Exploring Trainees' Behaviour in Hands-on Cybersecurity Exercises Through Data Mining

Muaan ur Rehman[1], Hayretdin Bahsi[1,2], Linas Bukauskas[3] and Benjamin James Knox[4]
[1]Department of Software Sciences, Tallinn University of Technology, Estonia
[2]School of Informatics, Computing and Cyber Systems, Northern Arizona University, USA
[3]Institute of Computer Science, Vilnius University, Lithuania
[4]Faculty of Health, Welfare and Organization, Østfold University College, Halden, Norway

muaaur@taltech.ee
hayretdin.bahsi@taltech.ee
linas.bukauskas@mif.vu.lt
benjamin.knox@hiof.no

Abstract: Despite the rising number of cybersecurity professionals, the demand for more experts in this field is still substantial. Cybersecurity professionals must also possess up-to-date knowledge and skills to counter cybersecurity threats' dynamicity and rapidly evolving nature. Hands-on cybersecurity training is mandatory to practice various tools and improve one's technical cybersecurity skills. Generally, an interactive learning environment is set, where trainees perform sophisticated tasks by accessing complete operating systems, applications, and networks. One of the main challenges that cybersecurity organizations are facing today is the generation of massive data through practice exercises. So, it becomes a problem to convert this data into knowledge to improve the overall quality of the learning system. The large amount of interaction data and its complexity also limit us to do automated analysis. Thus, these challenges for cybersecurity learners can be addressed through appropriate educational data analysis by having insights or testing hypotheses or models on a proper dataset. Revealing the patterns, rules, item sets and time taken by trainees while using any command line tool could help the trainer to assess the trainees and to provide feedback. Therefore, in this paper we are analyzing the frequency patterns and timing information captured from the trainees' command line log to reveal their solving techniques, easy and struggling stages, slipups, and individual performance. Through our study, we aim to show how education and training providers can foresee learners who are less likely to succeed in a task or exhibit low performance, which can impede learning proficiency. With this knowledge, organizations and trainers can identify trainees who require additional attention or support. It may also be able to identify elements related to an organization like training aids, training methodology, etc. that need improvement. This study demonstrates the utility of data-mining techniques, specifically rule mining and sequential mining, to empower training designers to delve into datasets derived from cyber security training exercises.

Keywords: Cybersecurity Education, Educational Data-Mining, Learning Analytics, Cybersecurity Training

1. Introduction

Data-mining techniques are being used in the learning environment to predict trainees' learning behavior and improve their performance. The main goal of the institutions offering cyber security courses is to allow a quality education among individuals which in turn increases the learner's performance and assists in better decision-making. Trainees' performance is influenced by a variety of factors, which in turn affect the quality of training at different levels. Hands-on cybersecurity training exercises enable learners to practice their skills in a controlled environment. Hands-on exercises generate massive amounts of data which contain useful patterns which may reflect learners learning behaviors, their participation in different lab exercises, their interest in different attacking techniques and overall performance. Due to the enormous number of logs generated through hands-on exercises, identification of factors which affect performance is challenging. Thus, an appropriate educational data analysis on a proper dataset is required to address these challenges. In this paper, we aim to show how data-mining techniques, such as rule mining and sequential mining, can enable cybersecurity training providers to foresee at-risk learners and identify trainees who require additional attention or support.

We have investigated the dataset (Švábenský, Vykopal, et al., 2021) which consists of command logs recorded from trainees during security training sessions. The dataset has a high potential for accurate analysis of trainees as it contains enough attack log records acquired from different command line tools. Accessing and analysing this data set has significant potential in the area of learner/trainee assessment. To design future cybersecurity training further research is required in the exploration of past training data. There is a clear need to identify and explore additional factors, that may go unnoticed, from an increasing amount of cybersecurity exercise data. This research makes a notable contribution to exploring issues regarding the assessment of cybersecurity trainees through data-mining techniques to assist trainers. This study can also be useful to training designers in ongoing training exercises and providing some support in designing new effective training.

In Section 1 the work is introduced and the main contributions are formulated. Section 2 provides a review of the related studies. Section 3 explains the methodology used for data collection and analysis. In Section 4 the results and findings are discussed. Finally, Section 5 summarizes the contribution and proposes some future work.

2. Related Work

In the realm of educational data-mining-based assessment of cybersecurity learners, Švábenský et al. (2022) performed student assessments in cybersecurity training via pattern mining and clustering. Seda et al. (2022) also presented an instructor guide and a tool to improve the creation of cybersecurity hands-on training with adaptive learning support, which uses students' performance and abilities to assign suitable tasks to them. (Švábenský, Weiss, et al., 2022) performed cyber security student assessment via visualizing and contextualization of the command history logs. Li et al. (2021) utilized the Apriori algorithm for the data-mining of global cyberspace security issues involving human participation (Li et al., 2021). These studies offer diverse methodologies for assessing cybersecurity learners and thus present opportunities to build upon these techniques in further research.

Hands-on cybersecurity training exercises enable learners to practice their skills in a controlled environment. Cyber ranges, the interactive simulated platforms are being used for developing cyber skills or testing products (Jani Päijänen et al., 2021). One of the main challenges that cybersecurity organizations are facing today is the generation of massive amounts of data through practice exercises. The issue lies in converting this data into knowledge to improve the overall quality of the learning system (Asif et al., 2017; Bhutto et al., 2020). Having this huge amount of data in educational environments hides useful patterns which may reflect learning behaviours, participation in different lab exercises, interest in different attacking techniques and overall performance. Furthermore, due to the enormous number of logs generated through hands-on exercises, identification of factors which affect performance is challenging. Thus, these challenges for cybersecurity learners can be addressed through appropriate educational data analysis by having insights or testing hypotheses or models on a proper database.

3. Methodology

The present study aims to overcome the problem of converting massive amounts of cybersecurity exercise log data into knowledge and educational insight. This is achieved by investigating temporal data-mining techniques to assess trainees and identify the factors that influence their performance. The study proposes a data-mining-based framework to predict trainee performance, which can be used to classify trainees, based on their exercise solving procedure. The approach is to identify and examine the possible factors that can be correlated to learner performance or can be explained as causing the performance. Fig. 1 shows the steps of the proposed framework. The following sections further explain each step of our methodology.

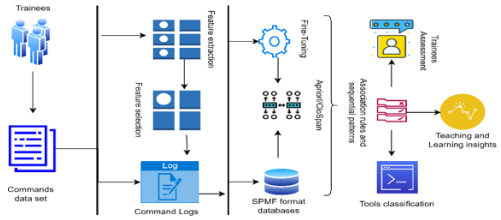

Fig 1. Methodology employed in this research, wherein command logs from the dataset are utilized to generate rules and patterns, yielding valuable insights.

3.1 Dataset

The first step is to gather data from trainees completing cybersecurity tasks during lab exercises. This was achieved by taking advantage of available datasets. In this case, a dataset (Švábenský, Vykopal, et al., 2021) of shell commands was used by participants who solved tasks in an interactive learning environment. This dataset contains 13446 Linux shell commands which were recorded from 175 participants. The Cyber Range platform KYPO (Vykopal et al., 2021) and Cyber Sandbox Creator were used to emulate complex networks and host a mini virtual lab environment on the trainee's machine. The data record contains a command, its argument and some metadata i.e., sandbox identification, timestamp, host identification and working directory in the emulated training infrastructure as shown in Fig 2. These commands were acquired by Bash, ZSH and Metasploit shells.

{"timestamp_str":"2021-02-11T15:21: 34+01:00","sandbox_id":"12",
"pool_id":"1","cmd":"ls", "username": "root", "wd":"/root",
"hostname":"attacker","ip":"10.1.26.23","cmd_type":"bash-command"}

Fig 2. Log Record

3.2 Game-based Hands-on Exercises:

The research focused on the analysis of cybersecurity practical exercise data taken from an exercise that was especially focused on offensive security training. For this purpose, we have targeted the mentioned dataset (Švábenský, Vykopal, et al., 2021) which is publicly available. It includes the following games[1] (Švábenský, Vykopal, et al., 2018).

Junior-hacker/Junior-hacker Adaptive: In these types of games, the host just connects to the sandbox, reads info, scans with *nmap*, manually guesses the password with SSH, transfers files with *scp*, using *fcrackzip* for password cracking and then changes the password. Participants in these types of games are learning experimentally via simple *Bash-Commands*.

SQL-Injection: *Sqlmap* and *ssh* are used for SQL injection; however, comparatively fewer commands and participants due to the usage of graphical tools only and no *Metasploit* usage.
Locust-3302: Vulnerabilities are searched via *Metasploit* commands. Among the key *Metasploit* commands are *use, set, run/exploit, search, show options* and *check* for executing the exploits in this type of game. Host scanning, password cracking, SSH connection and Webmin exploitation are performed.

Secret-Lab/Webmin-Exploit: These games are variations of Locust-3302. *nmap* is used for scanning, *webmin* is exploited with *Metasploit*. The host explores and analyses the .bash history. Furthermore, password cracking is performed via John the Ripper.

House-of-Cards: Both *Metasploit* and *Bash-commands* are being used for the exploitation of CVE-2018-10933 (Common Vulnerabilities and Exposures), a critical security issue in the LibSSH Library. Tools such as *nmap* and *ssh* are common tools in it.

3.3 Dataset Pre-Processing

Proper pre-processing is required to make the raw dataset ready for rules mining and pattern mining. Due to our temporal approach, *timestamp_str*, which represents the exact date and time of a command typed, is very crucial. To evaluate the performance of a particular participant, we consider *sandbox_id* as a potential attribute of interaction from a unique trainee. We have considered the feature *cmd* as a whole command typed, then also separated it from the passed arguments and labelled it as a *tool*. We have ignored the *cmd_type*, since it is already known which type of command is used in a particular exercise and hence redundant rules and patterns are avoided. Furthermore, since the dataset consists of different games some are easy and others are harder. We have extracted the json files from different game-based log folders and classified them into 3 types of major games. i.e. Junior-hackers' level, Intermediate (Locust, Secret-Lab and Webmin-Exploit) and challenging one (House-of-Card). We also have made a separate input folder consisting of all the participant's interactions together. To mine the association rules and patterns, we have converted the logs into BMS IBM Data Quest Generator format to feed the data as input to mining algorithms in the shape of input SPMF databases.

3.4 Time Lapses:

To know the reliable performance measures, time taken by a trainee to type and execute a command is highly significant. The time distance between two commands typed by trainee shows, either the efficiency of a trainee or the difficulty of a specific commands to be executed. However, the dataset (Švábenský, Vykopal, et al., 2021) does not directly record the time taken by a specific command to be executed. We have assessed the efficiency of a particular participant by measuring the time elapsed between the current execution time (*timestamp_str)* and the *timestamp_str* of the subsequent command execution. To find association rules and patterns, it is best to assign labels for the time taken until next recoded command. For this purpose, we have labelled the time

[1] https://gitlab.ics.muni.cz/muni-kypo-trainings/games/all-games-index

spent between adjacent commands from the logs as low, medium, high, and undefined. It is particularly important to investigate the way we can label some commands as low and other as high.

Let C be list of commands i.e.

$$C = \{ c_1, c_2, \ldots, c_n \} \quad (1)$$

The function $f_{gap}(C)$ return a list of the time differences between consecutive commands as shown below in (2).

$$f_{gap}(C) = \{\Delta t_1, \Delta t_1, \ldots, \Delta t_{n-1}\} \quad (2)$$

For instance, a command c_i is typed at *timestamp_str*: t_i and the next immediate command c_{i+1} at *timestamp_str* t_{i+1}. The formula to compute time difference (Δt_i) is shown in equation (3).

$$\Delta t_i = \sec(t_{i+1} - t_i), \quad \text{for } 1 \leq i \leq n\text{-}1 \quad (3)$$

We find the average gap time G_{avg}, which is the average time taken by all the participants in a specific game exercise. This is calculated based on mean of all the gaps in a particular game. G_{avg} combined with corresponding constants k_{min}, k_{max} and k_{undf} leads us to calculate the lower-gap, high-gap and undefined-gap represented by G_{low}, G_{high} and G_{undf} respectively. To calculate the minimum gap threshold G_{low} we divide the average gap (G_{avg}) by different number k_{min} which gives us a number less than the average and hence could be the minimum threshold value as shown in (4). For finding the higher threshold (G_{high}), we multiply the average (G_{avg}) by some small number k_{max} which gives number bigger than average as shown in (5). To find the undefined outlier, we have formulated equation (6) which gives us number far greater than highest threshold (G_{high}).

$$G_{low} = \frac{1}{k_{min}} G_{avg} \quad (4)$$

$$G_{high} = k_{max}\, G_{avg} \quad (5)$$

$$G_{undf} = k_{undf} G_{avg} \quad (6)$$

The values of the corresponding constant k_{min}, k_{max} and k_{undf} are identified by considering different factors. Such as domain knowledge, consideration of outliers and empirical analysis. We did the analysis and performed a series of experiments with varying constant values considering the standard deviations. Through this iterative process, we converged on optimal values of $k_{\min} = 3$, $k_{\max} = 1.25$ and $k_{\text{undf}} = 90$, aligning with the characteristics and general analysis of the dataset (Švábenský, Vykopal, et al., 2021) logs. Setting different values to these constants gives us variable results. Therefore, setting the time threshold is an important step. The values of these constants must be the same across the whole analysis process. Substantially deviating k_{min} from the G_{avg} value is observed to have a noteworthy impact on the potential to achieve time interval-based results. So, it is recommended to maintain k_{min} in close proximity to the value of G_{avg}.

To classify the time intervals Δt_i associated with command c_i, denoted as $f_{gap}(c_i)$, we have formulated distinct gap ranges as follows:

- $0 \leq \Delta t_i \leq G_{low}$: Low
- $G_{low} < \Delta t_i \leq G_{high}$: Medium
- $G_{high} < \Delta t_i \leq G_{undf}$: High
- $\Delta t_i > G_{high}$: Undefined

3.5 Data Analytics:

SPMF (Fournier-Viger, 2021) open-source tool to mine rules and patterns was applied due to its ability to be used as a library and could be embedded into our python code. To find the item co-occurrences we mined simple association rules. We have used Apriori (Li et al., 2021) for association rules mining. To discover closed command sequences in the dataset, we have mined sequential patterns by utilizing CloSpan (Yan et al., 2003). This algorithm effectively extracts comprehensive insights from sequential and large amount of data. *TopSeqRule*s (Fournier-Viger and Tseng, 2011) is used for Sequential rules mining to further uncover ordered tool sequences.

The minimum support and confidence thresholds varied across different mining processes. In our experiments, we set the minimum support above 0.5 for association rule mining and 0.3 for pattern mining. Additionally, confidence has been maintained at a level above 0.4. These parameters are grounded in careful consideration of the dataset (Švábenský, Vykopal, et al., 2021) features and the desire to discover useful rules and patterns from it. In the next section, we are going to discuss the results we achieved through this methodology.

4. Results and Discussion

In our study, we conducted a comprehensive analysis of the hands-on cybersecurity dataset (Švábenský, Vykopal, et al., 2021) using various data-mining approaches. Specifically, we mined simple, temporal and sequential association rules. We also identified close sequential patterns to deepen our understanding of the dataset.

Table 1: Association rules (Multiple games)

No.	Rules	Support	Confidence	Game	Type
1.1	TOOL=fcrackzip==>GAP=high	47	0.58	Junior Hacker	Basic
1.2	TOOL=ifconfig==>GAP=Medium	11	0.61		
1.3	TOOL=sudo==>GAP=medium	14	0.51		
1.4	TOOL=ssh==>ARGS=['admin@10.1.26.9']	50	0.58		
1.5	SB=12 ==>GAP=low	95	0.61		
1.6	SB=2 ==>GAP=high	16	0.41		
1.7	SB=18==>GAP=undefined	1	1		
2.1	TOOL=man==>ARGS=['fcrackzip']	93	0.69	Junior Hacker Adaptive	
3.1	GAP=high==>TOOL=sqlmap	6	0.54	SQL Injection	
3.2	TOOL=cd, SB=116==>GAP=low	9	0.63		
3.3	TOOL=ssh==>GAP=undefined	9	0.9		
3.4	SB=122==>GAP=undefined	3	0.6		
4.1	ARGS=['options']==>TOOL=show	28	0.97	Locust	Medium
4.2	TOOL=nmap ==>GAP=high	21	0.71		
4.3	TOOL=clear==>SB=126, GAP=low	7	0.57	Secret Lab	
5.1	SB=351==>GAP=low	67	0.53	Web-Min-Exploit	
5.2	TOOL=man ==>SB=351	7	1		
6.1	TOOL=nmap==>GAP=high	50	0.50	House of Cards	Advance
6.2	ARGS=['rhosts','172.18.1.5'], TOOL=set ==>GAP=low	37	0.88		
6.3	SB=131==>GAP=high	11	0.5		
6.4	SB=401==>GAP=low	58	0.63		
6.5	SB=93==>GAP=medium	26	0.48		
7.1	TOOL=nmap==>GAP=high	33	0.72	All Games	NA

SB: Sandbox ID, represents a single participant; Support range: 1-95; Confidence: above 0.4

Table 1 shows some of the interesting association rules mined. Among the interesting command-based rules for Junior-hacker we found that: *Fcrackzip* is used with a very high time gap, and tool *ifconfig* is used with a medium gap. *Ifconfig* and *sudo* tools took average time gaps. The rules from Junior-hacker also give us some of the instances of the participant's efficiencies. Rule 1.5 illustrates that Participant *12* has 95 commands (support=95) typed very quickly (low-gap), which shows a high level of engagement and possibly interest and dedication. In contrast, SB 2 has a higher time gap (Rule 1.6), indicating a different pattern of behaviour. Rule 1.7 suggests that SB *18* could have had some problems in the machine, could have lost interest, or was possibly making errors due to extremely high gaps. Rule 2.1 depicts that if the command arguments are ['fcrackzip'], then the corresponding tool is *man* in the case of Junior-hacker Adaptive: We assume then, that man is used to display manuals for a command. It seems these participants needed more reading time. Such a command does not exist in Junior-hacker and Junior adaptive. It is possible these participants required more training time about *fcrackzip*.

As expected, *Sqlmap* looks to be the primary tool for *Sql-injection* as suggested by rule 3.1. Another rule here is change of the directories more frequently and quickly, especially by trainee *116*. In general trainee *116* was very fast having support of 19 being typing with lower-gaps and using the *cd* command more frequently and faster than others and suggest his navigation skills (Rule 3.2). Furthermore, the *ssh* command, in the majority of cases, took an abnormal amount of time. Reason for such abnormal time should be investigated. We also see that user *122* had more undefined-gaps and therefore, he/she could possibly require more trainer attention. In *Locust*, we see is that command *show* with argument option is used frequently. This could imply that the participant needed help. An interesting result in *Secret-Lab* is that participant *SB=126* used *clear* command more than usual and also with very high speed. This could suggest the participant is performing many mistakes while typing commands, or is more concerned about deleting the logs history. At *WebMin-Exploit* level user *351* is too quick in typing commands compared to other trainees (rule 5.1). However, the same participant focused a great deal on *man* commands. That the participant requested manuals could suggest a need for help, although his/her overall execution of commands is fast. For *House-of-Card* game in Table 1 rule 6.1 shows that the tool *nmap* has taken the highest time. Therefore, either the command looks difficult, or the participants have less knowledge of using this command. This may indicate that *nmap* itself takes too much time. Furthermore, the trainees in House-of-Card game quickly and effectively executed the Metasploit set command, especially when the target system IP 172.18.1.5', which employ the vulnerabilities and has easy access to that particular IP. Rules 6.3-6.5 in Table 1 also depicts participant's performance in terms of time taken to execute commands. We can see that in the case of *SB=401*, the participant has taken the lowest time while executing commands. This can be shown via the highest support of 58 as a low-gap interval. Users with high support and higher confidence, for low time intervals, such as *SB=401*, may indicate their capability and engagement. In the highest time intervals, *SB=131* has the highest support among high-gap participants. This suggests that for the same type of game/exercise, this trainee may need to benefit from additional support based upon the delay in typing and executing commands. The medium-gaps, participant could be considered as improving trainees.

In order to identify additional factors, Sequential Rules Mining, is performed. It is a useful data mining technique to find sequential rules from large database. These rules indicate that participants practice certain command combinations with high certainty. Unlike simple association rules in Table 1, sequential rules presented in Table 2 provide some patterns about the whole command sequence. Furthermore, alternative to simple association rules, it also considers the probability of succeeding patterns (Abdelwahab & Youssef, 2022). Moreover, they provide insights into the typical actions' trainees take in various games and can be valuable for guiding the trainees, optimizing hands-on exercise content, and ensuring better performance in exercises.

Table 2: Sequential Rules (Multiple Games)

No	Sequential Rules	S	C	Type
1.1	nmap ==>set	111	0.85	Locust
2.1	sqlmap ==>ssh	4	1	SQL Injection
3.1	nmap==>msfconsole, use	10	1	Webmin-Exploit

S: Support, support threshold: 0.5; C: Confidence, threshold: 0.4

Rule 1.1 in Table 2 illustrates that in the case of Locust participants performing a "nmap" scan, they have a 85.38% chance of following up with a set action. It is expected that the user use nmap for exploration and then use set for configuration of the Metasploit for exploitation stage. The high confidence suggests a strong sequential pattern where *nmap* often leads to setting (*set*). If we observe Sql-injection exercise, rule 2.1 depicts that the tool sqlmap is immediately followed by the ssh. This indicates a common sequence where trainees use Sql-injection to gain access and then proceed to execute SSH. Webmin-Exploit's sequential rule 3.1 suggests a strategic approach to penetration testing. Starting with network scanning *nmap*, they then move to the Metasploit console *msfconsole* and use it to execute exploits by the tool *use*. This pattern suggests participants' recognition of the value of reconnaissance in identifying potential targets and their subsequent use of Metasploit for targeted exploitation. It exemplifies the significance of assembling intelligence before launching attacks.

For further exploration of the hands-on cybersecurity dataset we have mined the closed sequential patterns, which are maximally specific sequential pattern, indicating significant and non-redundant sequences.

Table 3: Close Sequential Patterns (House of card)

No	Patterns	Support	Trainee/SB IDs
1	nmap→msfconsole→set	18	112,113,115,116,130,137,144,148,149,152,156,396,398,399,400,401,93,95
2	ssh→nmap→ssh	18	101,112,113,115,116,137,148,149,151,152,156,396,398,399,400,93,397,95
3	ssh→nmap→set→set→run	18	101,112,113,115,116,137,148,149,151,152,156,396,398,399,400,93,397,95
4	nmap→set→set	22	101,112,113,115,116,130,131,137,144,148,149,151,152,156,396,398,399,400,401,93,397,95
5	ssh→ssh→ssh	21	101,112,113,115,116,130,131,137,148,149,151,152,156,396,398,399,400,401,93,397,95
6	nmap→use→set→set	18	101,112,116,131,137,144,148,149,152,156,396,398,399,400,401,93,397,95
7	nmap→set→set→ssh→ssh	20	101,112,113,115,116,130,131,137,148,149,151,152,156,396,399,400,401,93,397,95
8	nmap → use → set → ssh	20	101,112,113,115,116,130,131,137,148,149,152,156,396,398,399,400,401,93,397,95
9	nmap→set→set→run	19	101,1,156,115,116,130,137,144,148,149,152,156,396,398,399,400,401,93,397,95
10	nmap→use→set→ssh→ssh	19	101,112,113,115,116,130,131,137,148,149,152,156,396,399,400,401,93,397,95
11	nmap→use→set→run→ssh→ ssh	18	101,112,113,115,116,130,137,148,149,152,156,396,399,400,401,93,397,95

Table 3 shows closed sequential patterns along with the support values and associated trainees (represented by Sandbox IDs) mined from the House-of-Cards game. Participants followed various combinations of tools to achieve a similar goal. Based on the logs from the mentioned game, support values equal to 22 are considered the highest level of support, while values less than 18 are categorized as low support. The discussion in the following paragraph centres around specific tools-based analysis on patterns mined from House-of-Card game (Table 3).

SSH-nmap combinations: There are several occurrences of the sequence “ssh→nmap→ssh" (pattern 2) with a high support (e.g., 18). This pattern could indicate a common workflow that involves using SSH to connect and then performing network scanning with *nmap*. Such a sequence may be used in the reconnaissance stage of the task as it does not include any Metasploit command.

Successful attack cycle (ssh→nmap→set→run): The sequences ssh→nmap→set→set→run (pattern 3) has a substantial support i.e. 18. It indicates participants who used *nmap* then configured Metasploit (set) and finally launched the attack (run) might represent a successful approach to achieving the goal.

Incomplete Sequences: Trainees who consistently follow sequences like "nmap→set→set" (pattern 4) have a high support (e.g., 22). As we discussed earlier ssh→nmap→set→run is the most common successful attack cycle. Therefore, trainees who only follow pattern 4 i.e., do not execute the run command at the end, indicates they did not finish the full cycle. Further investigation is required concerning these participants (130, 131, 144 and 401) who could not run the exploitation in the target. A possible reason might be that they were unable to locate the relevant configuration.

Variations in SSH Sequences: The sequence "ssh→ ssh→ssh" (pattern 5) has significant support (e.g., 21). This repeated use of SSH could reveal an investigative type behaviour, as certain tasks or scenarios require multiple SSH connections.

Table 3 also depicts that some trainees exhibit patterns that stand out from the common sequences and could indicate specific preferences, skills, or focus areas. Further analysis and discussions with these trainees could provide insights into their learning strategies, tool preferences, and potential areas for improvement in their training. Following are a number of findings worth debating: Unlike other participants, trainee *144* does not follow ssh→nmap→ssh (pattern 2), nmap→use→set →ssh→ssh (pattern 10) and nmap→use→set→run→ ssh→ssh (pattern 11). These pattern shows more emphasis on SSH tool and initiation of multiple remote connections. However, each trainee may have their own unique or preferred task, which needs further investigation. Trainee *148* is consistently present in all the sequences, which may suggest this particular trainee is actively engaged in using different tools.

4.1 Discussion

Our study shows that data mining techniques enable the training designers to explore the cyber security exercise datasets that contain some low-level information such as the running commands of each participant. The rules derived from such techniques can give valuable insights into the ongoing trainings. However, it is important to note that experts are needed to evaluate the identified rules and select the ones which may be helpful in the analysis. On the other side, the results of these rules have an explorative nature, necessitating more investigation and, thus, augmented dataset collection, and meticulous analysis to establish statistical values for variables, identify the root causes of deviations and draw solid conclusions. Our study demonstrates that the data mining techniques we applied in this paper are well-suited to this context, in which training designers do not know where to start due to the complexity of the dataset.

The dataset (Švábenský, Vykopal, et al., 2021) used in this study constitutes a significant source for similar data analytic studies related to cyber security trainings or games. However, the number of security tools and commands that the dataset obtained (i.e., even in the sophisticated ones) is limited. Although we garnered valuable insights from this dataset, we contemplate that data mining methods can reveal more insights when more varied tools/commands are utilized in cyber security training.

In our study, the associate rule mining method enabled us to acquire a general overview of the dataset and identify some significant insights about tool usage, including timing issues. Although we understand some key points about the participants in general, sequential rule mining focused on the analysis of the command series used by each participant during the whole training. As the number of tools and commands is limited in the dataset, we identified a typical pattern in which reconnaissance activities (i.e., represented by the Nmap tool) are followed by exploitation activities (i.e., various commands of Metasploit) in the training where complete attack scenarios are used. The patterns using some commands (e.g., *run*) may indicate that the participant can complete the whole exploitation cycle. Nevertheless, the data mining methods applied in this study can give more insights into the patterns of cyber kill chain steps in the datasets collected from more advanced training environments (e.g., cyber security exercises having more complex scenarios).

This research also implies some factors affecting the performance of participants. For example, how the time gap between commands (c_i) and (c_{i+1}) can be a factor in assessing the difficulty, or ease, of executing/choosing a particular command. Apart from this, the time gap best represents the performance of participants. If the same command takes too much time for one student while less for another student, this could quite possibly be a performance correlate, by inferring attention or knowledge level. If the command typed with high time intervals is due to overcautiousness, that factor must be identified and balanced with efficiency. Among other factors, analysis of the command's specific difficulty level can support how we conclude whether a particular participant has taken more time or less according to the nature of the command. By considering different other factors i.e. training, experience level, specific domain knowledge, and self-efficacy trainees may be assessed and potentially predictions can be made about their [future] performance more effectively.

5. Conclusions

This research endeavours to enhance the effectiveness of cybersecurity interactive exercises by employing a data mining-based approach through exploring rules, patterns and behaviours exhibited by participants.

In this paper, we presented a comprehensive approach aiming to address the challenge of converting wide-ranging cybersecurity exercise logs into valuable instructive insights. Through the data mining techniques, trainee performance factors were investigated. These factors have the potential to predict and classify trainees based on their task solving abilities and approaches. The research can provide cybersecurity educational decision makers with a systematic approach to use the command logs and intervene in learning performance by analysing the performance factors identified e.g. time intervals between commands executed, specific command's difficulty, experience level and domain knowledge of the learners etc.

Through the lens of temporal association rules, our study uncovers the variability inherent in trainees' interaction and response times. Sequential rules shed more light on the individualized approach adopted by participants in different gaming scenarios. Mining of closed sequential patterns unveils common successful and unsuccessful (incomplete) sequences. Identification of such distinctive patterns may suggest areas of expertise, or skills of certain trainees. This trainee-centric analysis investigated individual behaviours, pinpointing participants with unique patterns and preferences. The individual participant scrutiny presents an avenue for tailored guidance and improvement strategies by identifying focused workflows and versatile usage of tools.

This paper offers many opportunities for future work. The research can be enhanced by incorporating predictive models that consider cognitive factors. In achieving a target, the trainee's success and failure can be predicted using machine learning approaches. A benchmarking mechanism can be established by comparative analyses to assess the effectiveness of interventions in different training exercise programs.

Acknowledgements

The "Advancing Human Performance in Cybersecurity'', ADVANCES, benefits from €1 million grant from Iceland, Liechtenstein, and Norway through the EEA Grants. The project aims to advance the performance of cybersecurity specialists by personalising the competence development path and risk assessment. The project contract with the Research Council of Lithuania (LMTLT) No is S-BMT-21-6 (LT08-2-LMT-K-01-051).

References

Abdelwahab, A. and Youssef, N. (2022) Performance Evaluation of Sequential Rule Mining Algorithms. *Applied Sciences*, *12*(10), 5230. https://doi.org/10.3390/app12105230

Abu, A. (2016) Educational Data Mining and amp; Students' Performance Prediction. *International Journal of Advanced Computer Science and Applications*, *7*(5) https://doi.org/10.14569/IJACSA.2016.070531

Asif, R., Merceron, A., Ali, S. A. and Haider, N. G. (2017) Analyzing undergraduate students' performance using educational data-mining. *Computers and Education*, *113*, 177–194. https://doi.org/10.1016/j.compedu.2017.05.007

Bhardwaj, B. K. and Pal, S. (2012) *Data Mining: A prediction for performance improvement using classification*.

Bhutto, Engr. S., Siddiqui, I. F., Arain, Q. A. and Anwar, M. (2020) Predicting Students' Academic Performance Through Supervised Machine Learning. *2020 International Conference on Information Science and Communication Technology (ICISCT)*, 1–6. https://doi.org/10.1109/ICISCT49550.2020.9080033

Fournier-Viger. (2021) *SPMF: An open-source data mining library*.

Fournier-Viger, P. and Tseng, V. S. (2011) *Mining Top-K Sequential Rules* (pp. 180–194) https://doi.org/10.1007/978-3-642-25856-5_14

Jani Päijänen, Karo Saharinen and Jarno Salonen. (2021) Cyber Range: Preparing for Crisis or Something Just for Technical People? *Proceedings of the European Conference on Information Warfare and Security*. https://doi.org/10.34190/EWS.21.012

Li, Z., Li, X., Tang, R. and Zhang, L. (2021) Apriori Algorithm for the Data Mining of Global Cyberspace Security Issues for Human Participatory Based on Association Rules. *Frontiers in Psychology*, *11*. https://doi.org/10.3389/fpsyg.2020.582480

R Negi, S. K. S. M. S. (2021) Automated Flag Detection and Participant Peformance Evaluation for Pwnable CTF. *Silicon Valley Cybersecurity Conference*.

Seda, P., Vykopal, J., Celeda, P. and Ignac, I. (2022) Designing Adaptive Cybersecurity Hands-on Training. *2022 IEEE Frontiers in Education Conference (FIE)*, 1–8. https://doi.org/10.1109/FIE56618.2022.9962663

Švábenský, V., Vykopal, J., Čeleda, P., Tkáčik, K. and Popovič, D. (2022) Student assessment in cybersecurity training automated by pattern mining and clustering. *Education and Information Technologies*, *27*(7), 9231–9262. https://doi.org/10.1007/s10639-022-10954-4

Švábenský, V., Vykopal, J., Seda, P. and Čeleda, P. (2021) Dataset of shell commands used by participants of hands-on cybersecurity training. *Data in Brief*, *38*, 107398. https://doi.org/10.1016/j.dib.2021.107398

Švábenský, V., Weiss, R., Cook, J., Vykopal, J., Čeleda, P., Mache, J., Chudovský, R. and Chattopadhyay, A. (2022) Evaluating Two Approaches to Assessing Student Progress in Cybersecurity Exercises. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, 787–793. https://doi.org/10.1145/3478431.3499414

Vykopal, J., Celeda, P., Seda, P., Svabensky, V. and Tovarnak, D. (2021) Scalable Learning Environments for Teaching Cybersecurity Hands-on. *2021 IEEE Frontiers in Education Conference (FIE)*, 1–9. https://doi.org/10.1109/FIE49875.2021.9637180

Yan, X., Han, J. and Afshar, R. (2003) CloSpan: Mining: Closed Sequential Patterns in Large Datasets. *Proceedings of the 2003 SIAM International Conference on Data Mining*, 166–177. https://doi.org/10.1137/1.9781611972733.15

Valdemar Švábenský, Jan Vykopal, Milan Cermak, and Martin Laštovička. (2018) 'Enhancing cybersecurity skills by creating serious games' in Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018). Association for Computing Machinery, New York, NY, USA, 194–199. https://doi.org/10.1145/3197091.3197123