Account Hijacking Detection with KATANA, a K-means Approach for Targeted user behavior ANAlysis

Pierpaolo Artioli, Alessio Magrì, and Pietro Spalluto

Cybersecurity Laboratory, BV TECH S.p.A., 20123 Milan, Italy;
pierpaolo.artioli@bvtech.com (P.A.); alessio.magri@bvtech.com (A.M.);
pietro.spalluto@bvtech.com (P. S.).

Abstract

The escalating threat landscape in cybersecurity is motivating organizations to enhance Security Operations Centers (SOCs) and Security Information and Event Management (SIEM) systems with automated anomaly detection capabilities. Leveraging Machine Learning (ML) to identify behavioral patterns from large amounts of data becomes increasingly important in addressing cybersecurity challenges. Specifically, applying unsupervised machine learning and clustering algorithms to User and Entity Behavior Analysis (UEBA) models can help identify cyber threats by providing crucial insights derived from behavioral patterns. Among an increasing variety of cyber attacks, account hijacking has emerged as a significant insider threat, challenging traditional anomaly detection methodologies due to the unpredictability of human behavior. In response to such challenge, an Online Learning framework can help identify hijacking events, factoring in constant updates to user behavior as it evolves over time. This paper introduces KATANA (a K-means Approach for Targeted user behavior ANAlysis), an online unsupervised ML framework for UEBA. Experimental results on the real-world dataset CLoud-based User Entity behavior analytics Log Data Set (CLUE-LDS) showcase the effectiveness of KATANA, demonstrating promising results across four distinct K-Means based clustering algorithms.

Keywords: UEBA, Unsupervised Machine Learning, Clustering, Online Learning, Anomaly Detection, Account Hijacking

1 Introduction

The ever-growing domain of IT services is being threatened by an increasing number of cyberattacks [1, 2, 3]. This not only presents persistent challenges to the security of critical systems and sensitive data within organizations, but also underscores the need for prompt identification of cybersecurity threats. The recognition of these threats becomes paramount to mitigate potential consequences, such as data breaches, financial loss, and potential harm to an organization's reputation. To strengthen the overall security posture, a synergistic approach that combines comprehensive security policies, employees training, and proactive monitoring is imperative [4]. These foundational elements lay the groundwork for a robust defense against evolving risks. However, to truly fortify an organization's security fabric, a seamless integration with advanced technologies and managed service infrastructures is indispensable. This is where SIEM systems [5] and SOCs [6] play a pivotal role. As organizations strive to stay ahead in the face of sophisticated threats, many of them have directed their efforts toward integrating solutions based on ML [7] into SIEM systems. Notably, this trend extends to UEBA [8] engines, where various forms of ML algorithms are employed to enhance detection capabilities, automating identification and response to potential threats by detecting deviations from legitimate behaviors. One prevalent concern in this context is the rising threat posed by insider attacks, with account hijacking [9, 10] standing out as a prominent form of such threats. Account hijacking

involves the unauthorized access and control of a user's account. Effectively addressing such insider threats is crucial to safeguard sensitive information and prevent unauthorized actions that could compromise an organization's digital security. One common method involves the use of anomaly detection techniques [11, 12] to identify patterns in user and entity behavioral data that do not conform to expected baselines, which are referred to as anomalies, or outliers [13]. Since such behavior is usually not known in advance, unsupervised ML methodologies [14] are well-suited for this purpose. Notably, K-Means based algorithms [15] provide a solid starting point to perform anomaly detection relying on the behavioral description of users. However, in the specific case of account hijacking, these methods may encounter significant challenges. The unpredictability of human behavior makes it difficult to establish a reliable baseline for anomaly detection. As described in [16], human behavior is characterized by sudden changes and a high degree of variability, leading to the risk of a high rate of false positives. Online learning algorithms offer a promising solution [17, 18] to this challenge. Unlike traditional machine learning models that require a static dataset to train upon, online learning algorithms continuously learn and adapt over time and update their models incrementally as each new data point arrives. This continuous learning approach enables these algorithms to quickly adapt to sudden changes in behavior. The state-of-the-art datasets applicable for Insider Threat Detection ML Algorithm training include the artificially generated dataset Community Emergency Response Team (CERT) [19] and the recently introduced CLUE-LDS [16]. While CERT relies on synthetic data, CLUE-LDS sets itself apart by incorporating authentic and consistently anonymized day-by-day user actions collected on a cloud-based document sharing platform. The authenticity of these actions establishes CLUE-LDS as a valuable reference for benchmarking UEBA models.

The primary contributions of this paper are the following:

1. Introduction of KATANA, an account hijacking detection framework based on online K-Means unsupervised machine learning, and the definition of an effective answer to the outlier detection problem in such a scenario;

2. Benchmarking of the proposed framework on the real-world dataset CLUE-LDS [16].

This paper is organized as follows. Section 2 provides a review of the literature on approaches that address anomaly detection in UEBA and the motivations that led to the design of KATANA. Section 3 introduces the KATANA framework and illustrates the clustering algorithms selected for experimental comparison. Section 4 describes the dataset selected for experimentation, configurations and methods used to perform the experiments. Section 5 presents experiment results. Section 6 provides a critical evaluation of the findings and further insights. Section 7 suggests future work in this area.

2 Related work and motivations

This section provides a literature review of different approaches to anomaly detection in UEBA scenarios and introduces the motivations that influenced the design of KATANA.

2.1 Methodologies for anomaly detection using UEBA models

In [15] four different approaches to anomaly detection using UEBA have been proposed to identify insider threats in CERT. The four proposed models are: Gaussian Density Estimation (GDE) [20]; Parzen Window Density Estimation (PWDE) [21]; Principal Component Analysis (PCA) [22]; K-Means [23] Clustering method (KMC). The latter has been examined in [15]

using three different K values (number of clusters). To calculate anomaly score, the KMC approach uses the distance between a new instance and its closest centroid. The numerical score is represented by the relative distance formula D_i/R where D is the euclidean distance between a new instance (i) and its closest centroid and R is the radius of the cluster (the distance between the centroid and the farthest instance from the centroid in the cluster). In [16], authors suggests a mathematical approach to identify user account hijacking scenarios in CLUE-LDS dataset based on daily frequencies of events and heterogeneity of their types. This approach generates a metric used for comparing new events to previous ones and classifies them as anomalous unless they meet the criteria for being similar enough, according to following formula.

$$score_{t_j}(u) = \min_{i=j-q}^{j-1} \left\{ \frac{\sum_{\alpha \in a(u)} \left| c_{\alpha,t_j}(u) - c_{\alpha,t_j}(u) \right|}{\sum_{\alpha \in a(u)} \max\left(c_{\alpha,t_j}(u) - c_{\alpha,t_j}(u) \right)} \right\} \tag{1}$$

where u is a given user, α a specific event type, t a specific day and $c_{\alpha,t_j}(u)$ is the counts of α at day t for u. Formula (1) defines the anomaly detection score for a single event that must be compared to a certain threshold θ. To be considered anomalous, an event needs to satisfy $score_{t_j}(u) > \theta$ where θ is equal to 0.5, as defined in [16].

2.2 Motivations

After evaluating the methodologies employed in existing UEBA anomaly detection approaches detailed in 2.1 within the specific use case of identifying account hijacking attacks in a real-world scenario, several significant insights have been uncovered. Namely, GDE does not reflect real-world scenarios because hijacked user distributions of events are not necessarily described by a multivariate Gaussian distribution. PWDE defines anomalies as events in low-density regions, and this assumption limits the detection of hijacking attack scenarios to only a subset of the possibilities. PCA is too reliant on the calculation of a well-defined threshold, leading to unreliable results. KMC-based algorithms, being thoroughly researched and highly adaptable, are among the most used techniques to find behavioral patterns due to their ability to group samples with similar features (clusters) [24, 25]. Following these considerations, this paper explores the KMC methods introduced in [15] and its relative distance formula in a more extensive way, embedding such methodologies in an unsupervised online learning framework.

3 KATANA framework

KATANA (K-means Approach for Targeted user behavior ANAlysis) is an online unsupervised machine learning framework applicable to UEBA that detects behavior anomalies in a real-world user account hijacking scenario. The main steps of the proposed framework, as shown in Figure 1, are as follows:

1. Baseline definition, in which a behavioral baseline is established through the analysis of preprocessed user historical data, training user-specific K-Means models;

2. Anomaly detection, conducted using a previously created model on preprocessed user real-time data that represents the actions currently being performed by a specific user;

3. (Optional) Supervised human-in-the-loop feedback, finalized to improve the performance of subsequent iterations of the learning process.

The online component [26] involves the continuous update of the user behavior model using real-time data.

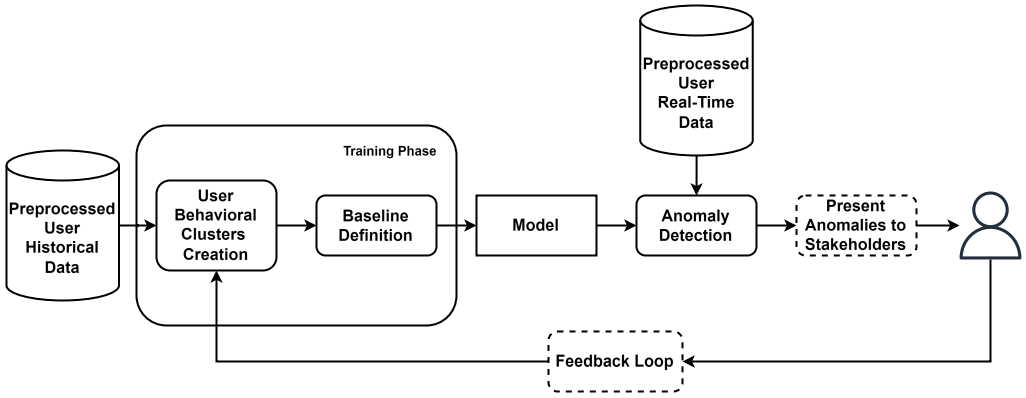


Figure 1: KATANA framework schema

3.1 Clustering Algorithms

In the following section, four K-Means based algorithms compared in KATANA experimentation are presented.

Standard K-Means [23] is a distance-based clustering algorithm that partitions a dataset into a fixed number of clusters denoted by hyperparameter K. The loss function to be minimized is

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_k\|^2 \tag{2}$$

where the function $\|\cdot\|$ is a distance metric, typically the Euclidean distance, x_i is a data point and u_j, $j = 1, ..., k$ is a randomly initialized centroid. The algorithm is iteratively repeated until a stopping criterion is met.

Scalable K-Means++ [27] is based on the K-Means++ [28] algorithm, which is an alternative initialization method in which previously selected centroids influence the next one. In this way, cluster centers are selected one by one, resulting in a decline in scalability. The scalable version partially solves this challenge by selecting more than one point at a time and simultaneously ensuring a good approximation.

Being an NP-Hard problem, K-Means is not suitable for large datasets, but Gradient Descent (GD) methods can reduce its complexity, making it usable on big sets of data. Mini-Batch K-Means [29] is an online clustering method that utilizes Stochastic Gradient Descent (SGD) optimization that takes mini-batches as input. This methodology allows for faster convergence with respect to standard K-Means and reduces the stochastic noise produced by computing the gradient descent one sample at a time.

Nested Mini-Batch K-Means [30] replaces random mini-batches with nested mini-batches which satisfy $\mathcal{M}_t \subseteq \mathcal{M}_{t+1}$, where $\mathcal{M}_t$ is the batch at the step t, so the size of the mini-batches is not decreasing. The selection of nested mini-batches reduces the number of distance calculations by introducing distance bounds and reusing samples selected in previous iterations.

3.2 Clusters definition and Behavior creation through k-means based algorithms

To define a behavioral clusters baseline for an individual user, it's essential to identify the optimal number of clusters needed to train the unsupervised machine learning algorithm. Once the optimal number of clusters (K_o) has been determined, a K-Means-based algorithm can be trained on the historical behavior of that user. The resulting clustering after training the algorithm with K_o clusters is shown in Figure 2(a), using a bidimensional dataset for visualization purposes. Centroids coordinates and cluster radius will be used during the next phases of KATANA to discriminate outliers from inliers when new events need to be classified. Centroids coordinates are automatically computed by the clustering algorithm, while cluster radius is calculated as:

$$R_{c_k} = \max_{x \in c_k} \left(\|x - \mu_k\|^2 \right) \tag{3}$$

where c_k is a given cluster, u_k is the cluster centroid, x belongs to c_k, and $\|\cdot\|^2$ is the Euclidean distance [31]. The concept of the cluster radius can be described as the distance between the cluster element furthest from the centroid. Cluster radius is represented in red in Figure 2(b).

3.3 Nearest cluster identification and anomaly detection

When a new event (x_e) needs to be classified, it is first assigned to one of the existing behavioral clusters, defined by the nearest centroid, which is calculated according to the following formula:

$$\mu_n = \{\mu \in A : \|x_e - \mu\|^2 \leq \|x_e - \mu_k\|^2, \forall \mu_k \in A\} \tag{4}$$

where $A = \{\mu_1, ..., \mu_{K_o}\}$ is the set of all centroids. Figure 2(c) describes how the nearest centroid, and consequently the belonging cluster, is selected for a new event. The new event (x_e) can be classified as anomalous according to the following rule:

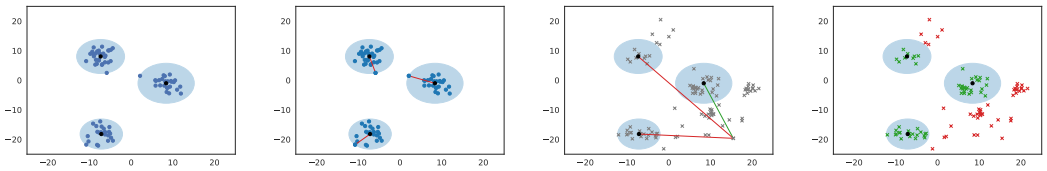
$$anomaly = \begin{cases} \text{true} & \text{if } \|x_e - \mu_n\|^2 > R_{c_k} \\ \text{false} & \text{if } \|x_e - \mu_n\|^2 \leq R_{c_k} \end{cases} \tag{5}$$

The event is classified as an outlier by KATANA if its Euclidean distance from the closest cluster centroid μ_n is greater than the cluster radius R_{c_k}. This formula derives from the ratio used in [15], described in 2.1.

Figure 2(d) shows how new events that are placed outside behavioral clusters are labeled as anomalous when matching the anomaly condition in (5).

3.4 Supervised feedback loop

Following real-time anomaly detection, anomalies are presented to cybersecurity analysts for validation. Confirmed True Positives (TP) could be removed from the online learning process, enhancing the model for future evaluations, but this step is not mandatory. Providing human-in-the-loop insights can help refine the system's performance against evolving cybersecurity threats and organizations or personal work habit changes.



(a) (b) (c) (d)

Figure 2: Example of KATANA framework phases: (a) clustering result using a K-Means based approach, (b) radius computation for each cluster, (c) belonging cluster for a new event, (d) resulting classification after anomaly detection. (Expanded plots available in Figure 3 in Appendix).

4 Experimental Setup

To simulate dynamic event acquisition starting from an existing dataset, a sliding window mechanism [32] has been implemented and detailed in Section 4.2. This allows KATANA to dynamically adapt over time, creating new baselines as new events are processed. Preliminary data cleaning has been conducted on the CLUE-LDS dataset to adjust it for the purposes of behavioral analysis and hijacking detection.

4.1 Dataset and Features Extraction

Since the main challenge in user behavior anomaly detection methods is the unpredictability of genuine user behavior, using a real-world derived dataset such as CLUE-LDS [16] provides a more realistic scenario. This dataset records authentic user events such as logins, file accesses, link shares, and configuration changes in a cloud document sharing platform, collected in JSON format. Some examples of such events are shown in Listing 1, 2. The dataset is anonymized consistently, thus safeguarding original users' privacy without losing information on their behavior. The dataset comprises only legitimate non-malicious users or technical users (entities) actions. To simulate account hijacking within the dataset, a framework that leverages a similarity matrix identifies couples of similar users based on the frequency of unique actions within specific time intervals has been devised in [16]. Such simulation injects anomalies into the dataset, by switching couples of similar users at a certain point in time. For the switching to be performed, two random users, u_1 and u_2, are selected and three conditions need to be simultaneously met, which take into account, for each user, the number of active days before the switch, the number of total events and distinct unique events, and a similarity score between u_1 and u_2.

For practical experimentation, the dataset was subsampled, resulting in six months of user activities. The reduced dataset was then used as the input for the account hijacking injection function introduced in [16] using the parameters: $d_{min} = 25$ days, $c_{min} = 100$ events, $a_{min} = 4$ unique events, $\omega_1 = 0.3$, $\omega_2 = 0.7$, $s_{min} = 0.1$, $s_{max} = 0.6$.

The resulting dataset after hijacking injection undergoes a process of feature extraction. In particular, `params` field, expressed as nested JSON elements, is used to produce multiple features corresponding to its keys. Moreover, the `time` feature is split into `year`, `month`, `day`, `hour`, `minute`, `second` and `dow` (day of week). Missing values are replaced with -1 and a label feature is added representing anomalous (1) or legitimate (0) events, defining a ground truth.

Categorical features encoding [33] and subsequent scaling through z-score normalization [34] is performed as the last step to make data suitable for the following training phase.

To build a personalized behavior for each user, normalized data are split per user. Zero-variance features are removed to avoid a negative influence during training. For experimentation purposes, only hijacked users with at least 6 months of activity have been considered.

Datasets summary is described in Table 1.

Username	Similarity score	# events	# post-switch events	# features	# months
apparent-apricot-lamprey-artexer	0.54	101.105	1.989	36	6
inevitable-olive-meerkat-metallurgist		20.566	3.808	37	6
bitter-red-echidna-retired	0.55	71.504	3.978	39	6
happy-rose-wasp-bingocaller		4.891	29	38	4
civilian-apricot-salamander-sportscoach	0.52	10.155	870	28	6
damp-bronze-leopard-kennelhand		6.793	2.971	40	6
conceptual-red-urial-leafletdistributor	0.26	417.648	959	32	6
married-copper-leech-motorengineer		8.971	6.266	34	5
ethnic-lavender-gerbil-gamingclubmanager	0.29	85.664	22.954	37	6
shared-fuchsia-cardinal-buildingadvisor		89.735	5.656	35	6
good-blush-tyrannosaurus-ambulancecontroller	0.46	16.234	62	21	6
competent-aqua-hare-buildinginspector		4.050	2.799	42	3
horrible-moccasin-mole-licensing	0.49	35.496	578	46	6
profound-amber-peafowl-typewriterengineer		28.542	12.450	37	6
lively-pink-narwhal-yachtmaster	0.41	6.895	225	38	6
mature-beige-takin-prisonchaplain		12.584	11.796	38	6
obedient-maroon-buzzard-caremanager	0.50	15.237	5.520	50	6
stupid-orange-ant-tacker		55.200	1.941	53	6
solid-fuchsia-hyena-metalworker	0.50	25.283	165	33	6
qualified-white-kangaroo-pianoteacher		13.682	2.837	39	6

Table 1: Total number of events, number of events after switching and number of features for each user. Similarity score is described in [16]. Users with less than three months of activity have been discarded during experimentation.

4.2 Online learning setup

To simulate an online learning process over the 6 months of user actions reported in the dataset, a sliding window mechanism is introduced. By setting a window size of 180 days and a step of 30 days, the simulated learning process unfolds over four iterations. For each iteration, the initial 60 days are dedicated to training the model, followed by the next 30 days serving as a test set for evaluation. The window is moved by 30 days for the next iteration. The hyperparameter of the framework consists of K_{max} which defines the maximum number of clusters with the aim of performing Elbow method [35, 36] for each train set. With this setup, outliers detected by (5) needed to be removed to create an updated behavior by adding legitimate events to historical data. In the next iteration of the online learning framework, the updated model and the current model must be compared in order to prevent injection of anomalous data in the historical behavior. This comparison is similar to the approach proposed in [26], in which two parallel models are retained to handle data drift, selecting the model that best fits the data. The comparison is based on the ratio between outliers found and the total number of new samples that must be greater than 0.18 to uphold the current one, according to [26].

4.3 Clusters Definition

In UEBA scenarios, the definition of behavioral patterns to identify legitimate user actions is very important because they represent a baseline to discriminate legitimate and anomalous

behaviors. To avoid the impact of outliers, an initial phase of outlier detection and removal is performed by computing the z-score, and samples with $z > 3$ are considered outliers according to [37], and thus discarded for the creation of the first user behavior.

Once outliers are removed, KATANA applies K-Means based algorithms, as presented in Section 3.1, to group similar events into different clusters. Choosing the number of clusters is often challenging; without a background knowledge of distinctive habits, it is difficult to define an optimal value for the K-Means hyperparameter K.The Elbow method has been selected to find the optimal number of clusters during each training phase. In this experimental setup K_{min} is set to 2, the lowest possible value, and K_{max} is equal to $\sqrt{\frac{n}{2}}$, following the rule of thumb [38], where n is the number of samples in the training set.

4.4 Evaluation metrics

Because account hijacking simulation swaps two similar users' actions, it does not necessarily imply that all subsequent post-switch actions should be considered malicious; rather, they should merely diverge from the original user behavior. The evaluation of the framework relies on assessing entire days' behavior following the switch as either anomalous or legitimate, determined by the ratio between anomalies and total events, similar to the anomaly detection method proposed in [16]. Receiver Operator Characteristic (ROC) [39] and Precision-Recall (PR) curve [40] are particularly suitable for evaluating a score against a ground truth since they apply different thresholds to predictions to evaluate if a sample is anomalous or not. ROC curve is useful to select a good trade-off between True Positive Rate (TPR) and False Positive Rate (FPR), while PR curve expresses the correlation between precision and recall. Area Under PR Curve (AUPRC) is specifically appropriate in presence of imbalanced data [41] as in the case of anomaly detection. Table 2 shows the values for Area Under ROC (AUROC) and AUPRC for each user using the four different algorithms.

Username	K-Means		Mini-Batch K-Means		Scalable K-Means++		Nested Mini-Batch K-Means	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
apparent-apricot-lamprey-artexer	0.9859	0.9334	0.9503	0.7582	0.8967	0.5092	0.9031	0.5036
bitter-red-echidna-retired	0.9986	0.9408	0.9783	0.8769	0.9453	0.8116	0.9786	0.8983
civilian-apricot-salamander-sportscoach	0.8895	0.6019	0.8000	0.4460	0.9001	0.6495	0.8676	0.6515
conceptual-red-urial-leafletdistributor	1.0000	1.0000	0.9900	0.9470	0.9916	0.9804	0.9978	0.9916
damp-bronze-leopard-kennelhand	0.4697	0.1750	0.9711	0.9438	0.4962	0.1821	0.6530	0.3679
ethnic-lavender-gerbil-gamingclubmanager	0.9551	0.8939	0.9694	0.9008	0.9959	0.9933	0.9770	0.9100
good-blush-tyrannosaurus-ambulancecontroller	0.7946	0.6119	0.6860	0.4262	0.7870	0.5918	0.6460	0.4619
horrible-moccasin-mole-licensing	0.9535	0.6420	0.9147	0.5035	0.8854	0.4267	0.8978	0.403
inevitable-olive-meerkat-metallurgist	0.9783	0.8968	0.9620	0.8690	0.9204	0.7029	0.9631	0.8549
lively-pink-narwhal-yachtmaster	0.9886	0.9520	0.9107	0.6460	0.7334	0.2996	0.7186	0.2851
mature-beige-takin-prisonchaplain	0.9279	0.4074	0.9595	0.5500	0.9505	0.5000	0.9505	0.5000
obedient-maroon-buzzard-caremanager	0.9029	0.7486	0.9403	0.8392	0.7156	0.4686	0.7325	0.4963
profound-amber-peafowl-typewriterengineer	0.9746	0.9626	0.9632	0.9318	0.9469	0.8926	0.9425	0.8704
qualified-white-kangaroo-pianoteacher	0.4760	0.2028	0.6370	0.2059	0.7618	0.2954	0.5346	0.2278
shared-fuchsia-cardinal-buildingadvisor	0.9995	0.9980	0.9980	0.9930	1.0000	1.0000	1.0000	1.0000
solid-fuchsia-hyena-metalworker	0.8319	0.2741	0.7677	0.1699	0.8485	0.2142	0.7648	0.1750
stupid-orange-ant-tacker	0.7936	0.5431	0.5903	0.2605	0.7020	0.3383	0.5583	0.3326

Table 2: AUROC and AUPRC for each user and for the four different K-Means-based methods.

Once such curves have been plotted, the best threshold is selected for each user by finding the point on the PR curve that gives the best F1-Score and applied to each day to retrieve Micro and Macro F1-Scores by comparing the predictions and the ground truth labels [42]. Micro-F1 gives an idea of the overall performance of the algorithm, but it does not take into account potentially imbalanced data. Macro-F1, on the other hand, assigns the same weight to each class obtaining a more meaningful result in presence of an under-represented class.

5 Results

Following the assumptions defined in Section 4.2 for the simulation setup, the evaluation of each user's behavior has been conducted over four iterations applied to the reduced CLUE-LDS dataset used for experimentation. Results are reported in Table 3 and best results for each month are highlighted.

Username	Month	K-Means			Mini-Batch K-Means			Scalable K-Means++			Nested Mini-Batch K-Means		
		Clusters	Micro-F1	Macro-F1	Clusters	Micro-F1	Macro-F1	Clusters	Micro-F1	Macro-F1	Clusters	Micro-F1	Macro-F1
apparent-apricot-lamprey-artexer	3	21	**1.0000**	**1.0000**	14	**1.0000**	**1.0000**	15	0.9677	0.4918	14	**1.0000**	**1.0000**
	4	15	**0.9333**	**0.4828**	22	0.7667	0.4340	25	0.5667	0.3617	24	**0.9333**	**0.4828**
	5	14	0.9355	0.8565	29	**0.9677**	**0.9350**	23	**0.9677**	**0.9350**	18	0.8710	0.4655
	6	22	**0.9667**	**0.9663**	19	0.8667	0.8667	14	0.9000	0.8999	19	0.4333	0.3023
bitter-red-echidna-retired	3	26	**1.0000**	**1.0000**	24	**1.0000**	**1.0000**	27	**1.0000**	**1.0000**	34	**1.0000**	**1.0000**
	4	33	**1.0000**	**1.0000**	32	0.9200	0.4792	20	**1.0000**	**1.0000**	27	**1.0000**	**1.000**
	5	27	**0.9355**	**0.9217**	19	0.8710	0.8524	15	0.7097	0.4151	47	0.8065	0.7786
	6	17	**0.9667**	**0.9590**	8	**0.9667**	**0.9590**	12	0.2667	0.2105	13	**0.9667**	**0.9590**
civilian-apricot-salamander-sportscoach	3	8	0.8710	0.4655	9	**1.0000**	**1.0000**	8	0.8710	0.4655	6	0.7419	0.4219
	4	10	0.8966	0.4727	9	**1.0000**	**1.0000**	8	0.8966	0.4727	8	0.8276	0.4528
	5	10	**0.7667**	**0.7532**	8	0.7333	0.4231	13	0.7333	0.7222	12	0.7333	0.7222
	6	8	**0.9000**	**0.8990**	10	0.5000	0.3333	10	**0.9000**	**0.8990**	12	0.8667	0.8643
conceptual-red-urial-leafletdistributor	3	41	**1.0000**	**1.0000**	36	**1.0000**	**1.0000**	35	**1.0000**	**1.0000**	43	0.9677	0.4918
	4	33	**1.0000**	**1.0000**	35	**1.0000**	**1.0000**	34	0.9667	0.4915	39	0.9667	0.4915
	5	30	**1.0000**	**1.0000**	29	0.6667	0.4000	25	**1.0000**	**1.0000**	29	**1.0000**	**1.0000**
	6	18	**1.0000**	**1.0000**	24	0.2500	0.2000	25	0.9000	0.8810	19	0.9500	0.9373
damp-bronze-leopard-kennelhand	3	8	0.8276	0.4528	6	0.9655	0.4912	5	**1.0000**	**1.0000**	6	0.9310	0.4821
	4	3	0.9615	0.4902	4	**1.0000**	**1.0000**	4	0.9615	0.4902	5	0.7692	0.4348
	5	6	0.5000	0.3333	21	**1.0000**	**1.0000**	4	0.5000	0.3333	4	0.6250	0.5636
	6	3	0.2609	0.2069	26	**0.8261**	**0.8083**	3	0.2609	0.2609	5	0.5652	0.5644
ethnic-lavender-gerbil-gamingclubmanager	3	19	0.9677	0.4918	19	0.9677	0.4918	21	**1.0000**	**1.0000**	19	**1.0000**	**1.0000**
	4	26	0.9655	0.4912	20	0.9655	0.4912	15	**1.0000**	**1.0000**	18	**1.0000**	**1.0000**
	5	24	0.7667	0.7643	17	**0.9333**	0.9330	28	**0.9333**	**0.9333**	19	0.5000	0.3333
	6	16	**1.0000**	**1.0000**	16	0.9667	0.4915	23	0.9000	0.4737	13	0.0000	0.0000
good-blush-tyrannosaurus-ambulancecontroller	3	11	**1.0000**	**1.0000**	12	**1.0000**	**1.0000**	14	**1.0000**	**1.0000**	8	**1.0000**	**1.0000**
	4	23	**1.0000**	**1.0000**	24	0.9655	0.4912	24	**1.0000**	**1.0000**	10	**1.0000**	**1.0000**
	5	6	0.9333	0.8148	7	0.9333	0.7321	8	**0.9667**	**0.8909**	9	0.9333	0.7321
	6	3	**0.8929**	**0.6706**	3	**0.8929**	**0.6706**	3	0.8571	0.4615	3	**0.8929**	**0.6706**
horrible-moccasin-mole-licensing	3	12	**1.0000**	**1.0000**	8	**1.0000**	**1.0000**	5	**1.0000**	**1.0000**	12	**1.0000**	**1.0000**
	4	10	**1.0000**	**1.0000**	9	**1.0000**	**1.0000**	12	**1.0000**	**1.0000**	18	**1.0000**	**1.0000**
	5	10	0.8400	0.4565	18	0.8400	0.4565	13	0.8400	0.4565	7	**0.8800**	**0.4681**
	6	23	**0.6087**	**0.3784**	8	**0.6087**	**0.3784**	8	**0.6087**	**0.3784**	17	**0.6087**	**0.3784**
inevitable-olive-meerkat-metallurgist	3	6	**1.0000**	**1.0000**	16	0.9677	0.4918	14	0.9032	0.4746	5	**1.0000**	**1.0000**
	4	14	**0.9333**	**0.4828**	3	**0.9333**	**0.4828**	17	0.7667	0.4340	21	0.9000	0.4737
	5	11	0.8710	0.7933	13	**0.9032**	**0.8050**	21	0.8387	0.7567	15	0.8387	0.5974
	6	14	**1.0000**	**1.0000**	19	0.9286	0.8783	16	0.9286	0.8783	3	0.9643	0.9338
lively-pink-narwhal-yachtmaster	3	4	**1.0000**	**1.0000**	7	0.7419	0.4259	10	0.8065	0.4464	17	0.7419	0.4259
	4	16	**0.8966**	**0.4727**	11	0.7931	0.4423	4	0.7931	0.4423	11	0.7931	0.4423
	5	12	**1.0000**	**1.0000**	16	0.9032	0.8963	5	0.6452	0.3922	6	0.6774	0.4038
	6	7	0.8261	0.8231	7	**1.0000**	**1.0000**	6	0.4783	0.3235	9	0.4783	0.3235
mature-beige-takin-prisonchaplain	3	6	**1.0000**	**1.0000**	6	**1.0000**	**1.0000**	5	**1.0000**	**1.0000**	6	**1.0000**	**1.0000**
	4	5	**1.0000**	**1.0000**	4	**1.0000**	**1.0000**	5	**1.0000**	**1.0000**	6	**1.0000**	**1.0000**
	5	5	**0.8966**	**0.4727**	6	**0.8966**	**0.4727**	4	**0.8966**	**0.4727**	4	**0.8966**	**0.4727**
	6	3	**0.7333**	**0.4231**	6	**0.7333**	**0.4231**	3	**0.7333**	**0.4231**	3	**0.7333**	**0.4231**
obedient-maroon-buzzard-caremanager	3	10	**1.0000**	**1.0000**	5	**1.0000**	**1.0000**	34	0.3226	0.2439	45	0.3226	0.2439
	4	5	0.9310	0.4821	5	**0.9655**	**0.4912**	4	0.5517	0.3556	5	0.5517	0.3556
	5	13	**0.7667**	0.7643	13	**0.7667**	**0.7664**	9	0.6667	0.6667	12	**0.7667**	**0.7664**
	6	15	0.7333	0.7129	11	0.9000	0.8769	13	**0.9333**	**0.9148**	9	**0.9333**	**0.9148**
profound-amber-peafowl-typewriterengineer	3	16	**0.9677**	**0.4918**	7	**0.9677**	**0.4918**	7	**0.9677**	**0.4918**	7	**0.9677**	**0.4918**
	4	8	**1.0000**	**1.0000**	7	0.9655	0.4912	17	0.6897	0.4082	9	0.8621	0.4630
	5	9	**1.0000**	**1.0000**	18	0.9333	0.9206	7	0.9667	0.9590	4	0.9667	0.9590
	6	9	0.9000	0.8667	7	**0.9333**	**0.9068**	12	**0.9333**	**0.9068**	6	0.9000	0.8667
qualified-white-kangaroo-pianoteacher	3	11	0.6774	0.4038	5	**1.0000**	**1.0000**	42	0.9677	0.4918	11	0.7742	0.4364
	4	8	0.7931	0.4423	14	0.8966	0.4727	23	**0.9655**	**0.4912**	23	0.4138	0.2927
	5	4	0.7333	0.5833	13	0.7333	0.4231	11	0.8000	0.4444	4	**0.8333**	**0.7948**
	6	7	**0.5667**	**0.4222**	6	0.5333	0.3478	10	0.5333	0.3478	6	**0.5667**	**0.4222**
shared-fuchsia-cardinal-buildingadvisor	3	44	**1.0000**	**1.0000**	39	**1.0000**	**1.0000**	43	**1.0000**	**1.0000**	33	**1.0000**	**1.0000**
	4	37	0.9667	0.4915	28	**1.0000**	**1.0000**	33	**1.0000**	**1.0000**	35	**1.0000**	**1.0000**
	5	11	**1.0000**	**1.0000**	26	**1.0000**	**1.0000**	32	**1.0000**	**1.0000**	10	**1.0000**	**1.0000**
	6	16	**0.9565**	**0.9565**	8	0.9130	0.9129	29	**0.9565**	**0.9565**	15	**0.9565**	**0.9565**
solid-fuchsia-hyena-metalworker	3	16	0.8065	0.4464	17	**0.8710**	**0.4655**	18	0.8387	0.4561	19	0.8387	0.4561
	4	16	0.9000	0.4737	16	**0.9667**	**0.4915**	14	**0.9667**	**0.4915**	18	0.9333	0.4828
	5	14	**0.9032**	**0.8050**	13	0.8387	0.4561	16	0.8710	0.4655	16	0.8387	0.4561
	6	14	**1.0000**	**1.0000**	16	0.5000	0.3333	16	0.5000	0.3333	13	0.5000	0.3333
stupid-orange-ant-tacker	3	7	**1.0000**	**1.0000**	6	**1.0000**	**1.0000**	9	**1.0000**	**1.0000**	13	**1.0000**	**1.0000**
	4	32	0.9310	0.4821	28	**1.0000**	**1.0000**	21	**1.0000**	**1.0000**	52	**1.0000**	**1.0000**
	5	44	0.5667	0.5417	18	**0.8333**	**0.4545**	9	**0.8333**	**0.4545**	34	**0.8333**	**0.4545**
	6	14	**0.7931**	**0.7803**	16	0.5172	0.3409	14	0.5172	0.3409	12	0.5172	0.3409

Table 3: Micro and Macro F1 scores for each user during the online learning process after the threshold has been applied to the scores.

6 Discussion

Taking into account Table 3, K-Means emerges as the top performer algorithm, consistently showing a better score compared to other algorithms through different iterations. From the ta-

ble, it is evident that the number of clusters in this online learning approach varies significantly. This demonstrates that user behaviors represent an actual scenario in which user actions might exhibit substantial temporal differentiation. The Elbow method can be useful to define an ideal number of clusters, but in real-world scenarios, where user behavior actions are significantly variable, the results it yields could be ambiguous.

In light of the previous statement, algorithms performed on the same dataset portions can be initialized using a different number of clusters. In all four K-Means based algorithms, due to random beginning centroid position, during elbow method calculation, different K values can be assigned to the same starting dataset. This produces underperforming results for iterations in which the number of clusters is suboptimal to describe that particular user behavior.

Before the user switching process in CLUE-LDS preprocessing phase, it cannot be ensured that historical user behavior does not contain anomalies. The definition of anomaly does not concern only actions related to hijacking attacks. This means that anomalies can often be found in user actions and negatively influence behavior definition. Using the z-score method to remove anomalies, as described in Section 4.3 may not always be enough to create an anomaly-free behavior baseline. After the switching phase, the above-mentioned problem remains the same because the subset of events moved to a different user can also contain anomalies.

Lastly, even if ROC and PR curves are valid metrics to evaluate a binary classifier performance, they are not suitable to select a proper threshold to discriminate between anomalies and legitimate actions.

7 Conclusions

This paper presented KATANA, an online learning framework applied on CLUE-LDS for user account hijacking detection, comparing four different K-Means based approaches. To handle the variability of users, an online learning setup was implemented using a sliding window mechanism allowing KATANA to be retrained to update the learned behavior models. Event-level predictions were aggregated into day-level predictions, and ROC and PR curves were used to validate the results and to obtain a threshold used to classify daily user behavior as anomalous or legitimate. KATANA experimentation diverges from the anomaly detection method presented in [16] since it establishes a behavioral baseline for each user and explores the entire proposed feature space, rather than considering all users as a whole, while also factoring in only a single feature, namely, the frequency of unique user actions. Finally, Micro-F1 and Macro-F1 were computed, showing that standard K-Means is the best algorithm for such a task. This work presents a critical point that needs further investigation: Elbow method does not guarantee the same result after each iteration, causing the algorithm to produce different results over more iterations on the same set of data. Further work ahead includes enhancing the framework's capabilities by conducting experiments across various cybersecurity applications, broadening the framework's utility and applicability.

8 Acknowledgements

This work was supported in part by the Fondo Europeo di Sviluppo Regionale Puglia 921 Programma Operativo Regionale (POR) Puglia 2014-2020-Axis I-Specific Objective 1a-Action 1.1 922 (Research and Development)-Project Title: CyberSecurity and Security Operation Center (SOC) 923 Product Suite by BV TECH S.p.A., under Grant CUP/CIG B93G18000040007.

References

[1] Wenli Duo, MengChu Zhou, and Abdullah Abusorrah. A survey of cyber attacks on cyber physical systems: Recent advances and challenges. *IEEE/CAA Journal of Automatica Sinica*, 9(5):784–800, 2022.

[2] Xiang Liu, Sayed Fayaz Ahmad, Muhammad Khalid Anser, Jingying Ke, Muhammad Irshad, Jabbar Ul-Haq, and Shujaat Abbas. Cyber security threats: A never-ending challenge for e-commerce. *Frontiers in Psychology*, 13, 2022.

[3] Harun Oz, Ahmet Aris, Albert Levi, and A. Selcuk Uluagac. A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Comput. Surv.*, 54(11s), sep 2022.

[4] Georgi Tsochev, Roumen Trifonov, Ognian Nakov, Slavcho Manolov, and Galya Pavlova. Cyber security: Threats and challenges. In *2020 International Conference Automatics and Informatics (ICAI)*, pages 1–6, 2020.

[5] Maximilian Rosenberg, Bettina Schneider, Christopher Scherb, and Petra Maria Asprion. An adaptable approach for successful siem adoption in companies, 2023.

[6] Arif Ali Mughal. Building and securing the modern security operations center (soc). *International Journal of Business Intelligence and Big Data Analytics*, 5(1):1–15, Jan. 2022.

[7] Zhimin Zhang, Huansheng Ning, Feifei Shi, Fadi Farha, Yang Xu, Jiabo Xu, Fan Zhang, and Kim-Kwang Raymond Choo. Artificial intelligence in cyber security: research advances, challenges, and opportunities. *Artificial Intelligence Review*, pages 1–25, 2022.

[8] Joyatee Datta, Rohini Dasgupta, Sayantan Dasgupta, and Karmuru Rohit Reddy. Real-time threat detection in ueba using unsupervised learning algorithms. In *2021 5th International Conference on Electronics, Materials Engineering Nano-Technology (IEMENTech)*, pages 1–6, 2021.

[9] Makhdoom Muhammad Naeem, Intesab Hussain, and Malik Muhammad Saad Missen. A survey on registration hijacking attack consequences and protection for session initiation protocol (sip). *Computer Networks*, 175:107250, 2020.

[10] Israel O. Ogundele, Abigail O. Akinade, and Harrison O. Alakiri. Detection and prevention of session hijacking in web application management. *International Journal of Advanced Research in Computer and Communication Engineering*, 9:1–10, 2020.

[11] Ali Bou Nassif, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. Machine learning for anomaly detection: A systematic review. *IEEE Access*, 9:78658–78700, 2021.

[12] Manya Ali Salitin and Ali Hussein Zolait. The role of user entity behavior analytics to detect network attacks in real time. In *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pages 1–5, 2018.

[13] Abir Smiti. A critical overview of outlier detection methods. *Computer Science Review*, 38:100306, 2020.

[14] Mohamed Alloghani, Dhiya Al-Jumeily, Jamila Mustafina, Abir Hussain, and Ahmed J. Aljaaf. *A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science*, pages 3–21. Springer International Publishing, Cham, 2020.

[15] Junhong Kim, Minsik Park, Haedong Kim, Suhyoun Cho, and Pilsung Kang. Insider threat detection based on user behavior modeling and anomaly detection algorithms. *Applied Sciences*, 9(19), 2019.

[16] Max Landauer, Florian Skopik, Georg Höld, and Markus Wurzenberger. A user and entity behavior analytics log data set for anomaly detection in cloud computing. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 4285–4294, 2022.

[17] Steven C.H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.

[18] Shangbin Han, Qianhong Wu, Han Zhang, Bo Qin, Jiankun Hu, Xingang Shi, Linfeng Liu, and Xia Yin. Log-based anomaly detection with robust feature extraction and online learning. *IEEE Transactions on Information Forensics and Security*, 16:2300–2311, 2021.

[19] Joshua Glasser and Brian Lindauer. Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE Security and Privacy Workshops*, pages 98–104, 2013.

[20] Mu Zhou, Xinyue Li, Ya Wang, Shanshan Li, Yingyi Ding, and Wei Nie. 6g multisource-information-fusion-based indoor positioning via gaussian kernel density estimation. *IEEE Internet of Things Journal*, 8(20):15117–15125, 2021.

[21] Van Nguyen. The anomaly detection efficiency of kernel density estimation functions on uav images. *Journal of Science and Technique*, 9, 05 2022.

[22] Arun Abhishek Imayakumar, Anamika Dubey, and Anjan Bose. Anomaly detection for primary distribution system measurements using principal component analysis. In *2020 IEEE Texas Power and Energy Conference (TPEC)*, pages 1–6, 2020.

[23] Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210, 2023.

[24] Sakib Shahriar, A. R. Al-Ali, Ahmed H. Osman, Salam Dhou, and Mais Nijim. Machine learning approaches for ev charging behavior: A review. *IEEE Access*, 8:168980–168993, 2020.

[25] Yu-Sheng Su and Sheng-Yi Wu. Applying data mining techniques to explore user behaviors and watching video patterns in converged it environments. *Journal of Ambient Intelligence and Humanized Computing*, Jan 2021.

[26] Mitchell D. Woodbright, Md Anisur Rahman, and Md Zahidul Islam. A novel incremental clustering technique with concept drift detection, 2020.

[27] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++, 2012.

[28] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[29] D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 1177–1178, New York, NY, USA, 2010. Association for Computing Machinery.

[30] James Newling and François Fleuret. Nested mini-batch k-means. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[31] Taher M Ghazal. Performances of k-means clustering algorithm with different distance metrics. *Intelligent Automation & Soft Computing*, 30(2):735–742, 2021.

[32] Lattawit Kulanuwat, Chantana Chantrapornchai, Montri Maleewong, Papis Wongchaisuwat, Supaluk Wimala, Kanoksri Sarinnapakorn, and Surajate Boonya-aroonnet. Anomaly detection using a sliding window technique and data imputation with machine learning for hydrological time series. *Water*, 13(13), 2021.

[33] John T. Hancock and Taghi M. Khoshgoftaar. Survey on categorical data for neural networks. *Journal of Big Data*, 7(1):28, Apr 2020.

[34] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.

[35] Fan Liu and Yong Deng. Determine the number of unknown targets in open world based on elbow method. *IEEE Transactions on Fuzzy Systems*, 29(5):986–995, 2021.

[36] Md Salik Parwez, Danda B. Rawat, and Moses Garuba. Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network. *IEEE Transactions on Industrial Informatics*, 13(4):2058–2065, 2017.

[37] Abdulmalik Shehu Yaro, Filip Maly, and Pavel Prazak. Outlier detection in time-series receive signal strength observation using z-score method with sn scale estimator for indoor localization. *Applied Sciences*, 13(6), 2023.

[38] Mohiuddin Ahmed and Abdun Naser Mahmood. Novel approach for network traffic pattern anal-

ysis using clustering-based collective anomaly detection. *Annals of Data Science*, 2(1):111–130, Mar 2015.

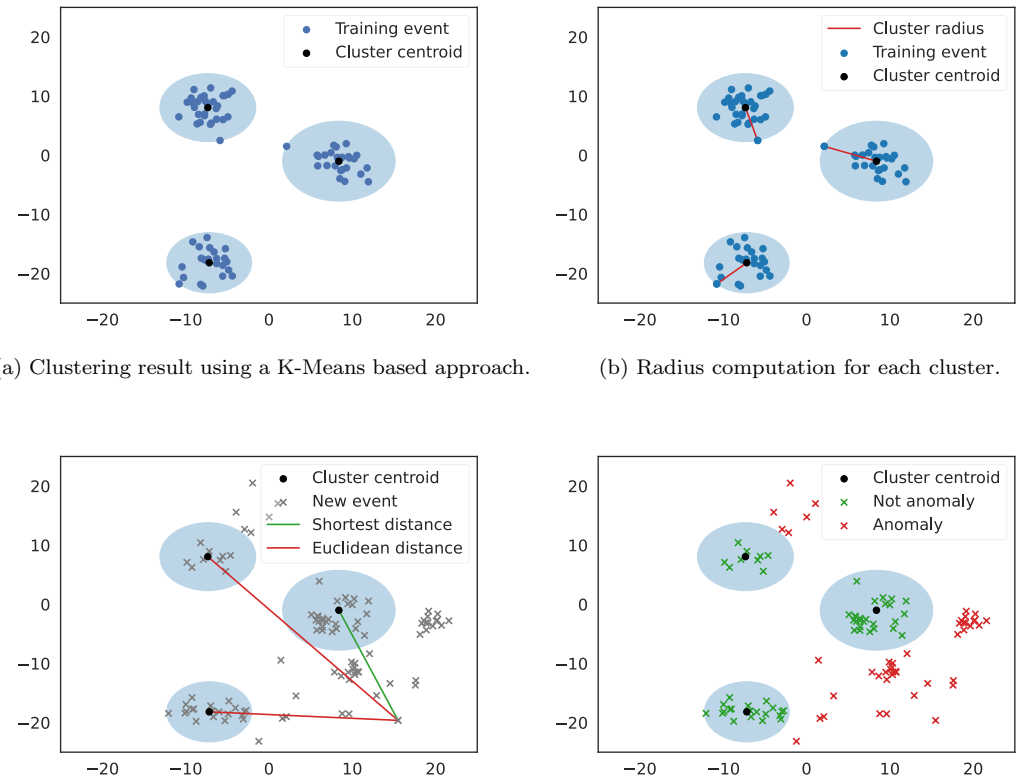
[39] Nebrase Elmrabit, Feixiang Zhou, Fengyin Li, and Huiyu Zhou. Evaluation of machine learning algorithms for anomaly detection. In *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–8, 2020.

[40] Sheraz Naseer, Yasir Saleem, Shehzad Khalid, Muhammad Khawar Bashir, Jihun Han, Muhammad Munwar Iqbal, and Kijun Han. Enhanced network anomaly detection based on deep neural networks. *IEEE Access*, 6:48231–48246, 2018.

[41] Shahzad Ali Khan and Zeeshan Ali Rana. Evaluating performance of software defect prediction models using area under precision-recall curve (auc-pr). In *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)*, pages 1–6, 2019.

[42] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020.

A Appendix



(a) Clustering result using a K-Means based approach.

(b) Radius computation for each cluster.

(c) Belonging cluster for a new event.

(d) Resulting classification after anomaly detection

Figure 3: Example of KATANA framework phases

```
{
   "params":{
      "user":"intact-gray-marlin-trademarkagent"
   },
   "type":"login_successful",
   "time":"2019-11-14T11:26:43Z",
   "uid":"intact-gray-marlin-trademarkagent",
   "id":21567530,
   "uidType":"name"
}
```


Listing 1: CLUE-LDS log representing a successful user login.


```
{
   "isLocalIP":false,
   "params":{
      "path":"/proud-copper-orangutan-artexer/doubtful-plum-
         ptarmigan-merchant/insufficient-amaranth-earthworm-
         qualitycontroller/curious-silver-galliform-
         tradingstandards/incredible-indigo-octopus-printfinisher
         /wicked-bronze-sloth-claimsmanager/frantic-aquamarine-
         horse-cleric"
   },
   "type":"file_accessed",
   "time":"2019-11-14T11:26:51Z",
   "uid":"graceful-olive-spoonbill-careersofficer",
   "id":21567531,
   "location":{
      "countryCode":"AT",
      "countryName":"Austria",
      "region":"4",
      "city":"Gmunden",
      "latitude":47.915,
      "longitude":13.7959,
      "timezone":"Europe/Vienna",
      "postalCode":"4810",
      "metroCode":null,
      "regionName":"Upper Austria",
      "isInEuropeanUnion":true,
      "continent":"Europe",
      "accuracyRadius":50
   },
   "uidType":"ipaddress"
}
```


Listing 2: CLUE-LDS log representing a file being accessed.