








# A Fast Methodology to Find Decisively Strong Association Rules (DSR) by Mining Datasets of Security Records

Claudia Cavallaro , Vincenzo Cutello , Mario Pavone  ,  
and Francesco Zito 

Department of Mathematics and Computer Science, University of Catania,  
V.le Andrea Doria 6, 95125 Catania, Italy  
{claudia.cavallaro,cutello}@unict.it, mpavone@dmf.unict.it,  
francesco.zito@phd.unict.it

**Abstract.** Cybersecurity bulletins officially recognize and publicly share the vulnerabilities of Information Systems. The attacks exploit various aspects of those vulnerabilities, compromising confidentiality, integrity or availability of the data collected. We analyze a public dataset of security records so to obtain some common features and to be able to forecast future attacks. We propose an intervention based on history of attacks through data mining methods and so a more dynamic risk analysis, by concentrating on some specific classes of cyberattacks in a period of two years. We devise a fast algorithm to find strong rules which provide an estimate of the probability that these attacks will occur so to identify adequate controls and countermeasures.

**Keywords:** Pattern analysis · Cyber security · Association Rules · Data Mining · Anomaly detection · Optimization

## 1 Introduction

Cyberattacks affect different sectors such as healthcare, government, financial and automotive industries. Incidents due to malware attacks impact industrial production and critical infrastructures, causing significant delays in control operations and consequent process anomalies.

Particularly for programmable cars, a compromise of the system can lead to risks to people safety, as well as to their privacy. Connected cars are targeted via Spear Phishing mechanisms which lead to the download of malicious attachments and payloads, or by Hardware Trojans which provide covert access to the onboard computer system and can disrupt communication of Controller Area Network buses. The vehicle can be affected by Ransomware attacks which encrypt user data causing operational disruptions. Via the infotainment system, the victim driver is threatened that the ignition of the car will be suspended until a ransom is paid.

Public data and technical reports regarding cybersecurity provide a description of vulnerabilities and exposures discovered over time, but the records contained in the published databases are very numerous if we consider long periods of time. Furthermore, faced with two or more vulnerabilities, it is generally not possible to decide which one is more urgent to deal with and, in particular, each vulnerability can have different impacts on different systems. Software developers are often forced to work within a limited time frame and are unable to analyze all security weaknesses. So they have to focus on targeting the most serious weaknesses or the ones related to specific characteristics, such as vulnerability metrics, type of exploits and so on.

It is therefore necessary to establish a priority among all the mitigation and detection measures to be adopted, on the basis of the frequent relationships among them, such as: basic metrics of vulnerability, weaknesses, attack tactics and techniques, operating systems or architectures.

We propose in this paper to simplify the standard and computational challenging general data mining problem of finding strong association rules by concentrating the search onto the prediction of specific attack and vulnerabilities, and in doing so create an information structure which can be easily updated.

Our work is organized as follows: Sect. 2 presents public security datasets, and some research work of data mining applied to the field of cybersecurity. Section 3 explains the methodology chosen to mine frequent patterns efficiently and prioritize actions to safeguard security. In Sect. 5 the results of our analysis show the forecasting of attacks based on past records. Section 6 concludes our study and outlines some future research directions.

## 2 Dataset and Background

Since 1999 the MITRE Corporation collects a catalog of known cybersecurity vulnerabilities and the NIST (National Institute of Standards and Technology) assigns to each of them a severity score, based on a standard called CVSS (Common Vulnerability Scoring System), and publishes them in the National Vulnerability Database (NVD), available online<sup>1</sup>. CVSS estimates the severity of a vulnerability and it is used by vendors, developers, researchers, security managers in companies and public administration and security agencies that deal with the publication of bulletins.

Common vulnerability and exposures entries, CVE for short, are reported with a unique identifier which is tagged with CVE-YYYY-XXXX, where YYYY is the year the vulnerability was discovered and XXXX is a sequential integer. The CVE archive, available online<sup>2</sup>, provides a description of the vulnerabilities included in MITRE reports. The CVE's perspective is to catalog errors after they have occurred and to investigate possible solutions. At the same time, MITRE is responsible for providing a list of CWE (Common Weakness Enumeration) to show the weaknesses in the architecture or in the code.

<sup>1</sup> <https://nvd.nist.gov>.

<sup>2</sup> <https://cve.mitre.org>.

All the catalogued CWE IDs are represented with a hierarchical tree organization, called “View 1000 - Research Concepts”<sup>3</sup>. The *Pillars* are the parents from whom the first branch starts, and which, then, describe common classifications of weaknesses.

CAPEC (Common Attack Pattern Enumeration and Classification), available online<sup>4</sup>, describes and classifies the attack patterns. The MITRE ATT&CK (Adversarial Tactics, Techniques and Common Knowledge) is a framework that describes all the main procedures used by attackers to violate systems and possibly gain persistent access to them. Attack procedures include tactics, which identify the attackers ultimate goals and the main purpose of their actions. Each attack tactic contains different techniques, which are concrete actions aimed at a specific goal and specify what an attacker achieves when finished. The MITRE ATT&CK matrices, available online<sup>5</sup> for the Enterprise, Mobile and ICS domains report the technical-tactics of violations and persistence of fixed corporate, mobile and industrial control systems.

ENISA, the European Network and Information Security Agency, aggregates the records from the official databases mentioned above and from other resources such as the Vulnerability Database (VULDB), online<sup>6</sup>, into a single .csv file. Each row contains information about these features: CVE ID, source database, severity level, impact score, exploitability score, attack vector, complexity, privilege, scope, confidentiality impact, integrity impact, availability impact, CWE ID, CAPEC ID, date published, attack technique ID and tactic.

In [12], ENISA presents a technical cybersecurity report about 2018–19, but it does not provide any prediction about future attacks. In this work we analyzed its aggregate information to establish what could be the next information (within some tolerance) that could be reported in the security bulletins. The ENISA statistics do not highlight the coexistence of vulnerabilities, weaknesses and attacks in frequent tuples of the dataset, features which are co-present in its rows according to a fixed minimum frequency.

We are interested in records that have common characteristics for a fixed minimum percentage of the analyzed data (a total of about 230k rows).

## 2.1 Data Mining and Cybersecurity

We will now overview some works that link data mining to the field of cybersecurity.

In [15] the authors, starting from the Record Audit data - Snort log, identify IP numbers and probable attacks, but they do not deal with the pattern detection of vulnerability features.

Fan et al. in [7] created a dataset by adding code changes and summary for C / C++ vulnerability to the CVE archive.

<sup>3</sup> <https://cwe.mitre.org/data/definitions/1000.html>.

<sup>4</sup> <https://capec.mitre.org>.

<sup>5</sup> <https://attack.mitre.org/matrices>.

<sup>6</sup> <https://vuldb.com>.

In [14], Murtaz et al. show that vulnerabilities can be treated like Markov Chains, and so they can predict the next vulnerability by using only the previous one.

The authors of [13] extract associations of words, used in websites for Cyber-security, through the well-known Apriori data mining algorithm.

Dodiya et al. [6] provide statistical distributions of the NVD, such as the number of new vulnerabilities reported by year, security levels, access complexity and integrity impact.

Threat searching can involve anomaly detection on machine logs, where behavioral data analysis is automatically separated from outliers using NLP and deep learning [4]. A Big Data Platform [16] was created to centralize collection of logs and metrics from heterogeneous data sources. It can be accessed so to perform a semi-supervised anomaly detection using the results of log clustering and visualize in real time the health of services through dashboards.

Anomaly detection finds application in many domains, including Cultural Heritage [8] and Urban Informatics [5]. In particular, data mining methods are also used to forecast next destinations [3].

### 3 Mining Association Rules

Let us start by introducing a mathematical formalization of the problem. Let  $\mathcal{D}$  be a dataset (matrix) with  $m$  rows and  $n$  columns. Each column represents a specific attribute  $ID_1, ID_2, \dots, ID_n$ , and each row represents a complete set of values for the  $n$  attributes. Any attribute  $ID_i$  and any of its values  $v$  found in the rows of  $\mathcal{D}$ , define the element  $\langle ID_i = v \rangle$ .

Given now any element  $I$ , the singleton  $\{I\}$ , also called 1-element itemset or itemset of length 1, is said to be “*infrequent*” if it is contained in a number  $k$  of rows of the dataset where  $\frac{k}{m} < \text{min\_supp}$ , i.e. is smaller than the fixed minimum support (we use the notation  $\text{supp}(\{I\}, \mathcal{D}) < \text{min\_supp}$ ). The minimum support represents then a fraction or percentage value of the rows of the dataset. If  $\text{supp}(\{I\}, \mathcal{D}) \geq \text{min\_supp}$  then  $\{I\}$  is said to be “*frequent*”. We generalize the above concept to itemsets of length  $h$  for any  $1 \leq h \leq n$ , as follows: an itemset of length  $h$  is a set of  $h$  elements,  $\{I_1, I_2, \dots, I_h\}$ , such that

- each element  $I_i$  represents the value of an attribute, i.e.  $I_i = \langle ID_{j_i} = a \rangle$  for some attribute  $ID_{j_i}$  and  $a$  one of the values of  $ID_{j_i}$ ;
- two distinct elements  $I_{i_1}$  and  $I_{i_2}$  represent values of two different attributes.

The frequency of the itemset  $\{I_1, I_2, \dots, I_h\}$  is the number of rows of  $\mathcal{D}$  which contain its values. As in the case of itemsets of length 1, the itemset is frequent if  $\text{supp}(\{I_1, I_2, \dots, I_h\}, \mathcal{D}) \geq \text{min\_supp}$ , otherwise is said to be infrequent.

Since  $\text{supp}(\{I_1, I_2, \dots, I_h\}, \mathcal{D}) \leq \text{supp}(S, \mathcal{D})$  for any  $S \subseteq \{I_1, I_2, \dots, I_h\}$ , it is clear that if  $\{I_1, I_2, \dots, I_h\}$  is frequent, all its subsets are also frequent. Thus, if any of its subsets is infrequent then the itemset is infrequent as well.

To clarify the above, let us consider the example of the dataset in Table 1. We have a dataset with 20 rows and 5 columns, corresponding at the attributes

$ID_1, ID_2, ID_3, ID_4, ID_5$ . If we choose  $min\_supp = 0.3$ , i.e. 30% of the total number of rows (6 in our case) the following elements, or 1-itemsets, are frequent (shown with their frequencies):

$$a1, 6; a2, 6; a3, 6; b1, 7; b2, 8; c1, 6; c2, 6; d3, 9; e4, 6.$$

The itemsets  $\{a1, b2\}$  and  $\{b1, c2, d3, e4\}$  have frequencies 6, so they are both frequent. The itemset  $\{a2, b1\}$ , instead, has frequency 3 and thus it is not frequent.

**Table 1.** Dataset with 5 attributes and 20 rows

$ID_1$	$ID_2$	$ID_3$	$ID_4$	$ID_5$
a1	b2	c3	d2	e1
a2	b1	c1	d3	e6
a1	b2	c1	d2	e1
a2	b1	c2	d3	e4
a1	b2	c1	d3	e2
a2	b1	c2	d3	e4
a1	b2	c3	d2	e2
a2	b3	c1	d5	e3
a4	b3	c3	d1	e2
a2	b4	c4	d3	e5
a1	b2	c4	d2	e2
a2	b2	c5	d6	e3
a3	b1	c2	d3	e4
a3	b4	c1	d1	e3
a3	b2	c1	d2	e2
a3	b1	c2	d3	e4
a1	b2	c3	d4	e3
a3	b5	c3	d1	e1
a3	b1	c2	d3	e4
a5	b1	c2	d3	e4

### 3.1 Mining Datasets

There are many algorithms available in literature for mining data and produce association rules. Given that the problem is clearly computationally challenging, many of these algorithms employ heuristics (see the excellent survey [9] for a comprehensive list of heuristics approach) or population based algorithms such as genetic algorithms (see for instance [17]) or particle swarm optimization (see [1]).

We briefly mention now the two most famous algorithms to find frequent itemsets. We start with Apriori [2], the most famous and first to be used algorithm for such a purpose, along with its successor FP-Growth [11]. It first computes the support of each single item and, then, it does the same for each itemset of cardinality 2, 3 and so on. In addition, the comparison of candidates for all rows becomes more expensive as the iterations of the algorithm increase and therefore the size of the itemsets to be generated increases. The Apriori algorithm requires  $l + 1$  scan of the dataset to find the longest patterns, of length  $l$ .

The second algorithm is Prefix-Span (PREFIX-projected Sequential PAttern mining), a data mining algorithm introduced by Pei et al. [10], which is used for marketing strategies.

Both algorithms would produce the entire collection of frequent itemsets. In our working example (itemsets are shown followed by their frequencies) the following itemsets are frequent:

$$\begin{aligned} &\{a1\} : 6; \{a2\} : 6; \{a3\} : 6; \{b1\} : 7; \{b2\} : 8; \{c1\} : 6; \{c2\} : 6; \{d3\} : 9; \{e4\} : 6; \\ &\{a1, b2\} : 6; \{c2, b1\} : 6; \{d3, b1\} : 7; \{e4, b1\} : 6; \{d3, c2\} : 6; \{c2, e4\} : 6; \{d3, e4\} : 6; \\ &\{d3, c2, b1\} : 6; \{c2, e4, b1\} : 6; \{d3, e4, b1\} : 6; \{d3, c2, e4\} : 6; \{d3, c2, e4, b1\} : 6 \end{aligned}$$

### 3.2 Association Rules and Confidence

The concept of Association Rules  $A \Rightarrow B$  was presented in [2] along with its related confidence value  $Confidence(A \Rightarrow B)$ , which represents, for instance, in market basket analysis the probability of buying a set of objects  $B$ , called consequent, given the purchase of a set of objects  $A$ , called antecedent, within the same transaction. More formally, given the probability distribution which generated the rows in the dataset,  $Confidence(A \Rightarrow B) = P(B|A)$ .

To generate an association rule  $A \Rightarrow B$ , where  $A$  and  $B$  are itemsets, we will take the support of  $A \cup B$ , and divide it by the support of  $A$ , thus computing, among the rows in the Dataset which contain  $A$ , the percentage of rows which contain also  $B$ .

If the itemsets satisfy two fixed parameters, that are the *min\_supp* and also the minimum value of Confidence  $c$  (see below), the predictions are called Strong Rules. So, formally we have

**Definition 1.** *Given a dataset  $\mathcal{D}$  and given two fixed parameters,  $0 \leq min\_supp \leq 1$  and the minimum value of Confidence  $0 \leq c \leq 1$ , and given two disjoint itemsets  $A, B$  such that  $supp(A \cup B, \mathcal{D}) \geq min\_supp$ , the association rule  $A \Rightarrow B$  is strong if  $\frac{supp(A \cup B, \mathcal{D})}{supp(A, \mathcal{D})} \geq c$ .*

In our work, we set as *Minimum Confidence* value  $c = 75\%$  to get only itemsets that have a higher (or equal) confidence and also a support that exceeds or equals the *Minimum Support* chosen (30%).

When searching for strong rules we pay particular attention to maximal frequent itemsets, i.e. itemsets which are frequent but such that by adding one more element would no longer be frequent. Thus, given a maximal frequent itemset

$M$  and any itemset  $B$  such that  $B \cap M = \emptyset$ , we know that the association rule  $M \Rightarrow B$ , will not be strong, since  $M \cup B$  is not frequent.

Going back to the example of Table 1, we have two maximal itemsets of cardinality greater than 1, namely  $\{b1, c2, d3, e4\}$  and  $\{a1, b2\}$ . The Association Rule  $\{d3\} \Rightarrow \{b1, c2, e4\}$  is not Strong because its confidence value is equal to 66.6%. Instead, the Association Rules  $\{e4\} \Rightarrow \{b1, c2, d3\}$ ,  $\{c2\} \Rightarrow \{b1, d3, e4\}$ , and  $\{b1\} \Rightarrow \{c2, d3, e4\}$  are all strong and, in particular, the first two have 100% confidence value while the last one 86%.

Table 2 shows the 15 strong association rules. In particular, rules 6, 7, 8, 12, 13, 14 are a consequence of the fact that rule 3 is strong. Same reasoning could be applied to the other rules which are a consequence of rules 4 and 5.

**Table 2.** Strong Rules for the maximal itemset of the example in Table 1

Rule n.	Antecedent	Consequent	Antecedent support	Itemset support	Confidence
1	{a1}	{b2}	6	6	100%
2	{b2}	{a1}	8	6	75%
3	{b1}	{e4, d3, c2}	7	6	86%
4	{c2}	{e4, b1, d3}	6	6	100%
5	{e4}	{c2, b1, d3}	6	6	100%
6	{b1, c2}	{e4, d3}	6	6	100%
7	{b1, d3}	{e4, c2}	7	6	86%
8	{b1, e4}	{c2, d3}	6	6	100%
9	{c2, d3}	{e4, b1}	6	6	100%
10	{c2, e4}	{b1, d3}	6	6	100%
11	{d3, e4}	{c2, b1}	6	6	100%
12	{b1, d3, e4}	{c2}	6	6	100%
13	{b1, c2, e4}	{d3}	6	6	100%
14	{b1, c2, d3}	{e4}	6	6	100%
15	{c2, d3, e4}	{b1}	6	6	100%

## 4 Mining Security Datasets for Decisively Strong Rules

In a field such as security, we are more interested in association rules where the antecedent is a set of events and the consequent is a specific type of attack. Same kind of reasoning may be applied in the medical field, where we are interested in diagnosing the likely disease given a list of symptoms.

So, we are considering the case that  $B$  contains a single element, i.e.  $B = \{id_j\}$  and  $A = \{id_1, id_2, \dots, id_i\}$  is an itemset with  $i$  elements non containing  $id_j$ . We have formally

$$\text{Confidence}(\{id_1, \dots, id_i\} \Rightarrow \{id_j\}) = P(B|A) = \frac{\text{supp}(\{id_1, \dots, id_i, id_j\}, \mathcal{D})}{\text{supp}(\{id_1, id_2, \dots, id_i\}, \mathcal{D})} \quad (1)$$

with  $\{id_1, \dots, id_i\} \cap \{id_j\} = \emptyset$ .

In other words, we would like to be able to infer which attack technique is likely being used, so to apply proper countermeasures. Obviously, such an ability is particularly important if the attack is not very common, i.e. the probability of such an attack, though frequent, is not likely or very likely.

Equation 1 gives us the probability that, given some specific attribute values  $id_1, id_2, \dots, id_i$  for weaknesses and vulnerabilities that occur as frequent itemsets, they will appear together with attack tactics and techniques  $id_j$  as maximal frequent itemsets. The greater the confidence the greater the reliability in forecasting a certain type of attack, and therefore priority will be given to defensive actions related to it.

In view of the above, let us define then, among the attributes in the dataset  $\mathcal{D}$  a specific attribute target  $T$ .

We introduce now the following definition, by recalling that any event whose probability is not higher than 0.5 is typically called *unlikely*.

**Definition 2.** *Given two fixed parameters,  $min\_supp$  and the minimum value of Confidence  $c$ , and given an itemset  $A$  and a single value  $I \notin A$  such that  $supp(A \cup \{I\}, \mathcal{D}) \geq min\_supp$ , the association rule  $A \rightarrow \{I\}$  is a Decisively Strong Rule (DSR for short), if  $\{I\}$  is frequent, i.e.  $min\_supp \leq supp(\{I\}, \mathcal{D})$  but unlikely, i.e.  $min\_supp \leq supp(\{I\}, \mathcal{D}) \leq 0.5$  and  $\frac{supp(A \cup \{I\}, \mathcal{D})}{supp(A, \mathcal{D})} \geq c$ .*

Our goal is to find all the decisively strong rules given the attribute target  $T$ , i.e. association rules  $A \rightarrow \{t_i\}$  where  $t_i$  is a frequent (at least 30%) but unlikely value of the attribute target  $T$ .

Since both  $A$  and  $\{t_i\}$  are frequent, i.e. their supports are both at least 30% of the rows of the dataset, it follows that if  $m$  are the rows of  $\mathcal{D}$ , since  $t_i$  is unlikely,  $supp(\{t_i\}, \mathcal{D}) = \alpha \cdot m$  with  $0.3 \leq \alpha \leq 0.5$ , while  $supp(A \cup \{T_i\}, \mathcal{D}) = \beta \cdot m$  with  $0.3 \leq \beta \leq \alpha$  then

$$\frac{supp(A \cup \{t_i\}, \mathcal{D})}{supp(\{t_i\}, \mathcal{D})} = \frac{\beta}{\alpha}$$

from which it follows that  $supp(A \cup \{t_i\}, \mathcal{D}) = \frac{\beta}{\alpha} supp(\{t_i\}, \mathcal{D})$ . Thus, we need to mine the sub-dataset where  $t_i$  occurs for itemsets with a minimum support of  $\frac{\beta}{\alpha}$ .

We notice that since  $\alpha \leq 0.5$  and  $\beta \geq 0.3$  we have

$$\frac{\beta}{\alpha} \geq \frac{0.3}{0.5} = 0.6$$

For instance, let us consider Table 1 and suppose our target is the value  $b1$  of  $ID2$ . The sub-table containing the value  $b1$  is shown in Table 3. Since the support of  $\{b1\}$  is  $\frac{7}{20} < 0.5$  we need to look for itemsets with support at least  $\frac{3}{10} \cdot \frac{20}{7} = \frac{6}{7}$ , and we find, as expected, just  $\{c2, d3, e4\}$ .



**Table 3.** Dataset for target  $b1$

$ID_1$	$ID_2$	$ID_3$	$ID_4$	$ID_5$
$a2$	$b1$	$c1$	$d3$	$e6$
$a2$	$b1$	$c2$	$d3$	$e4$
$a2$	$b1$	$c2$	$d3$	$e4$
$a3$	$b1$	$c2$	$d3$	$e4$
$a3$	$b1$	$c2$	$d3$	$e4$
$a3$	$b1$	$c2$	$d3$	$e4$
$a5$	$b1$	$c2$	$d3$	$e4$

The algorithm, called *DSR*, formally described in the pseudocode 1, takes as input the dataset  $\mathcal{D}$ , the minimum support value  $min\_supp$ , the confidence value  $c$ , a specific target attribute  $T$  and a frequent value  $t_i$  for  $T$ .

To explain how *DSR* works, we will use the following notations:

- $t_i$  will denote the singleton  $\{T = t_i\}$
- $\mathcal{D}(t_i)$  denotes the projections of the dataset  $\mathcal{D}$  on the value  $t_i$  for  $T$ , i.e. the dataset obtained eliminating all the rows where  $T \neq t_i$ .
- $supp(A, \mathcal{D}(t_i))$  the support of the itemset  $A$  in the dataset  $\mathcal{D}(t_i)$  while  $supp(A, \mathcal{D})$  is the support of the itemset  $A$  in the whole dataset  $\mathcal{D}$ .

Let us suppose that  $F = \{ID_i = x_i\}$  is the collection of frequent elements all of length 1, therefore for each element  $x \in F$ , we have  $supp(\{x\}, \mathcal{D}) \geq min\_supp$ . Let also  $F_T = \{t_1, \dots, t_h\}$  be the set of the frequent values of target attribute  $T$ . Thus, for each  $t_i \in F_T$  we have  $supp(\{t_i\}, \mathcal{D}) \geq min\_supp$ .

Our goal is to find all subsets  $F' \subseteq F$ , such that  $F' \Rightarrow t_i$  is a DSR for some  $t_i$  frequent value of  $T$ . So,

$$supp(F', \mathcal{D}(t_i)) \geq \frac{min\_supp}{supp(\{t_i\}, \mathcal{D})} \geq 50\% \quad \text{Searching condition}$$

$$\frac{supp(F' \cup \{t_i\}, \mathcal{D})}{supp(F', \mathcal{D})} > c \quad \text{Pruning condition}$$

DSR uses, as a subroutine, any fast algorithm to find maximal frequent itemsets but on possibly quite small sub-datasets. For our tests, we used Apriori.

**Algorithm 1.** Pseudo-code of DSR.

---

```

1: procedure DSR( $\mathcal{D}, min\_supp, c, T$ )
2:    $min\_supp = 0.3, c = 0.75$ 
3:   Compute  $F$  set of frequent elements,  $F_T$  set of frequent values of  $T$ ;
4:   for each attribute  $t_i \in F_T$  do
5:      $supp(t_i, \mathcal{D}) = \alpha_i$ 
6:      $F'(t_i) = \emptyset$ 
7:     for each  $x \in F$  do
8:       if  $supp(x, \mathcal{D}(t_i)) > \frac{min\_supp}{\alpha_i}$  then
9:         add  $x$  to  $F'(t_i)$ 
10:      end if
11:      Use General Algorithm to find max. freq. itemsets from  $F'(t_i)$  in  $\mathcal{D}_{t_i}$ 
12:      for each maximal frequent set  $A$  do
13:        if  $supp(A \cup \{t_i\}, \mathcal{D}) > c \cdot supp(A, \mathcal{D})$  then
14:          output DSR:  $A \Rightarrow t_i$ 
15:        end if
16:      end for
17:    end for
18:  end for
19: end procedure

```

---

## 5 Results

In order to predict future threats we divided the ENISA dataset into the set of vulnerabilities and exposures published up to December 31st of 2018 (training set) and the set of CVEs available for the first half of 2019 (testing set) for comparisons with the obtained prediction. We add a new feature column in the original ENISA dataset, and so processed the Pillars, as attribute targets, instead of the single CWE ID because they group the weaknesses in a more generic way and consequently the mitigation of the data predicted could be addressed on a wider range.

We set the minimum confidence to 75% and  $min\_supp$  to 30% and searched for decisively strong rules of the form  $\{id_1, \dots, id_k\} \Rightarrow \{attack\_technique\_id\}$ . For year 2018 we found just one attack technique with support between 0.3 and 0.5, namely  $T1027$  (*Obfuscated Files or Information*) with support value 39.99% and another attack technique  $T1148$  (*Impair Defenses: Impair Command History Logging*), whose support value is 66.25% therefore higher than 0.3 but not unlikely according to our definition.

For the target value  $T1027$ , we obtained 24 DSR but only two with an antecedent which are maximals, the following:

- $A = \{CVSS\_Complexity = Low, CVSS\_Scope = Unchanged, CWE\ Pillar = Improper\ neutralization, CAPEC = Leverage\ Alternate\ Encoding, Attack\ Tactic = Defense\ Evasion\}$
- $B = \{CVSS\_attack = Network, CWE\ Pillar = Improper\ neutralization, CAPEC = Leverage\ Alternate\ Encoding, Attack\ Tactic = Defense\ Evasion\}$

So, the two DSR found are  $A \Rightarrow \{T1027\}$  and  $B \Rightarrow \{T1027\}$ .

The total number of all strong rules (with  $\text{min\_supp}=0.3$  and  $\text{min\_conf}=0.75$ ) that we could have obtained with traditional data mining algorithm would have been 1073, so our procedure is way faster and it avoids many useless generation.

To test the accuracy of the found rules, we extracted the frequent itemsets of the testing set (the first semester of 2019) that contain the same  $T1027$  attack technique. By comparing the obtained prediction of the 2 DSR rules of 2018 with the restricted frequent itemsets of 2019, we obtained a perfect matching.

To justify, experimentally, our choice of considering just target values with support not higher than 50%, we use as an example the attack technique  $T1148$  (*Impair Defenses: Impair Command History Logging*) which has support 0.66.

From the sub-datasets containing the value  $T1148$  we searched for frequent itemsets with support  $(0.3/0.66) = 0.45$ . We found 303 frequent itemsets but only 5 maximal:

1. {CVSS\_severity = HIGH, CVSS\_scope = Unchanged, CAPEC = Subverting Environment Variable Values, Attack Tactic = Defense Evasion}
2. {CVSS\_complexity = Low, CVSS\_scope = Unchanged, CVSS\_availability = None, CAPEC = Subverting Environment Variable Values, Attack Tactic = Defense Evasion}
3. {CVSS\_complexity = Low, CVSS\_scope = Unchanged, CWE Pillar = Improper Neutralization, CAPEC = Subverting Environment Variable Values, Tactic = Defense Evasion}
4. {CVSS\_complexity = Low, CVSS\_scope = Unchanged, CVSS\_confidentiality = High, CVSS\_integrity = High, CAPEC = Subverting Environment Variable Values, Attack Tactic = Defense Evasion}
5. {CVSS\_attack = Network, CVSS\_complexity = Low, CVSS\_privileges = None, CVSS\_scope = Unchanged, CAPEC = Subverting Environment Variable Values, Attack Tactic = Defense Evasion}

The above 5 maximal frequent itemsets are the antecedent to 5 DSR with minimum confidence 0.75 and consequent  $T1148$ . After extracting the frequent itemsets of the first semester of 2019 which contain  $T1148$  we found that only 2 maximal itemsets out of 5 are also found for 2019. It follows that in this case the accuracy is only 40%.

## 6 Conclusion

In this work, we addressed the general data mining problem of finding strong association rules so to predict specific attacks and discover unknown vulnerabilities. We proposed a framework which takes into account frequent but not very likely attacks and proposed a fast way to compute strong association rules which turn out to be highly accurate. Our data-driven approach to deal with potential attacks in order of priority, could in future research be extended by experimentally setting the parameters of minimal support, confidence and likelihood of target values. Keeping into account past and recent work using population based methodologies [1, 17] and heuristics [9] a possible future works could involve population-based metaheuristics for the choice of such parameters.

## References

1. Agrawal, M., Mishra, M., Kushwah, S.P.S.: Association rules optimization using improved PSO algorithm. In: 2015 International Conference on Communication Networks (ICCN). IEEE (2015)
2. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. *ACM SIGMOD Rec.* **22**(2), 207–216 (1993)
3. Cavallaro, C., Verga, G., Tramontana, E., Muscato, O.: Suggesting just enough (Un)crowded routes and destinations. In: *CEUR Workshop Proceedings*, vol. 2706, pp. 237–251 (2020)
4. Cavallaro, C., Ronchieri, E.: Identifying anomaly detection patterns from log files: a dynamic approach. In: Gervasi, O., et al. (eds.) *ICCSA 2021. LNCS*, vol. 12950, pp. 517–532. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86960-1\\_36](https://doi.org/10.1007/978-3-030-86960-1_36)
5. Cavallaro, C., Vizzari, G.: A novel spatial-temporal analysis approach to pedestrian groups detection. *Procedia Comput. Sci.* **207**, 2364–2373 (2022)
6. Dodiya, B., Singh, U.K., Gupta, V.: Trend analysis of the CVE classes across CVSS metrics. *Int. J. Comput. Appl.* **183**(33), 23–30 (2021)
7. Fan, J., Li, Y., Wang, S., Nguyen, T.N.: A C/C++ code vulnerability dataset with code changes and CVE summaries. In: *Proceedings of the 17th International Conference on Mining Software Repositories*. ACM (2020)
8. Fouladvand, S., Osareh, A., Shadgar, B., Pavone, M., Sharafi, S.: DENSA: an effective negative selection algorithm with flexible boundaries for self-space and dynamic number of detectors. *Eng. Appl. Artif. Intell.* **62**, 359–372 (2017)
9. Ghafari, S.M., Tjortjis, C.: A survey on association rules mining using heuristics. *WIRES Data Min. Knowl. Discov.* **9**(4), e1307 (2019)
10. Han, J., et al.: PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: *Proceedings of the 17th International Conference on Data Engineering*, pp. 215–224. IEEE (2001)
11. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Min. Knowl. Discov.* **8**(1), 53–87 (2004)
12. Katos, V., et al.: State of vulnerabilities 2018/2019 : analysis of events in the life of vulnerabilities. European Network and Information Security Agency (2020). for Cybersecurity, E.U.A.
13. Li, Z., Li, X., Tang, R., Zhang, L.: Apriori algorithm for the data mining of global cyberspace security issues for human participatory based on association rules. *Front. Psychol.* **11**, 582480 (2021)
14. Murtaza, S.S., Khreich, W., Hamou-Lhadj, A., Bener, A.B.: Mining trends and patterns of software vulnerabilities. *J. Syst. Softw.* **117**, 218–228 (2016)
15. Saboori, E., Parsazad, S., Sanatkhani, Y.: Automatic firewall rules generator for anomaly detection systems with Apriori algorithm. In: *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. IEEE (2010)
16. Tisbeni, S.R., et al.: A big data platform for heterogeneous data collection and analysis in large-scale data centers. In: *Proceedings of International Symposium on Grids & Clouds 2021 — PoS (ISGC2021)*. Sissa Medialab (2021)
17. Yan, X., Zhang, C., Zhang, S.: Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. *Expert Syst. Appl.* **36**(2, Part 2), 3066–3076 (2009)