Journal of Artificial Intelligence and Computing Applications (2024) 2(2): 48–59

MAIKRON

Journal of Artificial Intelligence and Computing Applications

Research Article

An Explainable Clustering-Based Approach for Cyber Situational Awareness on Masquerade Attacks Detection

Nelva N. Almanza-Ortega[1], Joaquin Perez-Ortega[1], Sergio M. Martinez-Monterrubio[2], and Juan A. Recio-Garcia[3,*]

[1]National Technological Institute of Mexico, México
[2]International University of La Rioja, Spain
[3]University Complutense of Madrid, Spain

ABSTRACT

Masquerade attacks pose a significant challenge in cybersecurity, as intruders mimic legitimate user behavior to evade detection. In dynamic, data-intensive environments, traditional intrusion detection systems often struggle to provide both timely and interpretable results, limiting their usefulness for effective Cyber Situational Awareness (CSA). This article presents a clustering-based approach for detecting masquerade attacks using OK-Means—a variant of K-Means optimized for faster convergence—combined with a nearest neighbor classifier and noise reduction techniques. The proposed Intrusion Detection System (IDS) reduces computational overhead while enhancing explainability, leading to more reliable and transparent Cyber Threat Intelligence (CTI) decisions.

Keywords: cyber situational awareness (CSA), masquerade attack detection, explainable machine learning

1. Introduction

In the last three decades, humanity has witnessed a revolution in Information Technology (IT), that has liberated the need for a digital transformation in all economic sectors in cyberspace. It has become the mainstay of growth and prosperity in the world economy. As a result, different nations have been forced to implement legislation and regulations to protect their cyber assets and digital markets. Nowadays, it is mandatory to become aware of the cyber situation for the proper performance of command and control tasks. Cyber situational awareness (CSA) contributes to mission-centric Cyber Threat Intelligence (CTI) providing support for informed decisions required to maintain a safe and secure IT environment. According to [1], "situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and projection of their status." However, the dramatic increase in IT complexity and the large amount of raw data generated by such systems complicates the correct implementation of this model.

Current approaches for CTI from the point of view of CSA are based on Machine Learning (ML) techniques, like decision trees, support vector machines, neural networks, etc. [2]. These methods are based on the generation of a predictive model able to identify and classify threats. However, they have two significant drawbacks. Firstly, as CSA applies explicitly to dynamic environments, the ML predictive models must be re-trained as the complexity of IT systems and their

E-mail address: jareciog@fdi.ucm.es
https://doi.org/10.5281/zenodo.15098302

associated risks evolve. Secondly, most of these ML algorithms are not suitable for big-data scenarios where a continuous and massive data stream provides the information to be analyzed.

The problem of applying static predictive models can be addressed through the use of lazy-learning ML methods such a the k nearest-neighbors (k-NN). This method does not require a precomputed model because all the computation is deferred until classification. Moreover, k-NN has the advantage that classification is approximated locally, and therefore, only a subset of the data should be required in order to obtain the prediction. This makes, k-NN the right approach for processing big data streams if we can select the local subset of data required to classify a potential threat. However, as k-NN is an instance-based approach, it will obtain a poor performance if we compare a new input sample to the whole large-scale datasets of collected information. In this sense, its combination with clustering techniques stands as a possible alternative to improve its applicability in big data scenarios at the expense of lowering the performance of the classification. The primary assumption is that the k-nearest neighbors of a threat lie in the same cluster. Thus, the k-NN search can be efficiently performed in two steps: (1) reaching the nearest clusters; and (2) finding the k-nearest neighbors within the selected clusters. In a big data scenario, this would eventually save a massive amount of instance comparisons.

Typically, clustering methods such as K-Means take a considerable time to compute clusters. This is not a problem in static environments. However, in the heterogeneous and dynamic environments where CSA is required, it is an additional issue to be addressed: as the environment evolves, clusters must be recomputed to avoid the loss of performance. To address this problem, in this work we present an efficient approach for the detection of suspicious events based on the combination of k-NN and a novel clustering strategy, OK-Means [3], that decreases the cost of recomputing clusters as the environment evolves. Concretely, we will demonstrate the benefits of our approach for Masquerade Detection. The objective of such systems is to raise an alert when computer behavior differs to a certain extent from standard computer behavior, as profiled from a history of computer sessions. As the amount of information to be potentially logged is vast (user actions, files accesses, etc.), this kind of detection requires methods able to manage big data.

OK-Means is specifically useful in big data realms as it uses a criterion to balance the processing time and the solution quality when the number of instances is significant. Instead of trying to improve the initialization or classification steps as the majority of the known strategies aimed to improve the performance of k-mean, this algorithm applies in the convergence step. This way, it achieves a decrease in computing time of about a factor 4/100, yielding solutions whose quality reduces by less than 2%.

As we will present in the results of this paper, our experimental evaluations show that the combination of OK-Means and k-NN decreases the computing time without a significant quality loss when applied to the detection of treads in a CSA scenario. Moreover, the second major contribution of this paper is the proposal of explanatory strategies to improve the CSA by allowing the cybersecurity analyst to understand the outcomes of the intrusion detection system. Explainable AI (XAI) is nowadays a significant research challenge and is driven by the evidence that many AI applications lack trust on behalf of their users. The running hypothesis is that by building more transparent, interpretable, or explainable systems, users will be better equipped to understand and therefore trust the intelligent system [4]. This hypothesis can be directly extrapolated to CSA, where black-box machine learning algorithms such as neural networks or Bayesian networks are commonly applied for intrusion detection. Here, the CTI is limited by the lack of transparency of the CSA system as the cybersecurity analyst is not able to obtain a clear perception of the causes that led to an intrusion alert. Therefore, we have chosen k-NN and clustering not only because they are able to deal with big data, but also because they are white-box methods that can be introspected to provide explanations about the causes of a potential threat.

The paper runs as follows. Section 2 presents the related work. Section 3 describes our approach based on the application of OK-Means to large datasets and its combination with a k-NN classifier. The experimental evaluation and associated results is presented and discussed in Section 4 and 5. Section 6 presents visual explanatory strategies and finally, Section 7 concludes the paper.

2. Background

The Computer Emergency Response Team (CERT) defines malicious insider as "an employee, contractor, or business partner who has or had authorized access to an organization's network, system or data, and intentionally exceeded or misused that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information system" [6].

The importance of the above definition is that it delimits the term insider to a particular context since if only the definition of this concept is taken, the notion of belonging to a group could lead to infer the existence of a physical limit. However, with the advance of computer systems and telecommunications added to the evolution of organizations, this type of limit in practice is diffuse and difficult to identify. Therefore, it is no longer enough that the person belongs to the organization, but that the person has the authorization to interact with the organization's systems. In this sense, Bishop [7] has proposed an alternative definition that allows having a greater detail of the previous concept when considering the following aspects: (1) There is an entity (i.e., a person) that, by its level of trust has the power to violate one or more rules of a given security

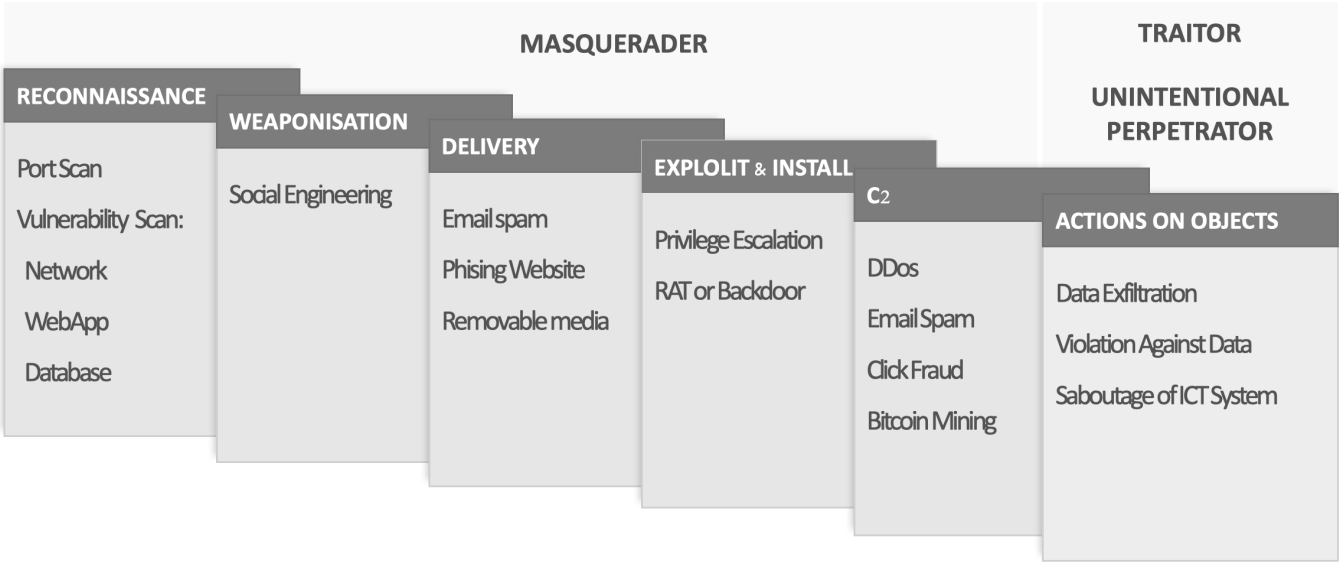

Figure 1. Taxonomy of the Insider Types and Specific Insider Threats [5].

policy. (2) The violation of the security policy is carried out using legitimate accesses. (3) A violation of the access control policy occurs when unauthorized access is obtained.

Figure 1 illustrates the most relevant types of insider and relevant threats [5]. This way, the most commonly seen insider threats are 1) data exfiltration, 2) violations against data integrity or availability, and 3) sabotage of IT systems. Technically, traitors and unintentional perpetrators can fulfill these threats straightway. A masquerader may pose the same threats via an intrusion campaign that consists of social engineering, eavesdropping, packet sniffing, malware delivery, installation, etc.

CSA is used to safeguard sensitive data, sustain fundamental operations, and protect national infrastructure from both malicious insiders and external attackers. It is a multifaceted and well-studied phenomenon, which can be looked upon from several different perspectives [**?**]. The need for situation awareness is essential to understand the organization's environment and accurately to predict and respond to potential problems that might occur. CSA involves three key areas: computing and network components, threat information, and mission dependencies [8]. Achieving this level of situation awareness requires an investment in data collection, data management, and analysis to maintain an ongoing picture of how the computer systems, networks, and users are operating in an organization. In threat awareness, the crucial facts are to identify and track internal incidents and suspicious behavior. Understanding these critical dependencies will anticipating and avoiding situations.

The process of situational awareness can be viewed as a three-phase process [9]: (1) Situation perception. Perception gains awareness about the status, attributes, and dynamics of relevant elements within the enterprise networks. (2) Situation comprehension. Comprehension of the situation encompasses how analysts combine, correlate, and interpret information. (3) Situation projection. Projection of the situation into the near future encompasses the ability to make predictions based on the knowledge acquired through perception and comprehension.

Here, it is essential to note that the comprehension phase is directly influenced by the capability of the IT systems to explain their internal processes. Both comprehension and interpretability are key aspects of Explainable AI. The goal of Explainable Artificial Intelligence (XAI) is "to create a suite of new or modified machine learning techniques that produce explainable models that, when combined with effective explanation techniques, enable end-users to understand, appropriately trust, and effectively manage the emerging generation of Artificial Intelligence (AI) systems" [10]. However, few works relate CSA and XAI. For example, Marino et al. [11] present an adversarial approach to generate explanations for incorrect classifications made by data-driven IDS, and Fujii et al. [12] propose an explainable AI intrusion detection system through the combination of deep tensor and knowledge graph.

3. Method

The global structure of the proposed method for masquerade detection is depicted in Figure 2. It consists of the following components:

Cyber sensing. The perception step is performed by User Activity Monitoring (UAM) sensors installed in every machine of the IT system. In this case, sensors focus on the elements required for the masquerade detection: commands, file ac-

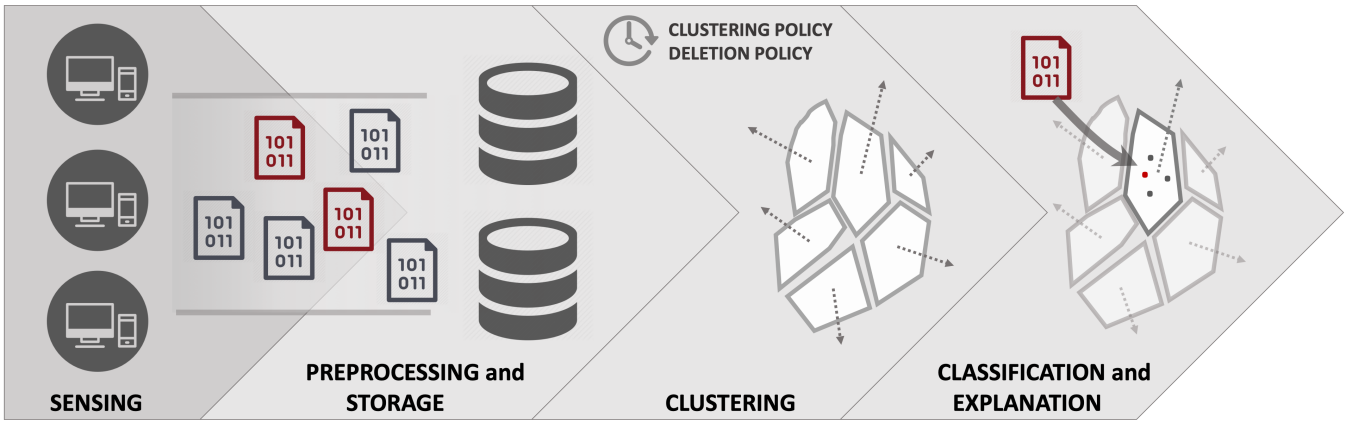

Figure 2. Global structure of the proposed method for masquerade treats detection.

cesses, etc.

Preprocessing and storage. The preprocessing and storage of the collected data must be implemented using the most suitable big-data solution such as Hadoop or Spark. See [13] for a comprehensive comparison.

Clustering policy. The clustering of the collected data must be executed periodically, according to a defined policy, in order to update the clusters used to classify potential threats. These policies can be based on the amount of new data collected or the increase of the error ratio of the classification.

Data deletion policy. As masquerade attacks evolve, it is necessary to define a forget policy in order to delete old data that does not contribute to the correct classification of the masquerade attacks. This policy must define periodical deletions of unuseful data based on noise reduction algorithms such as Repeated Edited Nearest Neighbour (RENN) [14], Blame-Based Noise Reduction (BBNR) [15] or Conservative Redundancy Reduction (CRR) [16].

Classification. This component performs the classification of a potential threat using a two steps process: (1) reaching the nearest clusters using their prototype; and (2) finding the k-nearest neighbors within the cluster.

Explanation. Explainability is one of the significant challenges in machine learning nowadays. In our case, the attack alerts raised by the system must be justified to improve user acceptance and provide appropriated counter-measures. Therefore, our method includes several visual explanation strategies to let the expert introspect the potential attack and understand its nature and severity.

The critical process within this method is the clustering of the dataset. As it is a very time-consuming task, we propose the OK-Means algorithm in order to decrease the classification time. The main schema consists of using the cluster's prototypes to filter and select those clusters containing the most similar records to the potential threats. Then, only the records from the selected clusters are used for the classification. However, this process has an obvious impact on classification performance. Figure 3 shows the expected performance of this approach according to our previous results [17]. Although the processing time follows a linear progression as the number of selected clusters grows, the accuracy may not increase proportionally, slowing down when the number of records is too high. This way, it is necessary to find a trade-off between the processing time and the prediction performance in order to be able to process as many masquerade attacks as possible with an acceptable success rate.

3.1 The OK-Means clustering algorithm

The clustering problem consists of dividing a set of n objects into two or more non-empty subsets or clusters, such that the objects in the same cluster have similar attribute values and have attribute values different from those of the objects in other clusters [18].

The OK-Means clustering algorithm is an improvement of K-Means [19], which by using a new stop or convergence criterion, allows reducing the processing time considerably at the expense of a small reduction of the solution quality. The new stop criterion consists in halting OK-Means when the number of objects that change cluster membership in an iteration is smaller than a threshold U. The value of U expresses an optimal compromise between the computational effort and the solution quality, and it is calculated by applying the Pareto Principle [3, 20].

According to [21, 22] the type of problems that are solved by K-Means belong to the NP-hard problems for $k \geq 2$ or $d \geq 2$. The complexity of K-Means is $O(nkdr)$ [21, 22], while the complexity of OK-Means is $O(nkdr\alpha)$, where r denotes the number of iterations and α is the ratio of the number of iterations of OK-Means

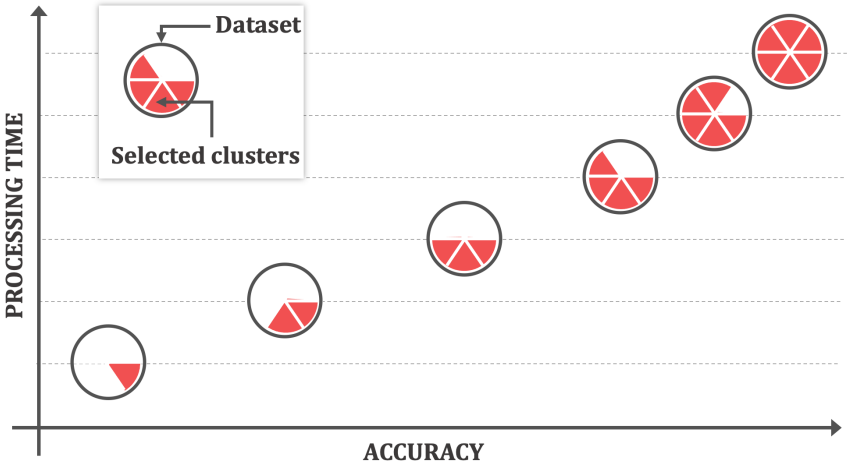

Figure 3. Expected time/performance as the number of selected clusters raises.

and the number of iterations of K-Means. In the experiments carried out, an average value of $\alpha = 0.0389$ was obtained, which shows that OK-Means significantly reduces the computational complexity of K-Means [3, 20].

Let $N = \{x_1, ..., x_n\}$ denote the set of n points to be grouped by a closeness criterion, where $x_i \in \Re^d$ for $i = 1, ..., n$, and $d \geq 1$ is the number of dimensions (the objects' attributes). Further, let $k \geq 2$ be an integer and $K = \{1, ..., k\}$. For a k-partition $P = \{G(1), ..., G(k)\}$ of N, denote μ_j the centroid of group (cluster) $G(j)$, for $j \in K$, and let $M = \{\mu_1, ..., \mu_k\}$.

Thus, the clustering problem can be formulated as a constrained optimization one (see, for instance, [23]):

$$\boldsymbol{P}\text{: minimize } z(W, M) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij} d(x_i, \mu_j) \quad (1)$$

$$\text{subject to } \sum_{j=1}^{k} w_{ij} = 1, \text{ for } i = 1, \ldots, n,$$

$$w_{ij} = 0 \text{ or } 1, \text{ for } i = 1, \ldots, n, \text{ and } j = 1, \ldots, k,$$

where $w_{ij} = 1 \iff$ point x_i belongs to cluster $G(j)$, and $d(x_i, \mu_j)$ denotes the Euclidean distance between x_i and μ_j, for $i = 1, \ldots, n$, and $j = 1, \ldots, k$.

Algorithm 1 shows the pseudocode of the OK-Means algorithm [3, 20].

Algorithm 1 OK-Means

Step 1 Initialization. Produce points $\mu_1, \ldots, \mu_k$, as a random subset of N.

Step 2 Classification. For all $x \in N$ and $j \in K$, compute the Euclidean distance between points x and μ_j, namely, $d(x, \mu_j)$. Then, point (object) $x \in N$ is assigned to a cluster $G(\ddot{\jmath})$ if $d(x, \mu_{\ddot{\jmath}}) \leq d(x, \mu_j)$, for $\ddot{\jmath}, j \in K$.

Step 3 Centroids. Determine the centroid μ_j of cluster $G(j)$, for $j \in K$.

Step 4 Convergence. If $\gamma_r \leq U$ stop the algorithm, otherwise perform another iteration starting from **Step 2**.

In **Step 1 (Initialization)**, the initial centroids are selected for generating the k clusters; in **Step 2 (Classification)**, the membership of an object is determined according to the smallest Euclidean object-centroid distance; in **Step 3 (Centroids)** the centroid of each cluster is calculated; and finally in **Step 4 (Convergence)**, the algorithm is halted when the number of objects that change cluster membership is smaller than the threshold value U.

The OK-Means algorithm was evaluated using a set of experiments with large-size data instances like those used in Big Data. For the experiments, instances of synthetic data were generated, and instances of real data were obtained from the UCI repository [24]. The experimental results show that the OK-Means algorithm reduces the processing time by a factor of 4% with a reduction of less than 2% of the quality of the solution.

4. Experimental evaluation

In order to prove the validity of our proposal, we have conducted an experimental evaluation for the Masquerade Detection scenario. We have chosen the Windows-Users and Intruder simulations Logs Dataset (WUIL), created by [25] that, instead of focusing on users' actions, is based on the objects that are subjects of those actions. Therefore, this dataset uses the concept of *locality*, the tendency of programs to cluster references to memory. Authors of the WUIL dataset define spatial locality as the property of the user to access files that are close to each other, and temporal locality as the

property of the user to access the same file in the near future. As a practical implementation of these ideas, the WUIL dataset abstracts low-level events into a set of 16 temporal, spatial, and directional locality features, showing better performance than a purely action-based approach. This dataset has been collected through a User Activity Monitoring (UAM) sensor installed in several computers running different flavors of Windows OS. This sensing software gathers information for the date, the time, the file access path, and another kind of information.

Afterward, the information collected by the software sensors is preprocessed in order to extract the locality features. These features, as described in [26], are:

Spatial Locality Features are based on the idea that, while working, a user may access files that are close to each other. User events are recorded, including the object and timestamp of the interaction, and segmented into fixed time windows. Then, the file path of the object is used to compute several event distances, summarized in four spatial locality features.

The rationale behind these features is that event distances tend to be short for a legitimate user working on specific tasks, thus visiting objects close one another. An intruder, by contrast, may perform hops between far separated objects, while moving around looking for files of interest.

Temporal Locality Features are based on the intuition that while working, a regular user will frequently access the same files within a short period of time, whereas an intruder will traverse the file system looking for vulnerabilities. Authors define several features taking into account the user's access frequency and the elapsed time between two consecutive accesses to a given file.

Direction Features attempt at capturing where a user is heading at while interacting with the file system. Authors claim that an ordinary user is expected to browse over the file system following a prevailing direction and that the masquerader will have a strange direction pattern. This way, the WUIL dataset includes features describing the user is traveling between file system objects.

The WUIL dataset contains 54,649 instances where 52,884 are legitimate actions, and 1,765 are intruder attacks. The main goal of the evaluation is to measure the performance of our proposed methodology, simulating several data-demanding scenarios. Therefore, we have conducted a cross-validation evaluation using a progressive subsampling of the dataset but keeping this stratification ratio (96.77% legitimate, 3.33% attack). Thus, we have conducted 10-fold evaluations using 20%, 40%, 60%, 80%, and 100% of the instances in the dataset. For every experimentation, we have measured the performance of the system, setting-up several configuration values. Firstly, we have modified the number of clusters calculated by the OK-Means algorithm (denoted as C), and the number of selected clusters (according to the similarity to the prototype) used to filter the most similar records (sC). Finally, we have also tested several values for the k parameter of the nearest neighbor algorithm. The outcome of the classification is calculated through a majority-voting strategy.

As a result, for every subsampling of the dataset we have conducted 15 different evaluations configuring $k = 1, 3, 5$, $C = 4, 8$ and $sC = 1, 2, 4$. The applied evaluation metrics are time improvement (in order to obtain comparable results, all the experimental evaluations were run in the same computer under identical execution conditions) and recall for class "Attack" as explained next.

5. Results

Before analyzing the experimental evaluation, we must notice that the dataset is very unbalanced as 96.77% of the collected instances belong to the class "Legitimate". This is a coherent ratio as "Attacks" are strange events. However, it presents a challenge for the correct detection of attack attempts. If we analyze the accuracy metrics and type of errors for our classification goal, we find that precision is not relevant. Actually, a dummy classifier that always returns class "Legitimate" for any instance would obtain a 96.77% of precision value due to the unbalanced class ratio. Therefore, we must focus on recall as it denotes false negatives. In the case of the "Legitimate" class, recall measures those instances that being classified as "Attack" their actual class is "Legitimate". Here, we can consider it as a false alarm that must be analyzed by the system administrator, but it does not represent any risk for the system. Contrarily, false-negative classification for class "Attack" does represent a serious risk for the organization as these actions would not be identified by the masquerade detection system and reported as legitimate actions. This way, the results presented next will focus on the recall values obtained for class "Attack".

5.1 Baseline performance

Before evaluating the performance of the clustering approach, it is necessary to define the baseline to compare with. In this case, we have tested several classification algorithms over the whole dataset without any clustering optimization. Results are shown in Figure 4. Firstly we have tested two state-of-the-art classification methods such as Random Forest and Multilayer Perceptron, achieving a 0.67 and 0.64 recall value for class "Attack" respectively. These values could be assumed as the baseline performance for our detection system. Then, we have evaluated the k-NN classifier using majority voting and different values for the k parameter, obtaining a lower performance [0.68 - 0.58], but demonstrating that the 1-NN is the best option for this domain. Next, assuming that our goal is to decrease the false-negative rate, we have changed the majority voting strategy to

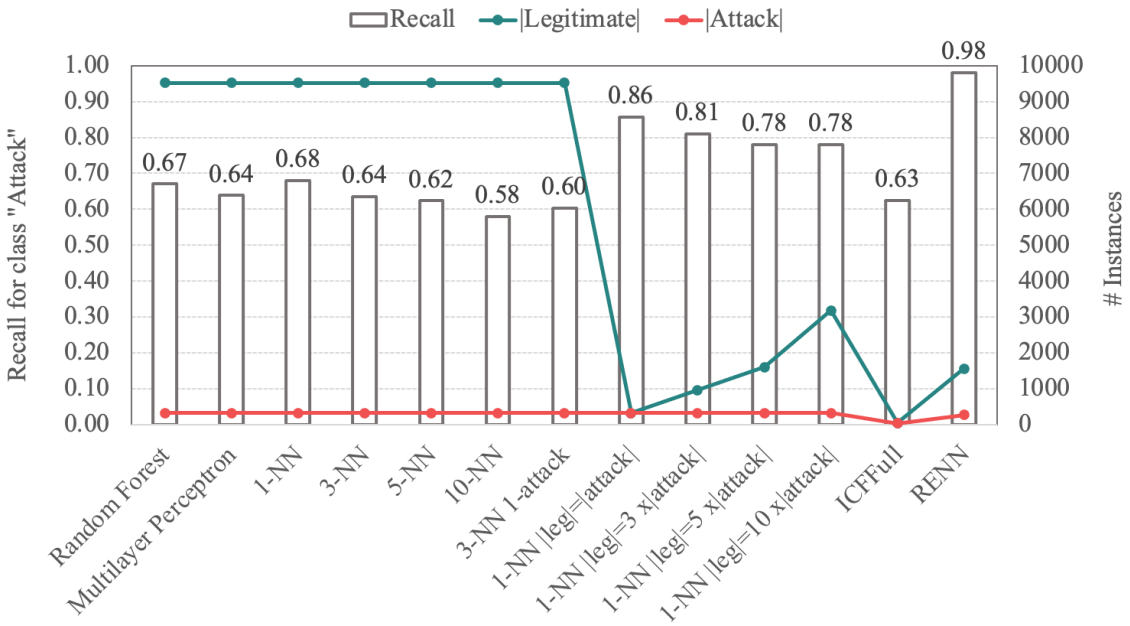

Figure 4. Baseline results using different classification strategies.

classify any instance as an attack if any of the three most similar neighbors is an attack. However, results were also disappointing, with a 0.6 recall value. The following strategies being evaluated consisted of applying an undersampling method for the predominant class "Legitimate". Therefore we randomly removed instances of this class according to the number of "Attack" instances, concretely, 1, 3, 5, and 10 times. Here, we noticed an improvement in the recall values, up to 0.86. From the observation that the removal of redundant instances from the dataset improved the performance of the system, we finally executed noise removal strategies, where RENN obtained a remarkable 0.98 recall value. RENN undersampling of the majority class is done by removing points whose class label differs from a majority of its k nearest neighbors. Removal is applied successively until it can remove no further points.

From this baseline evaluation, we can conclude that a noise removal method, concretely RENN, is a key component of the masquerade detection system in order to raise the performance. By combining this method with the clustering of the dataset, we will be able to obtain not only a very competitive classification but also a good time performance.

Next, we will analyze the impact of the clustering algorithm in the execution times.

5.2 Analysis of the execution time

The five subsamples of the dataset (20%, 40%, 60%, 80%, and 100%) were tested with 15 different configurations for the OK-Means and k-NN algorithms. Table 1 shows the time values achieved when executing 10-fold validation for every dataset subsample. As expected, the execution time decreases as the subset of clusters selected to retrieve instances (sC) decreases. Additionally, if the number of clusters (C) grows, each cluster has fewer instances, and therefore, the execution time also decreases.

From these results, we can observe that we could decrease execution time up to 85% approximately by using 8 clusters and selecting the most similar one (according to the prototype) to the query instance,

Obviously, the use of a subset of clusters in order to decrease execution time will make an impact on the classification performance, as explained next.

5.3 Performance analysis

From our preliminary analysis to obtain a baseline to compare with, we discovered the remarkable impact of the RENN algorithm regarding the recall. The next step consisted of combining this algorithm with the clustering strategy. Here, there are two possibilities. The first option consists of executing the RENN algorithm over the whole dataset and then perform the OK-Means clustering. Its alternative is to cluster first and then apply RENN to every clustering. Results are shown in Figure 5 (left). Surprisingly, the performance obtained when applying REEN to every cluster (labeled as "RENN intracluster" in Figure 5) was similar to the results obtained when this algorithm was not executed, and the complete dataset was clustered. This was the first indicator of the impact of the clustering quality in the performance results. As we will explain in Section 6, OK-Means is able to split legitimate and attack instances very efficiently. Therefore, the impact of the RENN algorithm in every cluster is minimized because noisy instances that may led to the miss-classification of the query are assigned to a different cluster. Although

Table 1. Execution time (in seconds) for different subsamples of the WUIL dataset under varying configurations of OK-Means (denoted as OK-NN when combined with k-NN). Results are shown for $k = 1$, two values of clusters ($C = 4$ and $C = 8$), and different numbers of selected clusters ($sC = 1, 2, 4$). Line 1 reports baseline k-NN execution times without clustering, and lines 2–5 show the percentage of time improvement (ΔT) relative to this baseline.

				Subsamples	20%	40%	60%	80%	100%
Line	Algorithm	k	C	sC	Time (seg.)	Time (seg.)	Time (seg.)	Time (seg.)	Time (seg.)
1	K-NN	1			159.02	350.14	548.94	746.17	941.87
					Δ T (%)	Δ T (%)	Δ T (%)	Δ T (%)	Δ T (%)
2	OK-NN	1	4	1	68.69	66.52	68.97	69.00	67.84
3	OK-NN	1	4	2	32.93	34.92	33.55	26.95	28.63
4	OK-NN	1	8	1	84.46	86.54	85.59	86.24	84.32
5	OK-NN	1	8	4	40.50	41.81	43.35	44.63	36.18

globally, it is a positive feature, if we join the instances of these clusters as the sC parameter grows, we will be recovering the original noisy classifications.

On the other hand, the results obtained when applying RENN to the complete dataset and later performing clustering were very satisfactory, as shown in Figure 5 (RENN complete series).

As there are several combination schemes for noise removal and clustering algorithms, we have also collected their execution times. Figure 5 (right) shows the results. As expected, the complete dataset without noise removal obtains the lowest values. Next, the application of RENN to every cluster obtains average times because the number of instances in each cluster is much lower than the complete dataset. Finally, applying RENN to the whole dataset and later performing clustering + classification obtains the worse execution times.

Once we can conclude the application of RENN to the complete dataset and its later clustering leads to the best recall values, we analyzed the stability of our approach for the different subsamples of the dataset. As Figure 6 reports, results were homogeneous when ranging from the 40% to the 100% percent of the instances. Performance for the 20% of the dataset was a bit lower, indicating that it may be an excessive subsampling. However, as a general result, we can conclude that our method can detect up to 99% of the masquerade attacks with a remarkable time performance.

6. Visual explanatory strategies

CSA consists of making informed decisions based on the comprehension of the environment and the meaning of an event [1]. Therefore, we propose the use of two explanatory strategies to let the security expert analyze both dimensions of the potential masquerade attack. These are the clustering analyzer and the attack introspection visual tools. Both are explained next.

6.1 Clustering analyzer

The clustering analyzer allows the cybersecurity analysts to evaluate the environment of attack through the visualization of the level of hazard of the cluster where the potential threat was classified. This classification is performed by comparing the potential threat to the prototypes of every cluster. Then, this analysis tool shows the proportional size and the Legitimate/Attack ratio of every cluster graphically.

This visual explanation is exemplified in Figure 7. In this case, we have visualized the resulting clusters using the 20% subsampling (left) and the complete dataset (right). This tool uses a hierarchical tree-map where the size (height) of every cluster represents the total number of belonging instances. Then every cluster is divided into the "Legitimate" (blue) and "Attack" (red) areas that are also proportional to the number of instances. Finally, the visualization highlights the "Attack" subcluster, where the potential threat has been classified.

As we can observe with this example, this tool lets the cybersecurity analyst evaluate the potential hazard of the alert. For example, in the first case (Figure 7 left) using the 20% subsampling, there is a potential attack classified in Cluster 4 that shows a minimal number of attacks, indicating a high possibility of false alarm. On the other hand, Figure 7 right shows an attack being classified in a cluster that contains a large number of past attacks, denoting its potential risk.

Finally, both figures demonstrate the excellent performance of the OK-Means algorithm graphically, which globally generates clusters that have either a majority of legitimate or attack instances, especially as the dataset grows. If the performance of the clustering was deficient, clusters should follow the 97% "Legitimate" and 3% "Attack" proportion of the dataset.

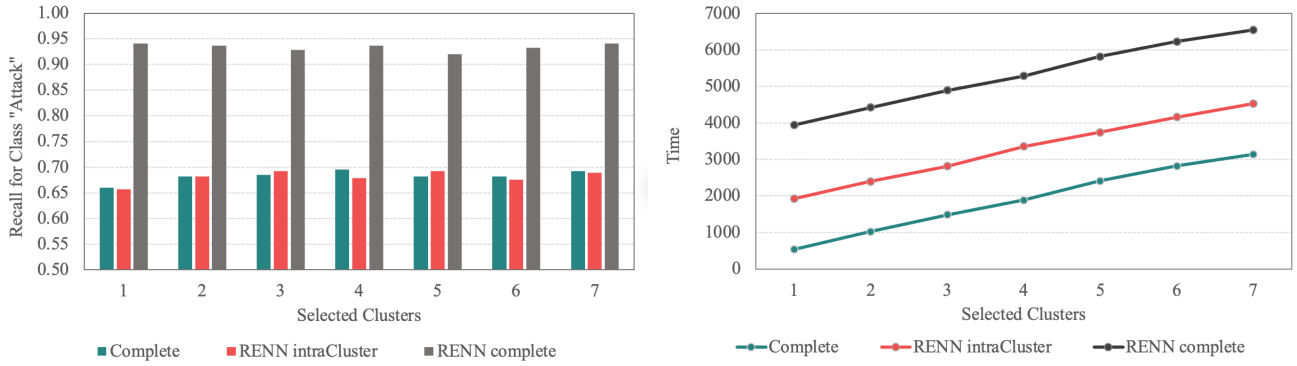

Figure 5. Recall (left) and processing time (right) using different noise removal approaches (dataset 20%).

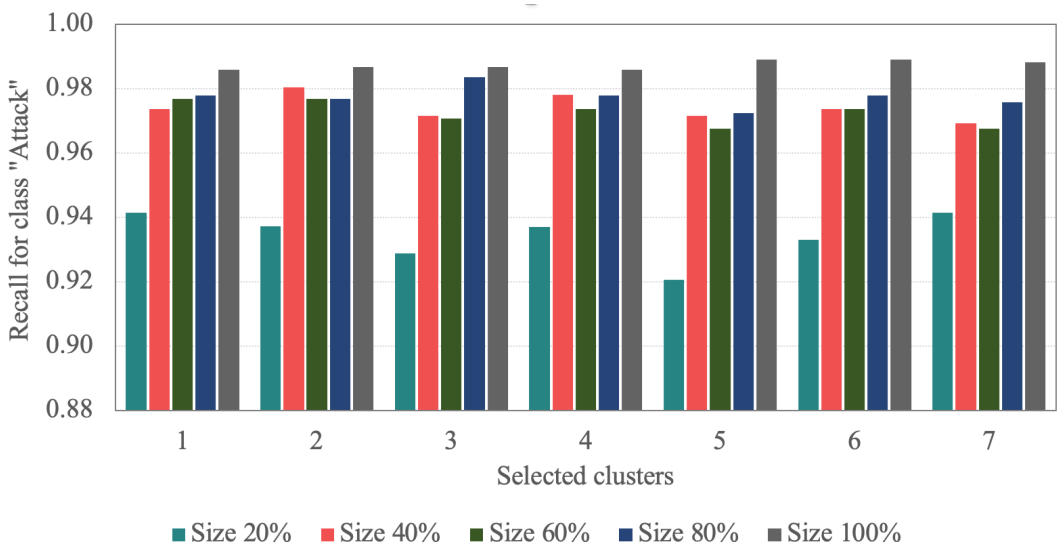

Figure 6. Recall when applying RENN to different subsamplings of the dataset

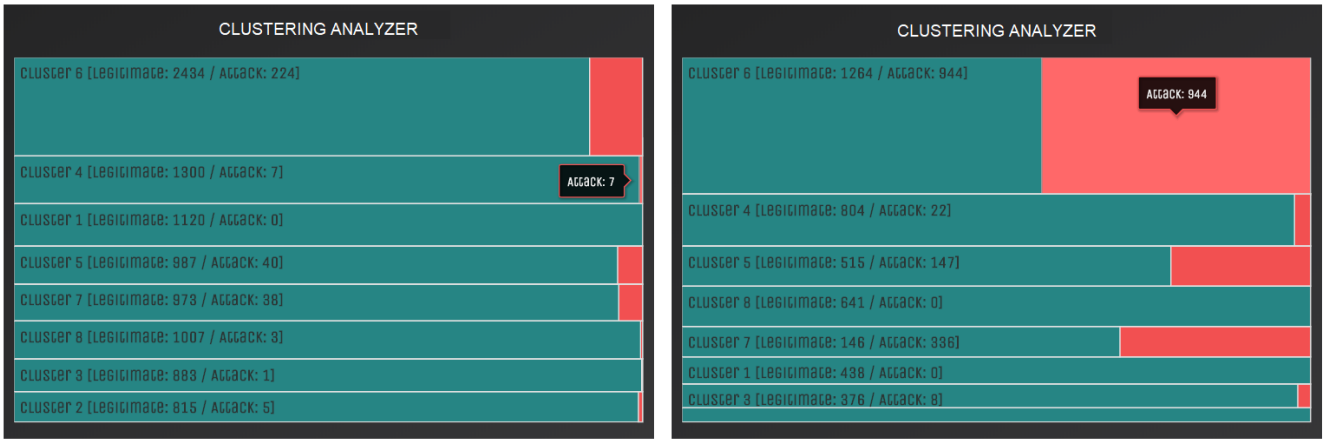

Figure 7. Clustering analyzer tool when visualizing the 20% subsampling (left) and the complete dataset (right).

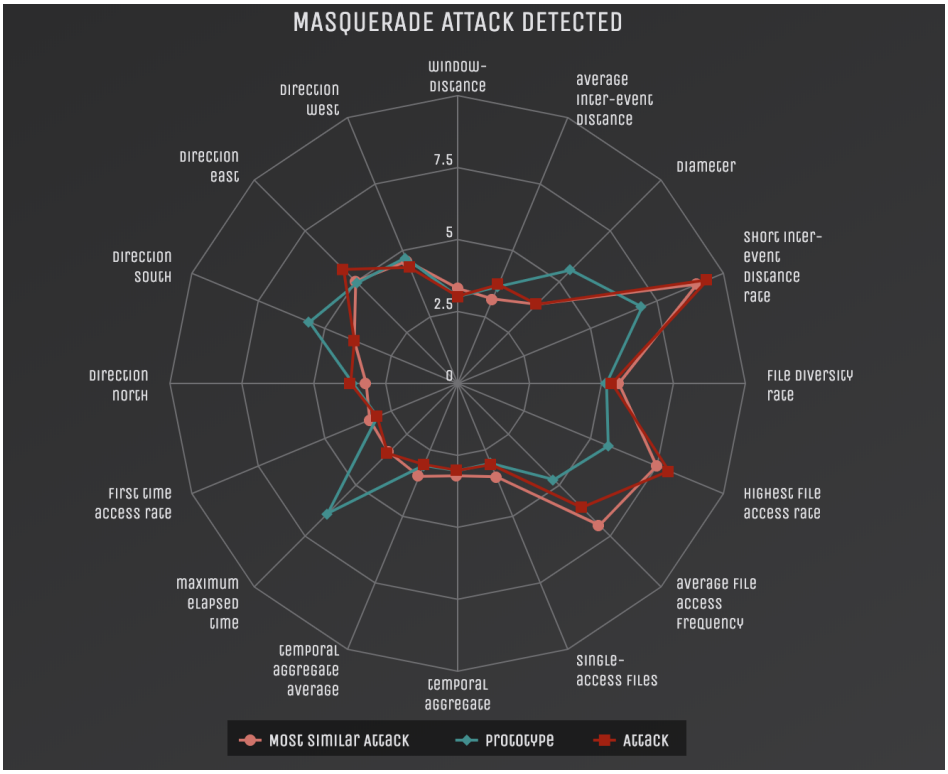

Figure 8. Screenshot of the attack introspection tool showing the features of the attack, the most similar attack that raised the alarm, and the cluster's prototype.

6.2 Attack introspection tool

The attack introspection tool complements the previous tool and allows the security analyst to understand the nature, meaning, and projection of a potential threat. In this case, the features of the attack are displayed graphically to compare them with the most similar attack that raised the alarm (note that we are applying 1-NN) and the cluster's prototype.

A screenshot of this tool is shown in Figure 8. In this case, we use a polar chart where every axis represents one instance's feature. As we are evaluating our approach with the WUIL dataset that uses 16 features to represent masquerade attacks, these are the features shown in the example.

We can observe that the prototype representing all instances in the cluster (blue line) contains average values. However, the potential attack (red line) and the most similar attack that raised the alert (orange) do have anomalous values for some features. In this case, the inter-event rate, the highest file access rate, and the average file access frequency are atypically high compared to the cluster average. These are clear indicators of an intrusion attack as the event, and file access values denote hops between far separated objects looking for files of interest. Additionally, the maximum elapsed time is also lower than average, denoting that the intruder is traversing the file system looking for vulnerabilities.

7. Conclusions

In this work, we have presented an efficient approach for the detection of malicious threats based on the combination of k-NN and a novel clustering strategy, OK-Means [3], that decreases the cost of recomputing clusters as the environment evolves. Concretely, we have demonstrated the benefits of our approach for Masquerade Detection, where execution times decreased up to 68%, and intrusion detection performance is able to detect up to 99% of the masquerade attacks. Moreover, we propose visual explanatory strategies to increase the cybersecurity analyst acceptance of the alerts raised by the IDS.

Data Availability

The dataset used to evaluate the system proposed in this paper is the Windows-Users and Intruder simulations Logs Dataset (WUIL), created by [25]. Availability can be inquired directly with the authors.

CRediT authorship contribution statement

Nelva N. Almanza-Ortega: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Visualization, Writing – original draft.

Joaquin Perez-Ortega: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing – original draft. **Sergio M. Martinez-Monterrubio:** Conceptualization, Writing – review & editing, Supervision. **Juan A. Recio-Garcia:** Conceptualization, Writing – review & editing, Supervision, Project administration.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT in order to improve readability. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by the Spanish Committee of Economy and Competitiveness [TIN2017-87330-R], the Spanish Committee of Science and Innovation [PID2020-114596RB-C21/AEI/10.13039/501100011033] and the BOSCH-UCM Chair on Artificial Intelligence applied to Internet of Things [20896].

References

[1] M. Endsley, "Endsley, m.r.: Toward a theory of situation awareness in dynamic systems. human factors journal 37(1), 32-64," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, pp. 32–64, 03 1995.

[2] M. Conti, A. Dehghantanha, and T. Dargahi, "Cyber threat intelligence : Challenges and opportunities," *CoRR*, vol. abs/1808.01162, 2018. [Online]. Available: http://arxiv.org/abs/1808.01162

[3] J. Pérez-Ortega, N. N. Almanza-Ortega, and D. Romero, "Balancing effort and benefit of k-means clustering algorithms in big data realms," *PloS one*, vol. 13, no. 9, p. e0201874, 2018.

[4] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1 – 38, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370218305988

[5] L. Liu, O. De Vel, Q. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1397–1417, Secondquarter 2018.

[6] D. Cappelli, A. Moore, and R. Trzeciak, *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*. Addison-Wesley Professional, 2012.

[7] M. Bishop, L. Coles-Kemp, D. Gollmann, J. Hunker, and C. W. Probst, Eds., *Insider Threats: Strategies for Prevention, Mitigation, and Response, 22.08. - 26.08.2010*, ser. Dagstuhl Seminar Proceedings, vol. 10341. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2010. [Online]. Available: http://drops.dagstuhl.de/portals/10341/

[8] J. Bone, *Cognitive hack : the new battleground in cybersecurity.*, 2017.

[9] M. Albanese and S. Jajodia, *Formation of Awareness*. Cham: Springer International Publishing, 2014, pp. 47–62. [Online]. Available: https://doi.org/10.1007/978-3-319-11391-3_4

[10] D. Gunning, "Darpa's explainable artificial intelligence (xai) program," 03 2019, pp. ii–ii.

[11] D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An adversarial approach for explainable AI in intrusion detection systems," *CoRR*, vol. abs/1811.11705, 2018. [Online]. Available: http://arxiv.org/abs/1811.11705

[12] M. Fuji, H. Morita, K. Goto, K. Maruhashi, H. Anai, and N. Igata, "Explainable ai through combination of deep tensor and knowledge graph," *Fujitsu Scientific and Technical Journal*, vol. 55, pp. 58–64, 01 2019.

[13] S. Garcia, S. Ramirez-Gallego, J. Luengo, J. Benitez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Analytics*, vol. 1, 12 2016.

[14] D. Hand and B. Batchelor, "Experiments on the edited condensed nearest neighbor rule," *Information Sciences*, vol. 14, no. 3, pp. 171 – 180, 1978. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0020025578900403

[15] B. Smyth and E. McKenna, "Competence models and the maintenance problem," *Computational Intelligence*, vol. 17, no. 2, pp. 235–249, 2001, cited By 51. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0035329732&doi=10.1111%2f0824-7935.00142&partnerID=40&md5=76bf7fe08a06a358f4ddab97a4d78047

[16] S. J. Delany and P. Cunningham, "An analysis of case-base editing in a spam filtering system," in *Advances in Case-Based Reasoning*, P. Funk and P. A. González Calero, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 128–141.

[17] A. Fornells, J. A. Recio-García, B. Díaz-Agudo, E. Golobardes, and E. Fornells, "Integration of a methodology for cluster-based retrieval in jcolibri," in *Case-Based Reasoning Research and Development, 8th International Conference*

on Case-Based Reasoning, ICCBR 2009, Seattle, WA, USA, July 20-23, 2009, Proceedings, ser. Lecture Notes in Computer Science, L. McGinty and D. C. Wilson, Eds., vol. 5650. Springer, 2009, pp. 418–433. [Online]. Available: https://doi.org/10.1007/978-3-642-02998-1_30

[18] R. Xu and D. Wunsch, *Clustering*. John Wiley & Sons, 2008, vol. 10.

[19] R. Jancey, "Multidimensional group analysis," *Australian Journal of Botany*, vol. 14, no. 1, pp. 127–130, 1966.

[20] N. N. Almanza-Ortega, "Desarrollo de heurísticas para la mejora del algoritmo k-means en las fases de clasificaión y convergencia," Ph.D. dissertation, Centro Nacional de Investigación y Desarrollo Tecnológico, 2018.

[21] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "Np-hardness of euclidean sum-of-squares clustering," *Machine learning*, vol. 75, no. 2, pp. 245–248, 2009.

[22] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is np-hard," *Theoretical Computer Science*, vol. 442, pp. 13–21, 2012.

[23] S. Z. Selim and M. A. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Transactions on pattern analysis and machine intelligence*, no. 1, pp. 81–87, 1984.

[24] M. Lichman, "Uci machine learning repository; 2013 [cited 24 aug 2017]. database: Repository of machine learning databases [internet]."

[25] J. B. Camiña, C. Hernandez-Gracidas, R. Monroy, and L. Trejo, "The windows-users and -intruder simulations logs dataset (wuil): An experimental framework for masquerade detection mechanisms," *Expert Systems with Applications*, vol. 41, no. 3, pp. 919 – 930, 2014, methods and Applications of Artificial and Computational Intelligence. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417413006349

[26] J. Camiña, R. Monroy, L. Trejo, and M. Medina-Pérez, "Temporal and spatial locality: An abstraction for masquerade detection," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 2036–2051, 06 2016.