Education and Information Technologies
https://doi.org/10.1007/s10639-024-12480-x

Using data clustering to reveal trainees' behavior in cybersecurity education

Karolína Dočkalová Burská[1] · Jakub Rudolf Mlynárik[1] · Radek Ošlejšek[1]

Received: 21 April 2023 / Accepted: 9 January 2024

Abstract

In cyber security education, hands-on training is a common type of exercise to help raise awareness and competence, and improve students' cybersecurity skills. To be able to measure the impact of the design of the particular courses, the designers need methods that can reveal hidden patterns in trainee behavior. However, the support of the designers in performing such analytic and evaluation tasks is ad-hoc and insufficient. With unsupervised machine learning methods, we designed a tool for clustering the trainee actions that can exhibit their strategies or help pinpoint flaws in the training design. By using a *k-means++* algorithm, we explore clusters of trainees that unveil their specific behavior within the training sessions. The final visualization tool consists of views with scatter plots and radar charts. The former provides a two-dimensional correlation of selected trainee actions and displays their clusters. In contrast, the radar chart displays distinct clusters of trainees based on their more specific strategies or approaches when solving tasks. Through iterative training redesign, the tool can help designers identify improper training parameters and improve the quality of the courses accordingly. To evaluate the tool, we performed a qualitative evaluation of its outcomes with cybersecurity experts. The results confirm the usability of the selected methods in discovering significant trainee behavior. Our insights and recommendations can be beneficial for the design of tools for educators, even beyond cyber security.

Keywords Visual analytics · Clustering analysis · Hands-on learning · Visualization

✉ Karolína Dočkalová Burská
burska@mail.muni.cz

Jakub Rudolf Mlynárik
445304@mail.muni.cz

Radek Ošlejšek
oslejsek@mail.muni.cz

[1] Faculty of Informatics, Masaryk University, Botanická 68a, Brno 60200, Czech Republic

Published online: 13 February 2024

1 Introduction

The shortage of cybersecurity workforce poses a critical danger for current companies ((ISC)^2 2022). As cybersecurity skills require higher-order thinking (McMurtrey et al., 2008), the best way to develop and ameliorate these abilities is through practical exercises that help raise awareness and competence and improve students' cybersecurity skills.

Regardless of the educational subject, tutors make intensive efforts to create, organize, and continually improve their hands-on courses. In contrast to many learning areas that produce tangible output suitable for checking, analysis, or assessment, e.g., a code of programming courses, practical cybersecurity training evinces a strong process-oriented characteristic. Tasks like "search for a vulnerability on server X" produce only sparse behavioral data that limit tutors' understanding of what trainees were really doing to solve the task. Therefore, we strive to support their endeavor by developing learning analytics tools to help tutors of cybersecurity hands-on exercises learn from conducted training sessions. Moreover, we apply methods of visual analytics to design and deliver easy-to-use analytical applications usable in practice.

1.1 Cybersecurity training background and limitations

Cybersecurity education can take many forms, from table-tops and online quizzes to hands-on drills. Our approach is based on data collected from hands-on training sessions organized in so-called cyber ranges (Knüpfer et al., 2020; Ukwandu et al., 2020; Yamin et al., 2020; Chouliaras et al., 2021). They serve as safe virtual environments emulating computer networks and enabling a data analyst to gather traces of trainees' behavior. The data has the form of event logs that can be further aggregated into relevant higher-level features for clustering.

However, even in the area of practical cyber exercises organized in cyber ranges, there are significant differences. Some of them follow free structure and rules, aiming to mimic real conditions. Typically, so-called cyber-defense exercises (CDX) are intended to train professionals (Eagle, 2013; Dasgupta et al., 2013). These competitions involve many teams like blue teams of defenders, red teams of attackers, or white teams responsible for the organization and compliance with rules. The complex scenarios of CDXs and many involved user roles introduce extremely variable behavior. On the contrary, training of beginners, typically students, often follows puzzle-based gamification principles of the educational content, where puzzles are used as a metaphor for getting students to think about how to frame and solve unstructured problems (Michalewicz & Michalewicz, 2008). In cyber security, such exercises are referred to as Capture the Flag (CTF) games (Werther et al., 2011; Davis et al., 2014; Švábenský et al., 2018; Kucek & Leitner, 2020).

In what follows, we focus primarily on puzzle-based hands-on exercises organized as time-restricted (usually supervised) training sessions. The formal puzzle-based structure of CTFs enables us to select relevant features for clustering-based analytical methods and overcome the gap between the raw data and analytical goals.

1.2 Analytical background and challenges

Data analysis can be conducted in different phases of a training life cycle. Based on the classification provided by Ošlejšek et al. (2021), objectives of this paper address the post-training analysis of the *quality of training exercise* (V_4) and *behavior analysis* (V_5). We apply clustering methods on behavioral event logs collected during the exercise to find common correlations (often subtle) in the behavior of groups of trainees.

Clustering techniques are among the unsupervised machine learning methods used to group data features by their similarity (Madhulatha, 2012). Their potential use in education is to identify typical or exceptional behavior of students, which may not be immediately obvious from individual data records. Group clusters and outliers observed in the data can raise hypotheses about used training strategies or features of training scenarios that analysts can further explore. Behavioral patterns revealed from cybersecurity exercises could be used, for instance, to estimate trainees' cybersecurity skills and the effectiveness of their actions, unveil attack-defense strategies, or identify possible issues in training scenarios.

Clustering methods deal with features extracted from raw data. For example, if we have the Bash commands each trainee used in a Linux server to protect it (i.e., the raw data), we can extract a feature like *the number of commands* and use it to cluster trainees into groups with respect to their efficiency (the fewer commands used to protect the server, the more effective the trainee was). Multiple features are usually combined to get meaningful behavioral clusters.

Even this trivial example demonstrates that features (*the number of commands*) and analytical goals (analyzing the efficiency of trainees in protecting the server) go hand-in-hand. Available raw data limits possible features and then possible analytical goals, and vice versa. Matching them up is challenging and requires employing some iterative strategy.

Also, the feature extraction process itself can be limiting, as it must be computed automatically. Suppose, for instance, we have individual Bash commands used for some cybersecurity task in the raw dataset. While counting them is simple and straightforward, what if we would like to define the *correctness of command sequence to protect the server* feature? In this case, assessing the correctness of an arbitrary sequence of commands algorithmically can be very difficult. Therefore, this feature can be considered too ambiguous and practically unusable for automated clustering.

This short discussion demonstrates that the definition of realistic analytical goals backed by available data and automatically retrievable features is challenging. In this paper, we apply an iterative visual-analytics development process to define analytical goals for data from CTF training sessions and to design and evaluate a practically usable analytical application.

1.3 Objectives

We aim to contribute to the state of the art of behavioral analysis in education practice with the following objectives:

- The formulation of analytical goals and related clustering method for the post-training analysis of hands-on cybersecurity Capture the Flag games.
- The design of an exploratory visual-analytics tool to support tutors in clustering-based behavioral analysis of trainees.
- The evaluation of the practical usability of the clustering method and visualizations.

Different stakeholders can benefit from the research. Mainly:

- *Researchers* can follow our approach to define additional analytical goals related to hands-on cybersecurity education or to apply described principles in different educational domains.
- *Developers of cybersecurity training platforms* can adopt and integrate our tool into their analytical dashboards.
- *Tutors of CTF training sessions* can use the proposed clustering-based analysis to spot their missteps during their tutoring. Or to explore exceptional or typical behavior of trainees.
- *Designers of training content* can use the proposed clustering-based analysis to notice an inaccurate or faulty training design.

1.4 Research method

Our approach to the clustering-based behavioral analysis of CTF trainees is based on the conceptual model for the visual analysis process Sacha et al. (2014), which is characterized by the interaction between data, visualizations, models of the data, and users discovering knowledge, as shown in the right-hand side of Fig. 1. The idea lies in automatically extracting features from raw data for a suitable clustering algorithm (the *model* part in the Figure) and gaining knowledge about trainees' behavior by interactively adjusting and exploring the data and clustering results via intuitive visualizations.

We applied the *Nested Model* (Munzner, 2009; Meyer et al., 2012) methodology to propose a relevant clustering method and deliver practically usable exploratory visualizations. This methodology guides designers of visual analytics tools through

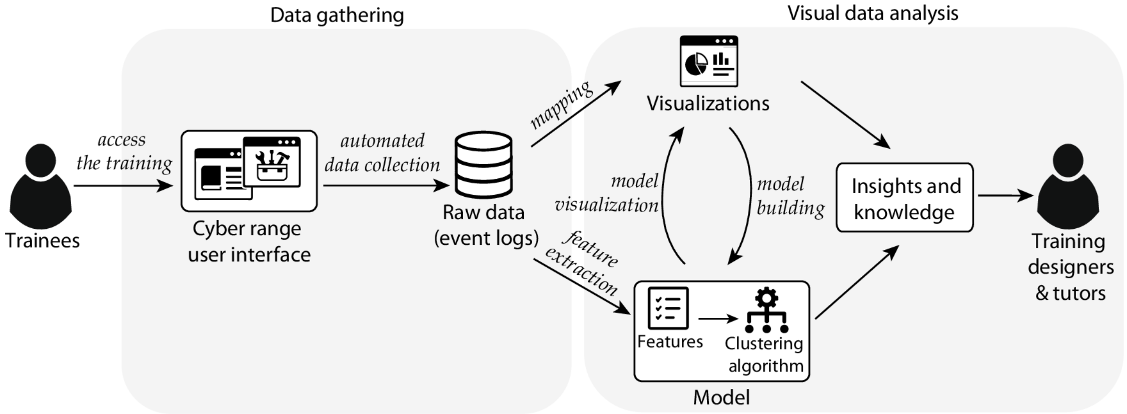

Fig. 1 The depiction of the whole data clustering process, including data collection, extraction, and application of the clustering algorithm

the whole process and enables them to independently validate each of its layers. The nested model consists of four phases:

Domain problem and data characterization aims to get familiar with the target domain. At this stage, we benefited from close collaboration with domain experts – tutors of hands-on cybersecurity courses, who gave us the necessary insight into their needs and actions. We conducted unstructured interviews and field observations of the hands-on cybersecurity training sessions. Furthermore, we collected data from the training sessions that consisted of succeeding tasks to solve. We focused on the training events described in Section 3.1 in more detail. A qualitative evaluation was then held to validate the fulfillment of the initial needs.

Operation and data type abstraction aims at mapping the input problems onto a more specific description. We identified the main needs of the organizers of the training sessions and transformed them into three analytical goals posed in Section 3.2. Each goal focuses on a different aggregation technique that utilizes data clusters to help identify three types of training data outcomes. After identifying those main areas in the form of requirements, we needed to determine the main measures that could help align the necessary training values in the form of features. We selected six of them, as defined in Section 3.3. A Simple Ease Question (SEQ) questionnaire was used to measure the outcomes related to the use of the tools.

Visual encoding and interaction design aims to interconnect visualization elements with interaction strategies. Once having the necessary characteristics, we had to encode the data into a suitable visual representation. As a result, we designed two types of visualization that deal with clustered data. They are described in Section 4. We then used the System Usability Scale (SUS) to measure the usability of the tool in the given context. The whole evaluation is discussed in detail in Section 5.

Algorithm design aims at carrying out the implementation of the visual encoding. We selected the unsupervised machine learning algorithm *k-means*++, which clusters data according to the measures (features) from the previous steps. To validate the selected approach, we measured the performance of the algorithm on different sizes of datasets. The results are discussed in Section 5.5.

2 Related work

Clustering is an essential part of data mining. The generic state-of-the-art overview of traditional and recently proposed clustering methods and their application domains can be found in Ezugwu et al. (2022).

Educational Data Mining (EDM) is an emerging discipline that exploits statistical, machine learning, and data mining algorithms over the different types of educational data (Romero & Ventura, 2010; Salloum et al., 2020). Dutt et al. (2017) provide a comprehensive overview of EDM techniques. Their educational data clustering process explains important steps in the design of clustering approaches. We adopt these steps within the nested model to design and validate visualization systems (Munzner, 2009), which we used to develop a practical exploratory tool for cybersecurity education.

A number of approaches aim to identify the effectiveness or pinpoint distinct student strategies in specific types of courses. Specific solutions can be found in the literature addressing, for instance, student performance in generic courses (Durairaj & Vijitha, 2014), classification of students of small online courses by features adopted from business systems (Wang, 2021), revealing patterns of engagement in massive open online courses (Khalil & Ebner, 2017), or understanding how students approached solving a particular programming problem (Yin et al., 2015).

In the area of cybersecurity education, which is the primary subject of our research, Švábenský et al. (2022) applied techniques of pattern mining and clustering to analyze the usage of command-line tools in hands-on cybersecurity exercises, aiming to support the automated assessment of students. Our solution focuses on different aspects of EDM – revealing gameplay strategies and possible flaws in the training content.

Despite the primary focus on fairness, the recent survey paper Le Quy et al. (2023) brings a useful classification of EDS tasks that use clustering models. Among them is the category “students’ behavior, interaction, engagement, motivation, and emotion,” which our research falls into and which provides a comprehensive overview of specific approaches, including the usage of k-mean clustering models.

Since clustering results can be influenced by the algorithms used, multiple studies compare the performance of clustering methods in EDM. DeFreitas and Bernard (2015) compared partition-based (k-means), density-based (DBSCAN), and hierarchical (BIRCH) methods to determine which technique is the most appropriate for performing clustering analysis within the Learning Management Systems (LMS), e.g., Moodle. Hooshyar et al. (2020) proposed an automatic comparative approach utilizing multiple internal and external performance measures to compare and accordingly recommend the most suitable clustering method for each LMS dataset.

The results of these studies indicate that the performance of clustering algorithms vary depending on the type of data, its size, and the performance measures being used. Moreover, these studies focus on LMS data from long-term courses. Our application is specific in that we do not use data from a general LSM but particular data from hands-on CTF games. Therefore, we chose the *k-means++* algorithm (Arthur & Vassilvitskii, 2006) – an improved version of k-means partitioning (Lloyd, 1982). This partition-based approach belongs to the top 10 algorithms in data mining (Wu et al., 2008) and also the most frequently used clustering methods by EDM and learning analytics (Dutt et al., 2017).

3 Data clustering

In this section, we explore three components of the visual analysis process (Fig. 1) that are required for the successful design of appropriate visual-analysis tool: *Raw data*, *Model* (i.e., features and clustering algorithm), and the formulation of analytical goals suitable to obtain *Insights and knowledge*.

3.1 Raw training data

Modern cyber ranges usually collect data in the form of event logs triggered by the trainees in the platform during the training session. For example, the Locust 3302 CTF game (Švábenský et al., 2018) is split into six consequent levels, each representing a single cybersecurity task – puzzles from the puzzle-based gamification perspective. The goal of the tasks is to

- scan a computer network,
- search for vulnerabilities on a web server,
- exploit the server,
- crack the SSH password,
- use the SSH password to steal data.

Successful completion of one task is required before proceeding to the next one. Trainees can receive various hints or a complete step-by-step solution.

The gathering of raw data is depicted on the left-hand side of Fig. 1, where a trainee interacts with the cyber range, and the interactions are stored in the form of event logs in a database. In general, the puzzle-based gamification produces two types of events that could be used to track the behavior of individuals and classify them: (a) logs capturing the game state, e.g., when a trainee took a hint or finished the task, and (b) commands used to find a solution of a task, e.g., the `nmap` used for scanning the network.

We used the open-source KYPO Cyber Range Platform (KYPO CRP) [Vykopal et al. 2017] to collect data from multiple games and design the clustering methods. Types of events produced by this cyber range are discussed in detail in Macák et al. (2022). They include game events, command histories, and the usage of the Metasploit tool. Considering the analytical goals outlined in Section 3.2, we utilize only a selected subset of game events for the clustering, omitting other data collected by the cyber range. We show that even with such limited data, we can obtain relevant clusters useful for learning analytics. Nevertheless, the unused data can be employed in the future to address other analytical goals, simply by following the same principles as discussed in the remainder of the paper.

Events used for clustering are summarized in Table 1 and the whole dataset is also available among the supplementary materials. The events are related to the higher-level abstraction of puzzle-based gamification. The *LevelStarted* and *LevelCompleted* events encode the start and end of each puzzle – a cybersecurity task called game level. If a task is successfully solved, the trainee finds a flag (hidden text), which is used to proceed to the next level. The *CorrectFlagSubmitted* and *WrongFlagSubmitted* events capture trainees' attempts to proceed to the next level. The assessment aspects of exercises are captured by the *HintTaken* and *SolutionDisplayed* events that are penalized.

All events are equipped with a timestamp, absolute training time, trainee ID, and level ID to trace walkthroughs of individual trainees. Besides this, additional records that vary depending on the event's type can be present. Relevant attribute types are summarized in the last column of Table 1 and used in the analytical workflow either for feature extraction, data clustering, or exploratory visualizations.

Table 1 Training events and their meaning

Event	Description	Event-specific records
LevelStarted	The trainee started a new level.	maximum achievable score
LevelCompleted	The trainee successfully finished the level.	awarded score
WrongFlagSubmitted	The trainee submitted a wrong flag.	provided flag (text), correct flag (text), penalty score
CorrectFlagSubmitted	The trainee submitted a correct flag.	provided flag (text)
HintTaken	The trainee took a hint.	hint title, hint wording, penalty score
SolutionDisplayed	The trainee viewed a complete task solution.	level ID, solution wording, penalty score

3.2 Analytical goals

Based on our long-term collaboration with domain experts on cybersecurity education, we identified three areas where a clustering tool could help improve the impact of current hands-on training programs on trainees (the *insight and knowledge* artifact in Fig. 1) and, simultaneously, for which common cyber ranges produce relevant event logs. We formulated them into three analytical goals.

G1: Examine typical gameplay strategies. The gamification process reflects the initial vision of a training designer who transfers ideas into game elements like abstraction, challenges, and rules (Kapp, 2012). However, the gameplay of real trainees often differs from these expectations as the users adapt to real-time conditions (time press, assessment rules, etc.) or their knowledge (e.g., using commands or steps leading to solving a task in an unexpected way).

Traditionally, the divergence of users' behavior from the expected one is analyzed by using so-called conformance checking (van der Aalst, 2016; Weiss et al., 2016; Švábenský et al., 2022), where a model of expected behavior is required, which makes these approaches laborious and prone to errors. On the contrary, clustering could reveal different gameplay strategies without this prior knowledge. The idea lies in the computation of behavioral clusters related to common gameplay strategies, paying attention to bigger clusters since they represent significant "herd behavior."

For example, suppose that we are able to cluster trainees by their dependency on using hints. If a significant group of trainees in the training session prefers taking hints while solving tasks, then this behavior can be considered a gameplay strategy indicating that taking hints can be more advantageous than making own effort to find the correct solution. On the contrary, if no such significant group is observable or there are only several individuals with different patterns of using hints (e.g., in different tasks), then the analyst would conclude that the training session does not evince a significant strategy related to using hints.

G2: Identify flaws in training design. Proposing educational content is a creative process that is always prone to errors. Recurring behavioral patterns in trainees' progression (such as the inability to solve a task even after using hints) may help the training designers to find flaws in the training design (e.g., the hint is useless or confusing). Identifying such situations can lead to more precise task delimitation and thus improve the quality of the training.

Clustering can help identify such situations and allow training designers to adapt game parameters (time limits, assessment rules, etc.) or game content (e.g., the wording of hints). Flaws can be observed using two approaches.

First, an analyst could extract specific flaw-relevant features from the raw data and use the clustering to determine whether a significant group of trainees struggled from such a potential flaw. For example, features encoding the "frequency of displaying solution at the very last level" could be used to indicate insufficient time allocated for the training (with respect to trainees' skills).

Another flaw-detection approach is based on *G1* and expert opinion on gameplay strategies. In this case, a training designer can intentionally search for clusters of gameplay strategies that might indicate some trouble. This approach could be more practical because we usually aim to identify unknown flaws.

Regardless of analytical workflow, more attention should be paid to larger clusters because they represent significant behavior (a shortcoming encountered by multiple trainees).

G3: Identify outliers. While the analytical goals *G1* and *G2* primarily target majority behavior, finding individuals or small groups with certain characteristics is also important. It is beneficial to identify exceptionally skilled people or rare gameplay strategies that are not desired in a certain course or scenario, for instance. However, this sparse behavior is often hidden in the amount of data, which makes its identification difficult. Therefore, the analytical tool has to provide a solution for these contradictory requirements: searching for typical or rare behavioral patterns.

Identification of outliers represents a mixture of *G1* and *G2* analytical workflows. Analysts should search for clusters related to gameplay strategies or the identification of flaws, be able to recognize small clusters (or individuals), and then assess their importance either using expert knowledge or comparing revealed behavior with "herd behavior" (i.e., bigger clusters). The solution to this issue lies in using proper visualization techniques that can emphasize small clusters and outliers alongside significant groups of trainees.

3.3 Extracted features

Considering analytical goals *G1–G3* and raw data produced by the cyber range, the following features reflecting trainees' game-play style were chosen for automated extraction. They can be computed for the whole training, e.g., the total number of hints taken by a trainee during the training session, or for a selected level, i.e., the number of hints taken by a trainee at level X. Which option is used depends on the granularity of the analysis, as discussed in the tool design in Section 4.

Defined features can be divided into three categories. First, we chose two main features that can be easily extracted from logs by counting specific events of individual trainees:

- **F1: The number of submitted wrong flags.** A high number can highlight trainees who could not reach the milestone. This may be due to an intentional strategy where the trainee is trying to guess the correct answer (*G1*) or as a result of a flaw, e.g., confusing task description or hints, where the trainee struggles with finding the correct solution, being convinced of the correct flag (*G2*).
- **F2: The number of taken hints.** If a trainee uses more hints to find the solution than others, it can again indicate either intention or trouble. It can be the result of the trainee's deliberate strategy, where he or she wants to go through the game or a level with minimum effort (*G1*), or the consequence of a wrong game design, e.g., a useless hints or too difficult assignment (*G2*).

While the previous features represent general statistics that can be easily obtained from raw data by counting corresponding event logs, the following two features have to be computed by analyzing sequences of events in individual trainees' walkthroughs:

- **F3: Time spent after using a hint.** This feature calculates the time between taking a last hint and providing the correct flag, i.e., solving the task. If the data from the whole game are examined, not just from one level, the value is obtained by averaging the trainee's values across all tasks. In both cases, long times should indicate the uselessness of hints (*G2*).
- **F4: Wrong flags submitted after a hint:** Similarly to the previous feature, if a hint is useful, it should not be accompanied by many wrong flags afterward. Otherwise, it may indicate a flaw in the game design (*G2*). When examining the whole game, the values from individual levels are averaged.

Since both *F3* and *F4* strictly indicate design flaws, their absence or presence in join clusters with *F1* and *F2* can help analysts to assess whether the trainees' behavior is due to errors in the game design (*G2*) or deliberate strategies (*G1*). Although the features *F1–F4* do not mention *G3* explicitly, they are also relevant to identifying outliers. The only difference is in looking for rare vs. obvious behavioral patterns.

The last two features introduce unifying indicators of the overall trainees' success or failure:

- **F5: The total time played.** The differences in the amount of time spent playing CTF can indicate different skills or interests. Extremely long or short playing time should always attract an analyst's attention because it can indicate a talented or indifferent individual (*G3*). However, a shorter time does not automatically mean a better trainee. A frustrated trainee, for instance, could use hits or solutions to go through the game as quickly as possible without any effort. Only if this feature is used with other features, then a real reason can be observed. On the other hand, the appearance of a long playtime together with a high number of taken hints (*F1*) is quite obvious because the reason for taking hints is often the lack of time, typically at the end of the training session. If this behavior is exceptional in a group of trainees, then we could interpret it like an intentional *"do my best during the game, then use hints when the end approaches"* strategy (*G1*). On the contrary, if this behavior is typical, then it could be interpreted instead as a design flaw (the training scenario is too complex for the allocated time).

- **F6: The total score.** A score earned by trainees can give a straightforward insight into the overall trainee success since it includes points for successful levels or penalties for used hints or for skipped levels. While the total time play provides only a very simplified view of the trainees' success, a total score introduces a more precise assessment. Besides the use cases mentioned in *F5*, the combination of total time and total score brings yet another analytical possibility. Trainees who quickly gain a high score can be considered talented or skilled, and vice versa. This makes their discovery easier.

3.4 Clustering method

As an unsupervised data mining technique, clustering of the provided data does not require pretraining models and, therefore, does not require human intervention. Many approaches to clustering exist nowadays. We can divide them into several categories – the most widely used include overlapping, partitional, and hierarchical clustering (Rai & Singh, 2010). At the same time, a number of comparisons of these methods is available (e.g., Rodriguez et al., 2019; Gelbard et al.,2007; Fraley and Raftery, 1998) for various requirements.

In our case, we aimed at the technique as a proof-of-concept, which would help us determine the fit of the whole approach of clustering for cybersecurity training data analysis. Partition-based algorithms are widely used in various fields because of their easy implementation (MacQueen et al., 1967). The most typical partitional method is K-means (Jain, 2010). The K-means algorithm is useful for our use case since it can adapt to sparse matrix data sets and efficiently organize large data sets. It is also suitable for numerical values that we use because it measures the squared Euclidean distances in the clustered data. However, the number of clusters and the selection of initial centers can significantly impact the clustering results of the K-means algorithm.

In our solution, we therefore use an improved *K-means++* variant (Arthur & Vassilvitskii, 2006) that provides better results. More specifically, it does not allocate all the cluster centers randomly. Instead, it chooses the first centroid randomly and then selects the remaining clusters from the rest of the points with probability proportional to its squared distance from the point's closest existing cluster center.

The algorithm requires specifying how to calculate the similarity of features. As the features *F1*–*F6* represent numbers, their combination defines the points in Euclidean space that can be measured by the Euclidean distance.

The algorithm takes the desired number of clusters k and points (features) to be classified as input. It divides the data records into k classes, starting with randomly selecting k data points as cluster centers. It then improves the clustering results by repetitively recalculating the centers of clusters by averaging cluster members.

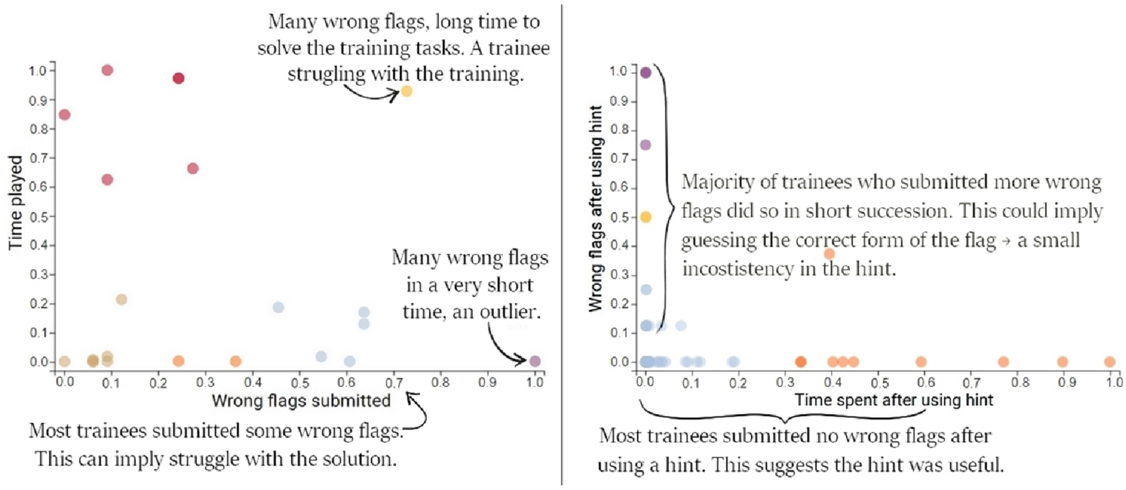

Fig. 2 Scatter plot visualization. On the left-hand side, it displays the wrong flags submitted in relation to the time of gameplay. It is distributed in 5 clusters. The right-hand side (with 4 clusters) shows how many wrong flags the trainees submit after asking for a hint

4 Visual-analysis tool

To support clustering-based post-training analysis covering goals *G1–G3*, we designed and implemented an exploratory tool[1]. Event logs generated by the KYPO Cyber Range are stored in the ElasticSearch no-SQL database. Features *F1–F6* are extracted by transforming and aggregating the raw event logs. The developed API unifies the aggregation services so that new features can be integrated in the future.

Raw data and data clusters produced by the *k-means*++ algorithm are consumed by several complementary visualizations (the *Visualizations* component in Fig. 1). Their full integration into the open-source cyber range provides an off-the-shelf selection of training sessions and their game levels, making the analysis comfortable and available right after a training session. This section explains key visualization principles and design decisions on several examples.

The analytical tool provides two primary views, both equipped with an interactive estimation of optimal clusters. Both visualizations are discussed in what follows.

4.1 Scatter plots

The scatter plot views provide a detailed comparison of a pair of features, as shown in two examples in Fig. 2. Points represent individual trainees, while the color denotes the clusters identified by the clustering algorithm from the distribution of points on the chart (their x and y coordinates). Points are semitransparent – a darker shade of the same color indicates multiple trainees with the same feature values. Axes are normalized according to data samples to provide relative values. Therefore, number one represents the time of the slowest trainee or the maximum number of wrong flags submitted by some trainee, for instance.

[1] The link to the source code, together with supplementary materials, is available at https://eait.surge.sh/.

The tool predefines plots of two specific pairs of features. They were selected in accordance with the domain experts' preferences, but other pairs can be easily integrated or selected dynamically. Both pairs primarily address possible flaws in the game settings (*G2*).

Wrong flags vs. time

A scatter plot dealing with the *wrong flags submitted (F1)* and *time played (F5)* features show how straightforward or confusing the training or task was in general or for individuals. For example, many wrong flags submitted over a long time period (the upper-right quadrant of the graph) can indicate that the task was rather difficult for trainees. Many wrong flags submitted over a short time period (points located near the right half of the x-axis) can indicate trainees who try to guess the correct flag. Points located close to the y-axis indicate trainees who were rather successful in finding a correct solution (in variable time).

To infer hypotheses or conclusions about behavioral aspects, the analyst has to consider the distribution of individual points and whole clusters on the chart. In the left-hand side scatter plot in Fig. 2, one can see two outliers – the yellow and violet clusters with a single trainee. Their positions on the right-hand side of the graph show they had trouble completing the training. Especially the most-right violet outlier close to the x-axis indicates suspicious behavior as the trainee has finished the training very quickly, using many wrong flags. In general, since many trainees submitted many wrong flags in a relatively short time (the lower-right quadrant of the graph), it might point to some inaccuracy in the task assignment.

Wrong flags vs. time after using a hint

The combination of *wrong flags after using hint (F4)* and *time spent after using a hint (F3)* features shows that if a trainee used a hint, how useful it was. The main assumption is that once a trainee reads a hint, the solution should be more straightforward, with only a minority of succeeding wrong flags. On the contrary, when many trainees still struggle with the solution, the situation can indicate a faulty or insufficiently explained hint.

Like the previous scatter plot view, this one also helps point out the wrong parameters of the training *G2* or discover possible outliers who submit too many wrong flags even after taking a hint *G3*.

The right view in Fig. 2 shows an example. As the graph contains only trainees who took any hint when solving a task, fewer points indicate a simpler task and vice versa. Points close to the y-axis can indicate a possibly confusing hint where the trainees could finish the task quickly at the cost of repeatedly providing an incorrect flag. On the contrary, points located near the horizontal x-axis can be interpreted as the existence of useful hints that lead to a correct solution without mistakes. The time of finding the solution (the distribution of points along the x axis) is not that important in this case.

Therefore, it might appear that a chart with points predominantly located at the bottom part could be considered a well-designed game level. However, it is true only for games that are organized to teach new cybersecurity concepts. Other types of hands-on sessions can produce different expected distributions. For instance, in tests or competitions, multiple wrong flags in a short time would be considered expected behavior due to time pressure. Therefore, the analyst needs to decide what is obvious.

4.2 Radar charts

The radar chart view depicted in Fig. 3 represents a dominant visualization for the analysis of gameplay strategies. Unlike in scatter plots of two values, multivariate features are captured compactly as two-dimensional volumes that clearly visualize commonalities between samples and help recognize more compound strategies or individual outliers (Chambers et al., 2018).

Similarly to the scatter plot views, either the whole game results or the results of selected game levels can be chosen. The color shapes help to distinguish different strategies visually. The number of shapes corresponds to the number of computed

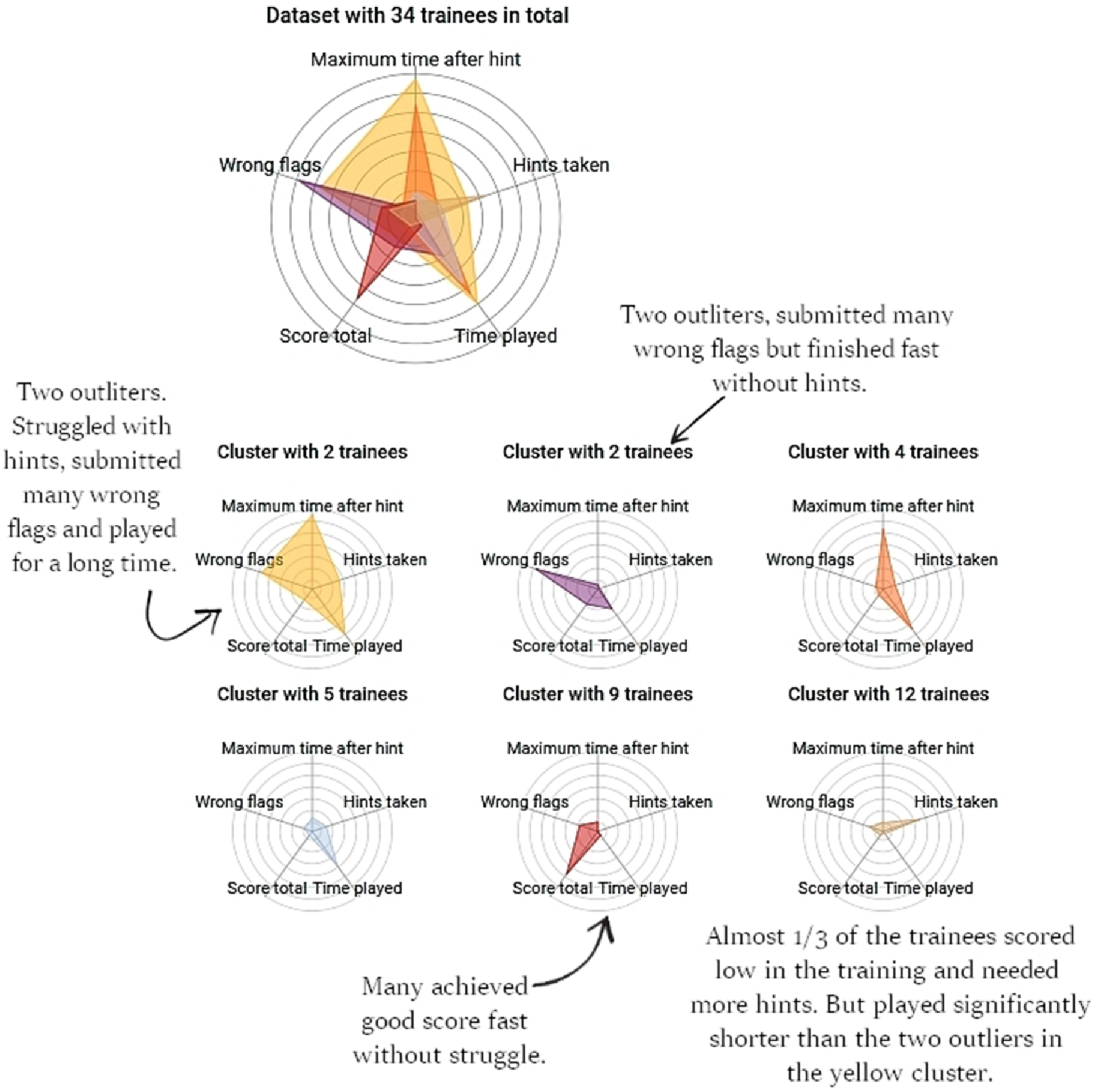

Fig. 3 The radar charts view. The main upper chart shows all the computed clusters combined, while the small charts below enable a better examination of individual clusters

clusters. The number of trainees in each cluster indicates groups of trainees using the same strategy. It helps the analyst assess the cluster's significance (typical vs. exceptional behavior).

In the radar chart in Fig. 3, six clusters were selected, resulting in two small groups – potential outliers. The yellow cluster reveals two trainees who played for a significantly long time and took hints that were not very helpful. In contrast, the two trainees from the violet cluster played relatively fast but submitted noticeably many wrong flags without trying to take any hints. Nine trainees out of 34 performed very well. They achieved good scores fast without struggle (the red cluster). Almost one-third of trainees scored low in training and needed more hints. But they played significantly shorter than the two outliers from the yellow cluster.

4.3 Elbow function

Both the scatter plot and radar chart views require the analyst to specify the number of clusters. Selecting them ad-hoc iteratively and inspecting obtained results is not a very efficient workflow. Therefore, we introduced a helping *elbow function* visualization aiming to support this crucial analytical decision.

Finding an optimal *k* for k-means clustering is based on finding the sum of the square distance between points in a cluster and the cluster centroid (Nainggolan et al., 2019). Drawing these values in a line chart allows the analyst to identify an elbow point where the curve is refracting. This point can be used as an initial number of clusters for the exploratory analysis. In the example in Fig. 4, clusters of sizes 4 and 5 are emphasized as candidates for the initial exploration. The elbow graphs are automatically computed for all scatter plots and radar chart views.

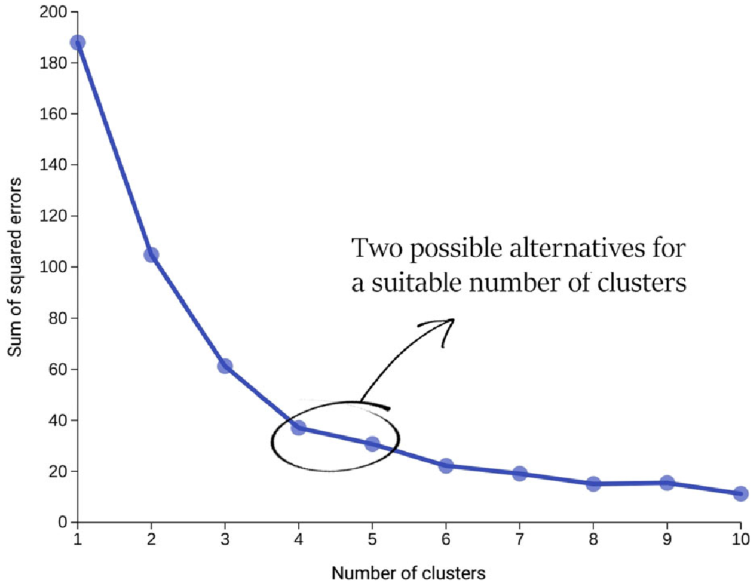

Fig. 4 A helper line chart representing the *elbow function*. It serves for the selection of an initial number of clusters for the analysis. In this case, 4 or 5 can be selected as suitable values

5 Evaluation

We conducted a qualitative user evaluation to receive feedback on the tool's usage and to verify that it provides the information outlined by the initial goals. Additionally, we evaluated the tool's usability and usefulness and gathered valuable remarks for further improvements.

5.1 Participants

The evaluation includes nine target users (P1 – P9). Due to the necessity of background domain knowledge, they need experience in designing and organizing cybersecurity training sessions. All the participants were familiar with the concept and design process of cybersecurity training in the Cyber Range and used the platform to conduct or design various types of educational training. Details of participants are summarized in Table 2.

5.2 Procedure

The user study sessions were held individually, in person for seven participants, and online (using MS Teams) for the remaining two. Each session lasted about 60 minutes and had four parts.

In the introductory part, the experimenter explained the evaluation procedure, and the participant consented and filled out the demography questionnaire. In the second, the familiarization phase, the experimenter presented the tool, and the participant spent 2-3 minutes familiarizing themselves with it using a demonstration dataset. Next, the respondent performed eight predefined analytical tasks (Table 3) that were formulated to cover all the goals put on the tool.

Because of the relatively small size of the participants' group, we decided to focus on inputs beyond simple textual feedback. The tasks do not have strictly correct answers and were purposefully formulated to require a more thorough justification. This ensures

Table 2 Demographic summary of the participants

ID	Age	Gender	Position	LE	OE	VE
P1	38	M	Senior lecturer, Researcher	5	>20	4
P2	36	M	Lecturer, Manager	5	>20	5
P3	29	M	Senior lecturer, Researcher	5	>20	5
P4	30	F	Seminar tutor, Researcher	4	<10	1
P5	30	M	Analyst, Tutor	4	<20	4
P6	41	M	Forensics Analyst, Lecturer	5	>20	3
P7	34	M	Data analyst, Lecturer	4	<10	3
P8	25	F	Training designer, Lecturer	3	<10	3
P9	46	M	Researcher, Seminar tutor	3	<2	5

LE – Lecturing experience, OE – Exercise organization experience, VE – Experience with analytical visualizations

Table 3 The tasks used for the evaluation

Task 1:	In the ‘Wrong flags per time played’ view, identify the most appropriate elbow method number in the helper elbow chart.
Task 2:	In the ’Wrong flags per time played’ view (for all levels), do you see any suspicious trainees? Why/why not? If so, what is the trainee ID?
Task 3:	In the ’Wrong flags per time played’ view (for level 5), what could the results imply regarding the level design?
Task 4:	In the ’Time spent after using the hint’ view (for all levels), what does the point distribution suggest? Does it imply a good training design or bad?
Task 5:	In the ’Radar chart’ (for all levels), are there any clusters that represent distinct strategies but share similar training success?
Task 6:	In the ’Radar chart’ (for all levels), are there any possible outliers?
Task 7:	In the ’Radar chart’, how variable is overall success of the trainees across the clusters?
Task 8:	In the dataset of ’Hacking Day Cyber Task Force Delta’ (all levels), determine which strategy (which cluster of trainees) was the most successful.

that, apart from the inputs of the answers, we can get more insight regarding the distinct understanding of concepts such as “success” or “good training design”. Therefore, the respondents were asked to comment on their actions and interpretation of the results. The experimenter took notes and recorded the screen and audio with the participant’s opinions and thoughts for further qualitative evaluation. In addition, the difficulty of each task was also formally evaluated using an SEQ – Single Ease Question (Sauro & Dumas, 2009) questionnaire to validate the process of mapping the abstract problem onto a specific visual form.

Lastly, for a complete assessment of the tool’s usability, we combined SEQ with the SUS – System Usability Scale (Sauro, 2011) metric. The SUS questionnaire helps us rate the overall design of visual encoding and exploratory interactions.

5.3 Datasets

We used a total of three datasets, hereafter referred to as DS1 – DS3. They were collected in past hands-on training sessions. All the datasets contain various events that occur during training: submission of a wrong or correct flag, taking a hint, finishing a level, not interacting with the training portal, etc. To avoid information bias, we used DS1 exclusively to introduce the visualizations and their capabilities to the respondents, while DS2 and DS3 were used for the evaluation itself.

DS2, on which the majority of the tasks were performed, contains data collected from 34 trainees of a training seminar with an attack-oriented scenario. The training definition contains six training levels in which the participants attempt to scan a server for vulnerabilities and exploit it. The session lasted seven days, during which 1741 events were collected.

The DS3 was used to determine the usability of our approach on a small dataset with only seven trainees and 121 collected events. The exercise was a 90-minute-long hacking competition with a similar scenario as the above but reduced to five levels.

5.4 Usability results

We compared the respective answers from SUS questionnaires to assess the overall usability of the proposed analytical tool. The obtained score of 75 lies in the interval from 68 to 80.3, which fits the **good** rating category according to Bangor et al. (2009).

While SUS evaluation addresses the overall usability, SEQ scores reflect the difficulty of solving individual tasks. The result summarized in Fig. 5 reveals that solving all tasks was rather easy (more than half of respondents ranked them from neutral to very easy). The lowest rank was assigned to Task 5 (an average score of 3.6), which can be considered the most difficult. Other tasks except Task 4 achieved very high average scores with values above five on the 7-point scale. Task 8, with an average score of 6.0, was rated as the most simple when using our tool. Therefore, also the SEQ results confirm an overall *good usability of the tool* for solving the tasks.

To access participants' insight gained into individual tasks during the data exploration, we observed how consistent their answers, comments, and recorded interactions are regarding the original analytical goals *G1–G3*. In what follows, we summarize our observations for individual goals. Task 1 is specific as it is related to all the goals. It can be considered an introductory task making respondents aware that they can change the desired number of clusters at any time.

Insight into the examination of gameplay strategies

The aim of the first analytical goal is to examine typical gameplay strategies. Our objective was to confirm that the majority of participants would state similar outcomes regarding the gameplay, the most often recurring behaviors, or the variability of the playing strategies.

This goal was covered by the radar chart view, to which Tasks 5, 7, and 8 were related. Radar charts in Fig. 3 illustrate the situation from the evaluation. However, it must be remembered that the evaluation is dynamic, and a particular view depends on

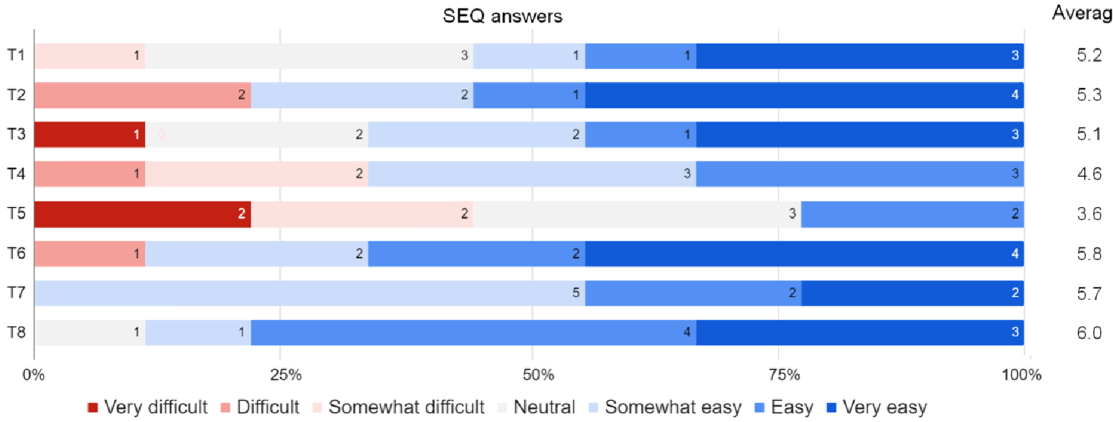

Fig. 5 SEQ score for individual tasks. The color scale rates each task from red – very difficult to blue – very easy. Numbers inside the color bars show the number of corresponding ratings. The numbers on the right-hand side of each task show average ratings (red = 1, blue = 7)

selected parameters, especially the number of desired clusters and whether a specific level or the entire training is examined.

To solve evaluation tasks, analysts must first clarify the meaning of "success" or "failure" in an exercise. Our participants assessed the success of computed clusters mainly by their *score total* and *time played* values, using other features as complementing. This approach confirms our expectations.

Task 7 was directly proposed to get the participants' remarks on the interpretation of the success. In general, most participants identified two groups of trainees: a majority that didn't perform well and one smaller group that was much more successful. Seven participants described the clusters as very variable, with different results. P7 identified four groups: normal, good, bad, and unusual. P3 measured the success by the *score total* axis length instead of comparing the clusters, thus ranking the success variability as lower. The rest of the participants correctly compared the clusters in relative scales between each other.

The ability to recognize and assess clusters with different degrees of training success implies that the participants chose the right number of desired clusters for the analysis. In Task 5, the goal was to identify distinct strategies leading to similar success or failure. The evaluation revealed that participants primarily compared the total *time played* with the number of *hints taken*, as these two features evinced significant differences among clusters with similar success.

Task 5 was rated as the most difficult (SEQ score 3.6). However, it was the first to work with the radar chart (before Task 7) in the evaluation process. All the succeeding tasks related to the radar chart were rated as easier (with SEQ scores ranging from 5.8 to 6.0). And because they were conducted with different datasets, it refutes the reason for the sudden rise of rating would be getting familiar with the data and thus subjectively seeing the tasks as easier. It rather suggests that users find it easier to comprehend encoded information after getting more familiar with the overall concept of the radar charts.

Task 8 was rather straightforward as all the participants pinpointed the same set of characteristics and selected the same groups of trainees. The participants measured success as a correlation between *score total* and *time played*. Some participants ignored the high number of *wrong flags* as a factor that should lower the success rate.

Overall, the participants were able to find the connections between the features in individual clusters that are related to distinct strategies and, moreover, to identify and name specific significant gameplay strategies. One of the often mentioned strategies was that some trainees were omitting the hints and were trying to pass the level on their own.

Insight into flaws in training design

The second goal focuses on *identifying flaws in training design*. It should help determine if there are any points where it is too hard for the trainees to solve the puzzle or, in contrast, some trainees get too good results too easily. This goal was covered by Tasks 3 and 4. Both of them relate to the scatter plot views, and their average SEQ score achieved 5.1 (Task 3) and 4.6 (Task 4).

Task 3 focuses on the relationship between the number of *wrong flags submitted* in a certain game level and the time of level playing (i.e., the *time played* feature). The answer to the question depended on how people perceive the quality of a training design with respect to the effort. In general, a ’good level design’ was mostly defined by participants as one in which the dots are adjacent to the y-axis (low number of wrong flags) and the level time is not too high (not too apparent in the current visualization – a time estimate could help according to one of the participants). The majority identified level 5 (depicted in Fig. 6) as quite easy as the significant amount of points lies in the left part of the chart, and it was considered a good sign for them (straightforward and balanced assignment).

The goal of Task 4 was to identify how the participants dealt with training design specifics related to hints. Data from the evaluation are depicted in Fig. 2 – right view. Clusters are computed for the entire game (averaged values across all levels), and they evince significant distribution close to both axes.

The interpretation of this distribution by respondents met our expectations. All the participants agreed on an equivalent response. They decided the design was rather good. They agreed that good design (related to hint usage) is denoted by a high concentration of dots on the bottom side (close to the x-axis), which suggests that once displaying the hint, there were not many successive issues (i.e., the hint was helpful). They noticed that after using a hint, the majority of trainees solved the level without too many wrong answers. Five participants would, however, analyze the hints further because some seem to be less useful for some trainees.

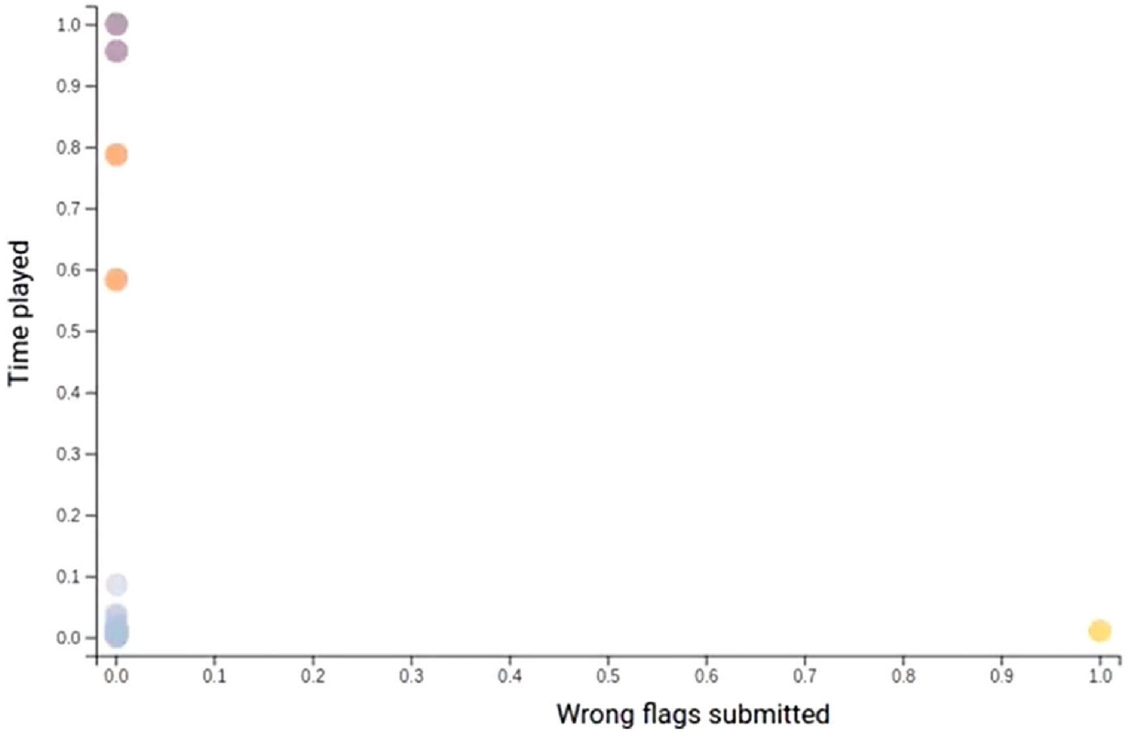

Fig. 6 Scatter plot with data for level 5 in evaluation Task 3

Identification of outliers

The third goal reflects the need to identify outliers among trainees. Both visualizations, i.e., the scatter plot and radar chart, have the potential to show them.

Task 2 focuses on the scatter plot view. More specifically, on spotting any suspicious trainees in terms of the number of submitted wrong flags. The majority of participants (7) noticed a trainee (the violet outlier on the left chart in Fig. 2) who was suspicious due to the very high number of flags per relatively short time. One participant did not notice the suspicious trainee at first because the chart boundaries were not clearly visible. Another trainee was reported for submitting a large number of flags and playing for a relatively short time (compared to the rest).

The radar-chart outlier discovery was evaluated via Task 6. Except for one participant, everyone was looking for small clusters (i.e., with a small number of trainees) that moreover had outstanding data. The cluster with a single trainee was the most frequently selected. Some also considered one with two trainees. However, a single participant (P4) who selected the same cluster as the rest additionally selected also a large cluster with 12 trainees. The stated reason was that it showed unusual behavior (high scores, low number of hints).

5.5 Scalability results

To find the limits of the selected algorithm in carrying out the clustering problem from the computational point of view, we measured and analyzed data volume bounds. The analysis was performed for the backend application (the Elasticsearch stack and Java application), which employs all the computationally intensive tasks. The tests were executed on a computer with the following specification: 32 GM RAM, Intel(R) Core(TM) i5-9600K processor, 512GB SSD M.2 disk, Windows 10 OS.

Performance analysis

The performance was evaluated by measuring the speed of the application (response time of each API request) and the utilization of resources, i.e., memory and CPU usage. Two tools were used for the testing to obtain these metrics. The first is the Apache JMeter, which measured the latency and overall sample time across different threads while several requests were executed. Since a single request would not provide a representative load, it puts the application under considerable stress resulting in more conclusive data.

The second tool, VisualVM, monitors Java applications while they are running on a Java Virtual Machine. Specifically, it can provide information about CPU usage as well as used heap or metaspace.

The tests for both tools consisted of batches of REST calls across different threads. While the tests in JMeter were running, VisualVM was used for monitoring. The first batch of tests was run on 50 threads executed in groups of five, totaling 300 calls in all tests. The chosen dataset for the test was a training definition with about 2300 records.

Another analysis run was performed with unrealistic experimental datasets to test the application with a higher amount of users. Specifically, the first test used a dataset of 100 users and 2652 records. Another experimental test was run with 1000 users and circa 10000 records.

Detailed results are available in Mlynárik (2022). In summary, most calls were finished in less than 200 milliseconds, with only a few exceptions. In the largest dataset, the results show an average value of 500 ms for cluster computation and 3 to 8 seconds for the sum of squared errors, which is used for optimal cluster number retrieval. It is expected that the calls for the sum of squared errors (which then enable the selection of the appropriate number of clusters) are more complex, and thus the response time is significantly longer. Anyway, measured times are suitable for the interactive exploration of data captured in real cybersecurity training sessions.

Lower bound analysis

The computation of relevant clusters requires a minimum amount of data. The estimations for a lower bound on the amount of data can be computed according to Formann (1984). The suggested minimum sample size should equal $2 \cdot d$, where d denotes the number of dimensions (features) used for the analysis. Preferably, the number should be higher, at $5 \cdot (2 \cdot d)$. This implies a minimum of 4 trainees for the scatter plot and 32 trainees for the radar chart. The optimum would then be 20 and 160 trainees, respectively.

Our practical experience shows that training events have around 20 participants per single training session, which satisfies the preferred amount for the scatter plots. On the other hand, the same game is often organized multiple times. The usage of radar charts is, therefore, restricted to bigger training events or to combine data from multiple sessions.

In another study regarding the sample sizes for clustering analysis (Dolnicar et al., 2014), the authors examine different data types. They do not bring a universal conclusion since specific knowledge of data structure is always necessary. To sum up their findings, the minimum requirement for sample size varies vastly across different data sets. However, they provide a recommendation of $70 \cdot d$ of data points for segmentation studies. Currently, we do not have enough data to fulfill this requirement, but gradually, as more data from the training definitions are collected, the analysis will become more accurate.

6 Conclusions

The presented research aims to gain insight into finished hands-on exercises by exploring automatically collected, processed, and visualized data. Without this automation topped off with well-elaborated interactive visual techniques, gaining insight is laborious, time-consuming, or even impossible.

6.1 Fulfillment of objectives

This paper shows how the clustering methods mediated by appropriate complementary visualizations can help tutors and training designers analyze typical behavioral data collected in cyber ranges. Proposed analytical goals *G1–G3* were formulated by studying the available data and consulting requirements with domain experts. The ability to examine analytical goals and infer hypotheses about the training content and the behavior of trainees was evaluated with the following results.

- *Examine typical gameplay strategies (G1):*. The participants were aimed at a more thorough recognition and assessment of the revealed clusters, and they identified similar types of trainee groups. The results show variability, but overall, the participants tended to pinpoint similar characteristics and groups of trainees. The average difficulty of the three related tasks, measured by the SEQ, ranged from 3.6 to 6.0 (from *somewhat difficult* to *easy*).
- *Identify flaws in training design (G2):* We covered the requirement in the scatter plot visualization with two initial views on the relationships between two sets of features. In two relevant tasks of the evaluation, the participants agreed on equivalent responses and defined a good training design by similar measures. On average, they rated the tasks as *somewhat easy*.
- *Identify outliers (G3):* The need to identify exceptional behavior among the trainees was supported by both visualization views. The evaluation was covered by two tasks, one for each visualization type. The radar chart proved more convenient for finding outliers with many distinct characteristics, while the scatter plot pinpoints only a small subset of specific behavior. The SEQ results rate the tasks as *somewhat easy* and *easy*.

6.2 Educational implications

The knowledge built gradually using the visual analysis tool can have significant implications for tutors who organize and supervise training sessions, training designers who prepare the content, and other stakeholders involved in cybersecurity education. In particular, the following types of learning insight can be gained:

- *Interest:* Using hints during the training can suggest the level of interest of trainees in solving tasks by themselves or, on the contrary, reveal the loss of motivation to make any effort (they take hints effortlessly instead). The goal of any content designer or training tutor should be to maintain high interest during the whole training because only then the impact on trainees can be expected.
- *Prerequisite knowledge:* Every exercise is designed for a certain level of assumed knowledge or skills. The training should be neither too easy nor too difficult. The proposed clustering-based analysis can show how much the assumptions correspond to reality or how much the input knowledge (and then the difficulty) was variable for a given study group. Observing these aspects allows tutors and content designers to take action, e.g., by introducing a theoretical lecture before the next hands-on training.
- *Technical or conceptual flaws:* Any technical (e.g., a long response time, authentication problems, or non-functionality of copy-and-past) or conceptual (e.g., typos in the assignment, confusing tasks or hints) problems can cause frustration followed by loss of motivation. Although the clustering-based visualizations cannot show the appearance of such issues directly, they can help to spot these situations

from the changed behavior of trainees, allowing tutors and designers to actively search for the flaws.

- *Training parameters:* Training difficulty can be managed by fine-tuning training parameters, especially allocated time and weakening or hardening penalties. Post-training review of training parameters can be performed by studying behavioral clusters.

6.3 Limitations

The number of participants involved in the evaluation was limited due to their need for experience designing and organizing practical cybersecurity exercises. However, this paper aims to provide a methodology for the application of clustering methods into process-oriented cybersecurity training sessions so that the behavior of trainees hidden in the raw event logs can be efficiently analyzed. In this sense, this paper should sufficiently demonstrate the analytical principles, and the evaluation brings promising results for its usability.

Another limitation relates to the performance of the initial clustering algorithm. It is important to note that in a typical setting of hands-on cybersecurity training, data of tens of users at most are processed at once. In such scenarios, the tool's performance suffices. However, since we evaluated its usefulness and usability with positive results, we plan to improve the tool's scalability for larger datasets in the following iterations to enable analysis over more extensive data.

The techniques of cluster analysis in education are still an emerging discipline. Our work aims at a small portion of the whole area, and there is still a need to find ways to provide new insight into the support of the training organizers and designers.

6.4 Future work

For future improvements, there are techniques that could further enhance the clustering output. Specifically, initializing the clusters using heuristics is an interesting factor in reducing the produced errors. In Fränti and Sieranoja (2019), the authors suggest using simple heuristics, which can improve the error from 15% to 6% on average. This difference can lead to better and more consistent results in clusters with several features and varied data. Optionally, we also plan to explore the viability of other clustering algorithms, such as density-based, which could provide interesting results. The visualization integrated into the training platform will also provide an option to show actual data values instead of the normalized variant since the evaluation reveals that they present more specific outcomes for further analysis.

As for the interaction with the tool, we plan to provide more customization and interaction for the visualizations. For example, by introducing an extended set of available features for the charts. Other possible features include the number of shell commands used to solve a task, inactivity indicators, the speed of submitting wrong flags, or taking a hint at the beginning of the level.

Our solution proved usability for the well-structured cybersecurity Capture the Flag games. Extending the same principles and exploratory visualizations to other types of hands-on exercises is an open problem and a subject of our future research.

Author Contributions Karolína Dočkalová Burská: Conceptualization, Methodology, Validation, Investigation, Resources, Data curation, Writing – original draft. Rudolf Mlynárik: Software, Validation. Radek Ošlejšek: Conceptualization, Investigation, Writing – review and editing.

Funding Open access publishing supported by the National Technical Library in Prague. This research was supported by the ERDF project CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence (No. CZ.02.1.01/0.0/0.0/16 019/0000822).

Data availibility The accompanying data and evaluation results are available on https://eait.surge.sh/. The links to the source code will be included in the final, non-anonymized version of the manuscript.

Declarations

Competing interests We have no competing interests to declare.

References

(ISC)^2. (2022). Cybersecurity workforce study. *Technical report*, $(ISC)^2$, https://www.isc2.org/Research/Workforce-Study.

Arthur, D., & Vassilvitskii, S. (2006). *k-means++: The advantages of careful seeding*. Tech. rep.

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *J Usability Studies, 4*(3), 114–123.

Chambers, J. M., Cleveland, W. S., Kleiner, B., et al. (2018). *Graphical methods for data analysis*. Chapman and Hall/CRC.

Chouliaras, N., Kittes, G., Kantzavelou, I., et al. (2021). Cyber ranges and testbeds for education, training, and research. *Applied Sciences, 11*(4).

Dasgupta, D., Ferebee, D.M., Michalewicz, Z. (2013). Applying puzzle-based learning to cyber-security education. In: *Proceedings of the 2013 on InfoSecCD'13: Information security curriculum development conference* (pp. 20–26).

Davis, A., Leek, T., Zhivich, M., et al. (2014). The fun and future of CTF. In: *2014 USENIX summit on gaming, games, and gamification in security education* (3GSE 14).

DeFreitas, K., & Bernard, M. (2015). Comparative performance analysis of clustering techniques in educational data mining. *IADIS International Journal on Computer Science & Information Systems, 10*(2), 65–78.

Dolnicar, S., Grün, B., Leisch, F., et al. (2014). Required sample sizes for data-driven market segmentation analyses in tourism. *Journal of Travel Research, 53*(3), 296–306.

Durairaj, M., & Vijitha, C. (2014). Educational data mining for prediction of student performance using clustering algorithms. *International Journal of Computer Science and Information Technologies, 5*(4), 5987–5991.

Dutt, A., Ismail, M. A., & Herawan, T. (2017). A systematic review on educational data mining. *IEEE Access, 5*, 15,991-16,005.

Eagle, C. (2013). Computer security competitions: Expanding educational outcomes. *IEEE Security & Privacy, 11*(4), 69–71.

Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., et al. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence, 110*(104), 743.

Formann, A.K. (1984). *Die latent-class-analyse: Einführung in Theorie und Anwendung*. Beltz

Fraley, C., & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal, 41*(8), 578–588.

Fränti, P., & Sieranoja, S. (2019). How much can k-means be improved by using better initialization and repeats? *Pattern Recognition, 93*, 95–112.

Gelbard, R., Goldman, O., & Spiegler, I. (2007). Investigating diversity of clustering methods: An empirical comparison. *Data & Knowledge Engineering,63*(1), 155–166, data Warehouse and Knowledge Discovery (DAWAK '05). https://doi.org/10.1016/j.datak.2007.01.002https://www.sciencedirect.com/science/article/pii/S0169023X07000031.

Hooshyar, D., Yang, Y., Pedaste, M., et al. (2020). Clustering algorithms in an educational context: An automatic comparative approach. *IEEE Access, 8*, 146,994-147,014.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters, 31*(8), 651–666.

Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons.

Khalil, M., & Ebner, M. (2017). Clustering patterns of engagement in massive open online courses (moocs): The use of learning analytics to reveal student categories. *Journal of Computing in Higher Education, 29*, 114–132.

Knüpfer, M., Bierwirth, T., Stiemert, L., et al. (2020). Cyber taxi: A taxonomy of interactive cyber training and education systems. In G. Hatzivasilis & S. Ioannidis (Eds.), *Model-driven simulation and training environments for cybersecurity* (pp. 3–21). Springer International Publishing.

Kucek, S., & Leitner, M. (2020). An empirical survey of functions and configurations of open-source capture the flag (ctf) environments. *Journal of Network and Computer Applications, 151*(102), 470.

Le Quy, T., Friege, G., Ntoutsi, E. (2023). A review of clustering models in educational data science toward fairness-aware learning. *Educational data science: Essentials, approaches, and tendencies: Proactive education based on empirical big data evidence* (pp. 43–94).

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory, 28*(2), 129–137.

Macák, M., Ošlejšek, R., Bühnová, B. (2022). Process mining analysis of puzzle-based cybersecurity training. In: *Proceedings of the 27th ACM conference on on innovation and technology in computer science education vol. 1 (ITiCSE '22)* (pp. 449–455). Association for Computing Machinery. https://doi.org/10.1145/3502718.3524819.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (pp. 281–297). Oakland.

Madhulatha, T. S. (2012). An overview on clustering methods. *IOSR Journal of Engineering, 2*(4), 719–725. https://doi.org/10.9790/3021-0204719725

McMurtrey, M. E., Downey, J. P., Zeltmann, S. M., et al. (2008). Critical skill sets of entry-level IT professionals: An empirical examination of perceptions from field personnel. *J of Inf Tech Education: Research, 7*, 101–120.

Meyer, M., Sedlmair, M., Munzner, T. (2012). The four-level nested model revisited: Blocks and guidelines. In: *Proceedings of the 2012 BELIV workshop: Beyond time and errors-novel evaluation methods for visualization* (pp. 1–6).

Michalewicz, Z., & Michalewicz, M. (2008). *Puzzle-based learning*. Ormond, Australia: Hybrid Publishers.

Mlynárik, J.R. (2022). Cluster analysis and visualization of cybersecurity games progress. Diploma thesis, Masaryk University, Faculty of Informatics, Brno, https://is.muni.cz/th/aeszy/, supervisor: Karolína Dočkalová Burská

Munzner, T. (2009). A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics, 15*(6), 921–928. https://doi.org/10.1109/TVCG.2009.111

Nainggolan R, Perangin-angin R, Simarmata E, et al. (2019) Improved the performance of the k-means cluster using the sum of squared error (sse) optimized by using the elbow method. In: *Journal of Physics: Conference Series* (p. 012015). IOP Publishing.

Ošlejšek, R., Rusňák, V., Burská, K., et al. (2021). Conceptual model of visual analytics for hands-on cybersecurity training. *IEEE Transactions on Visualization and Computer Graphics, 27*, 1–13. https://doi.org/10.1109/TVCG.2020.2977336

Rai, P., & Singh, S. (2010). A survey of clustering techniques. *International Journal of Computer Applications, 7*(12), 1–5.

Rodriguez, M. Z., Comin, C. H., Casanova, D., et al. (2019). Clustering algorithms: A comparative approach. *PloS one, 14*(1), e0210,236.

Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (applications and reviews), 40*(6), 601–618.

Sacha, D., Stoffel, A., Stoffel, F., et al. (2014). Knowledge generation model for visual analytics. *IEEE Transactions on Visualization and Computer Graphics, 20*(12), 1604–1613. https://doi.org/10.1109/TVCG.2014.2346481

Salloum, S.A., Alshurideh, M., Elnagar, A., et al. (2020). Mining in educational data: Review and future directions. In: *Proceedings of the international conference on Artificial Intelligence and Computer Vision (AICV2020)* (pp. 92–102). Springer.

Sauro, J. (2011). *A practical guide to the system usability scale: Background*. Benchmarks & Best Practices: CreateSpace Independent Publishing Platform.

Sauro, J., Dumas, J.S. (2009). Comparison of three one-question, post-task usability questionnaires. In: *Proceeding of the SIGCHI conference on human factors in computing systems* (pp. 1599–1608). ACM.

Švábenský, V., Vykopal, J., Cermak, M., et al. (2018). Enhancing cybersecurity skills by creating serious games. In: *Proceedings of the 23rd annual ACM conference on innovation and technology in computer science education* (pp. 194–199).

Švábenský, V., Weiss, R., Cook, J., et al. (2022). Evaluating two approaches to assessing student progress in cybersecurity exercises. In: *Proceedings of the 53rd ACM technical symposium on computer science education (SIGCSE '22)* (To appear). ACM.

Švábenský, V., Vykopal, J., Čeleda, P., et al. (2022). Student assessment in cybersecurity training automated by pattern mining and clustering. *Education and Information Technologies, 27*(7), 9231–9262.

Ukwandu, E., Farah, M. A. B., Hindy, H., et al. (2020). A review of cyber-ranges and test-beds: Current and future trends. *Sensors, 20*(24).

van der Aalst, W. (2016). *Process Mining: Data Science in Action* (2nd ed.). Incorporated: Springer Publishing Company.

Vykopal, J., Ošlejšek, R., Čeleda, P., et al. (2017). Kypo cyber range: Design and use cases. In: *Proceedings of the 12th international conference on software technologies - volume 1: ICSOFT* (pp. 310–321). SciTePress. https://doi.org/10.5220/0006428203100321

Wang, H. (2021). Research of spoc online learning behavior analysis based on rft. In: *2021 IEEE international conference on progress in informatics and computing (PIC), IEEE* (pp. 515–518).

Weiss, R., Locasto, M.E., Mache, J. (2016). A reflective approach to assessing student performance in cybersecurity exercises. In: *Proceedings of the 47th ACM technical symposium on computing science education* (pp. 597–602).

Werther, J., Zhivich, M., Leek, T., et al. (2011). Experiences in cyber security education: The MIT lincoln laboratory capture-the-flag exercise. In: *4th workshop on Cyber Security Experimentation and Test (CSET 11)*

Wu, X., Kumar, V., Ross Quinlan, J., et al. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*, 1–37.

Yamin, M. M., Katt, B., & Gkioulos, V. (2020). Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. *Computers & Security, 88*(101), 636. https://doi.org/10.1016/j.cose.2019.101636

Yin, H., Moghadam, J., Fox, A. (2015). Clustering student programming assignments to multiply instructor leverage. In: *Proceedings of the second (2015) ACM conference on learning@ scale* (pp. 367–372).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.