithm designed to find the hyperplane that best discriminates between different classes for the analysis of data in high-dimensional feature spaces using support vectors.

Regarding the K-means algorithm is a clustering technique widely used in machine learning that aims to divide the set of data points into k groups, where k is a predefined number. The algorithm works according to the iterative assignment of each data point to the nearest cluster center after the centroid is updated, which will depend on the new task.



```
Algorithm 1: K-Means Algorithm
Input: $X = \{x_1, x_2, ..., x_n\}$: set of data points, $k$: number of clusters
Output: $C = \{c_1, c_2, ..., c_k\}$: set of clusters
Randomly select $k$ data points from $X$ as the initial centroids $C = \{c_1, c_2, ..., c_k\}$;
while not converged do
    Assign each data point to the nearest centroid: $C_i = \{x | dist(x, c_i) \leq dist(x, c_j), \forall j \neq i\}$;
    Update each centroid: $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$;
end
```

Figure 1. Example of k-means algorithm.

3 Results

According to the procedure performed, new values are randomly added to the data frame, and these values can have normal or abnormal behavior, indicating whether they are outside the upper k range [2]. In addition, the graphical display can be used to identify the specific supplier responsible for such anomalies. The values used were the interquartile range (IQR) as a measure of dispersion. The IQR shall be defined as the difference between the third quartile (Q3) and the first quartile (Q1) of the ordered data.

The 1st quartile, Q1, is the value that will divide the ordered data into 2 equal parts, so that 25% of the data is below Q1 and 75% is above. The 3rd quartile, Q3, is the value that will divide the ordered data into 2 equal parts, so that 75% of the data is below Q3 and 25% is above. Therefore, the interquartile range is calculated as:

$$IQR = Q3 - Q1 \qquad (1)$$

Equivalently, the interquartile range is the region between the 75th and 25th percentiles (75 − 25 = 50% of the data).

Later, the results are shown by category of the anomalous values, where the random results added 7 new users and 3 anomalous values, due to a function in Python, this number of users was randomly generated.

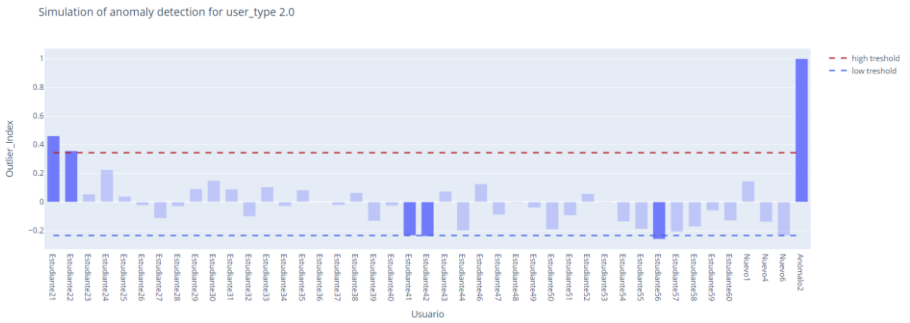


Figure 2. Simulation of the student's anomalous detection group.

An unusual data point is generated in the student category that is detected by the category. Anomaly 2 generates completely different traffic than the student category, indicating a completely different traffic generation pattern. The algorithm compares the data with new data or the user's k-value and displays the results for the specified group after completion.

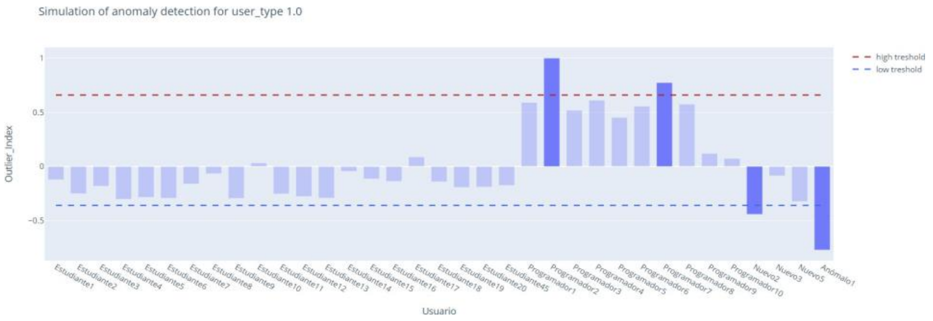


Figure 3. Simulation of the programmer of the anomalous detection group.

After analysis, it was found that Student 2 and Programmer 7 had a high threshold for abnormal traffic generation in their respective groups. This indicates that these users are generating abnormal traffic compared to the group average. On the contrary, outlier 1 did not generate the same traffic as the average of the group, and these results can give different conclusions because anomalous users can be classified into different profiles; for example, programmers can also be developers. However, these results will help the researchers to validate a controlled environment on the network.

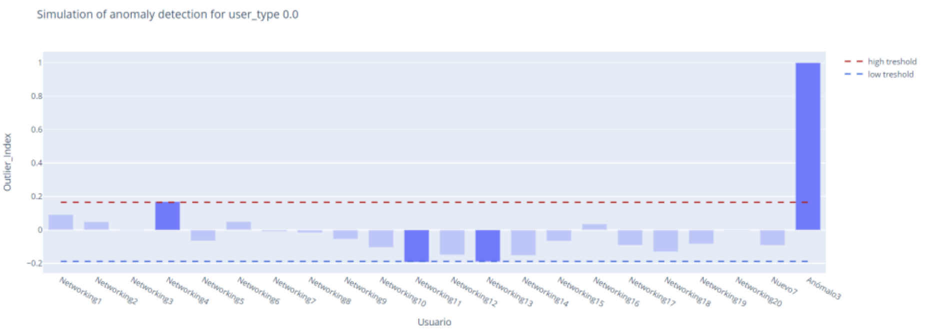


Figure 4. Simulation of Anomaly's detection group 0.

In group 0, among the mayority network users, an unusual user was found that generated a complete different traffic from the other group users.This user should be investigated as soon as possible to determine the cause of the unusual traffic pattern.

Conclusions

Artificial intelligence contributes significantly to improving the methods used in the process of controlling data network traffic in the areas of traffic classification and prediction, network security, and route optimization. This is due to the advantages of the machine and deep learning algorithms (speed, accuracy, self-learning, etc.), which allow the network management system to work more autonomously, actively, and efficiently, allowing the network to operate in an optimal state and thus optimizing the user's practice and the quality of the service provided.

On the other hand, there is a clear trend to implement deep learning algorithms compared to traditional machine learning algorithms since the former require less preparation of the data retrieved from the network to create a classification model and a process for identifying a user. The most commonly used algorithms based on neural networks stand out for their powerful ability to process information in parallel and with great precision.

Currently, this method allows users to determine expected traffic and user behavior, if applicable. Another user has differences and can be classified as uncollected or infected, which is the goal of this study. For a better understanding, a complete top-k should be generated, that is, capture user traffic to obtain a better multiple and add more categories to determine the correct behavior in the network according to the profile, It is recommended to use a neural network that is used to predict the best users [2].

The approach confirmed that the evaluation of the data was accurate. However, to achieve greater accuracy, a greater amount of data needs to be generated to evaluate the methods used. So far, three types of profiles (learner, network, and scheduler) have been used to obtain a k-value that represents the traffic generated by each profile. Once this was done, the types of variables can be identified, or top-k that are affecting the groups to detect anomalous users. In the next version, the researchers aim to generate a greater variety of profiles and a greater number of users.

It is also clear that, regardless of the algorithm used, the stage for classifying must be well constituted and projected in relation to the specific problem, since the effectiveness of the algorithm and/or the accuracy of the results will depend on this phase. Finally, it should be noted that, although the information processing mechanism has been optimized and progress has been made in improving accuracy and decreasing the error rate, there are still a few errors that require human supervision in some scenarios.

The principal component analysis algorithm (PCA) is a technique for reducing dimensionality used to simplify high-dimensional complex data. PCA is used to find linear relationships between input variables and reduce the number of variables used to model the data. PCA is a very useful data analysis method because it reduces the complexity of the data without losing too much information. Therefore, users can improve model accuracy by reducing data noise and eliminating duplicate variables. With PCA, the data is normalized, and the covariance matrix of the data is calculated to understand the relationship between the different variables and find the main devices that will explain most of the variance of the data.

Recognition

Thank you Dr. Luis Julián Domínguez Pérez for being my thesis advisor. The authors thank Dr. Pedro Álvaro Parrés Peredo, Iván Piza Dávila, and Francisco Cervantes, for allowing me to contribute to this research. Thanks to Conacyt for their opportunity and support with the scholarship. I want to thank my wife, for her patience and company on this journey.

References

1. Wood P. ISTR Internet Security Threat Report. Symantec. [En línea].; 2016.
2. Fagin R, Kumar R, Sivakumar D. Comparando las listas principales k. SIAM Journal on Discrete Mathematics. 17 (1). págs. 134–160. [En línea].; 2003.
3. Singh R, Kumar H, Singla R. Un sistema de detección de intrusos utilizando perfiles de tráfico de red y máquina de aprendizaje extremo secuencial en línea. Expert Systems with Applications. 42 (22). p 8609-8624. [Online].; 2015.
4. Randow K. Charles Proxy. [En línea].; 2002. Disponible y: https://www.charlesproxy.com/.
5. Kihl M, Dling P, Lagerstedt C, Aurelius A. Análisis de tráfico y caracterización del comportamiento del usuario de Internet. Congreso Internacional sobre Telecomunicaciones Ultramodernas y Sistemas de Control y Talleres (ICUMT). Moscú. p 224–231. [Online].; 2010.
6. Pérez J, Hidalgo M, Castro N, Pazos R, Díaz O, Olivares V, et al. Una heur´ıstica eficiente aplicada al algoritmo K-means para el agrupamiento de grandes instancias altamente agrupadas. Computación y Sistemas. 22 (2). p 607–619. [Online].; 2018. Disponible en: https://www.scielo.org.mx/pdf/cys/v22n2/1405-5546-cys-22-02-607.pdf.
7. Bonaccorso G. Dominar algoritmos de aprendizaje automático. Packt Publishing. [Online].; 2020.