Iterative Selection of Categorical Variables for Log Data Anomaly Detection

Max Landauer[1](✉), Georg Höld[1], Markus Wurzenberger[1], Florian Skopik[1], and Andreas Rauber[2]

[1] Austrian Institute of Technology, Giefinggasse 4, Vienna, Austria
{max.landauer,georg.hoeld,markus.wurzenberger,florian.skopik}@ait.ac.at
[2] Vienna University of Technology, Favoritenstraße 9-11, Vienna, Austria
rauber@ifs.tuwien.ac.at

Abstract. Log data is a well-known source for anomaly detection in cyber security. Accordingly, a large number of approaches based on self-learning algorithms have been proposed in the past. Most of these approaches focus on numeric features extracted from logs, since these variables are convenient to use with commonly known machine learning techniques. However, system log data frequently involves multiple categorical features that provide further insights into the state of a computer system and thus have the potential to improve detection accuracy. Unfortunately, it is non-trivial to derive useful correlation rules from the vast number of possible values of all available categorical variables. Therefore, we propose the Variable Correlation Detector (VCD) that employs a sequence of selection constraints to efficiently disclose pairs of variables with correlating values. The approach also comprises of an online mode that continuously updates the identified variable correlations to account for system evolution and applies statistical tests on conditional occurrence probabilities for anomaly detection. Our evaluations show that the VCD is well adjustable to fit properties of the data at hand and discloses associated variables with high accuracy. Our experiments with real log data indicate that the VCD is capable of detecting attacks such as scans and brute-force intrusions with higher accuracy than existing detectors.

1 Introduction

Modern computer systems are permanently exposed to cyber threats, such as intrusions or denial-of-service attacks. Consequently, cyber security experts develop intrusion detection systems that monitor system behavior through analysis of continuously generated log events and autonomously disclose any malicious activity. Thereby, anomaly detection is particularly interesting, because it employs self-learning techniques that are capable of recognizing unknown attacks without the need for pre-existing or manually coded knowledge [4].

Log data is a suitable source for such techniques as it keeps track of almost all events and thus provides detailed insights into the state of a computer system. Most existing analysis techniques thereby focus on network traffic, because it

E. Bertino et al. (Eds.): ESORICS 2021, LNCS 12972, pp. 757–777, 2021.
https://doi.org/10.1007/978-3-030-88418-5_36

contains numeric features such as packet count or duration that fit well-known machine learning methods, e.g., support vector machines or neural networks. Few approaches use categorical variables, because they lack intuitive distance metrics and comprise of immense amounts of possible combinations [5,19].

However, categorical variables are common in system logs and complement the detection of anomalous events. In particular, variables such as user identifiers, IP addresses, service names, system operations, or program states, occur in regular patterns that are expected to persist over time as long as the system behavior remains steady. For example, services utilize specific subsets of all available system operations and execute them with particular relative frequencies. Unexpected deviations from such conditional occurrence distributions indicate a change of system behavior and should therefore be reported to the system operators as anomalies. Unfortunately, the selection of variables suitable for such a detection mechanism is non-trivial, because it usually relies on expert knowledge about the system at hand and is difficult to automatize.

We propose the Variable Correlation Detector (VCD) as a solution to aforementioned issues. The approach comprises of a sequence of selection constraints to reduce the search space and identify interesting correlations between categorical variables. In addition, the VCD reuses conditional distributions of value occurrences computed in the selection phase for the disclosure of deviations in a subsequent detection phase. Our approach has several advantages over state-of-the-art methods. First, it identifies interesting correlations independent from the total occurrences of the involved values, which is different to approaches based on frequent itemset mining [19]. This is especially important for the detection of stealthy attacks that only produce infrequent values. Second, our approach does not generate strict rules for value co-occurrences, but instead involves fuzzy rules that do not always have to be fulfilled by employing statistical tests on chunks of events. Third, our approach is designed for online detection in streams of log data, which is essential for application in real-world scenarios.

This paper presents the correlation selection and anomaly detection mechanisms of the VCD. An implementation is available online as part of our log-based anomaly detection system [21]. We summarize our contributions as follows:

- An iterative method for selecting useful correlations of categorical variables.
- An online anomaly detection technique based on identified correlations.
- An evaluation of our open-source implementation of the proposed concepts.

The remainder of this paper is structured as follows. Section 2 reviews the state-of-the-art of correlation analysis in categorical log data. In Sect. 3, we outline the concept of the VCD. We then provide details of our proposed correlation selection constraints in Sect. 4. We present the evaluation of our algorithm in Sect. 5 and discuss the results in Sect. 6. Finally, Sect. 7 concludes the paper.

2 Related Work

Research on association mining between categorical variables in database transactions has been ongoing for many years. One of the main issues prevalent in this field is the immense search space arising from the many possible combinations of variables and values [19]. Accordingly, approaches such as the well-known Apriori algorithm [1] are usually designed for efficient searching and pruning.

To enable outlier or anomaly detection in categorical data, it is usually necessary to adjust or extend association mining algorithms. For example, Narita and Kitagawa [15] propose techniques to detect records that fail to occur in expected associations and to compute outlier scores that are also suitable for speeding up the search. Khalili and Sami [12] show that the Apriori algorithm is suitable to be used for intrusion detection, in particular, by identifying critical states of industrial systems with sensor outputs as variables. One of the downsides of algorithms based on frequent itemset mining is that they require multiple passes over the data, which prevents online processing. Djenouri et al. [7] therefore propose a single-pass technique with improved parameter selection and use pruning to limit the search space to itemsets that cover the largest amount of events.

The problem with such approaches based on frequent itemset mining is that they omit infrequent values, because they are not interesting for the associations. Anomalies are then considered as infrequent combinations of otherwise frequent values [19]. However, infrequent values are important for anomaly detection, as long as they occur consistently with their associated values. Accordingly, Das and Schneider [6] replace rare values with placeholders and use conditional probabilities to disclose associations. While our approach also employs conditional probability distributions, we propose a sequence of selection steps rather than value replacement to reduce complexity without loosing precision.

Distance-based techniques are commonly used for anomaly detection in numeric data, however, it is non-trivial to compute distances between categorical values. Eiras-Franco et al. [8] solve this problem by encoding categories as binary vectors to apply maximum likelihood analysis. Similarly, one-hot encoding is also used by Moustafa and Slay [14], who measure the association strength between variables using the Pearson correlation coefficient as well as Information Gain. Ren et al. [18] support anomaly detection on data streams by computing cluster references on chunks of data, where a distance function based on value equality is used. Our approach also analyzes chunks of data rather than individual lines, but employs statistical tests on conditional probability distributions.

A different strategy to tackle the lack of a distance metric and large event space is pursued by Chen et al. [5], who embed the data in a latent space and mine associations between pairs of variables, which include user IDs, IP addresses, and URLs. Similarly, Pande and Ahuja [16] use an embedding method based on word2vec for anomaly classification in HTTP logs. Alternatively, Ienco et al. [11] measure the similarity between value co-occurrences by applying distance metrics on their conditional occurrence probabilities. The advantage of this method is that it enables anomaly score computation for ranking. Conditional probabilities are also used by Tuor et al. [20], who show that neural networks are suitable for anomaly detection in categorical user data. We argue that the downside of these approaches is that they suffer from lower explainability than frequent itemset methods, where variable associations are more intuitive.

Table 1. Value co-occurrences of syscall types and items in Audit logs.

Items	Syscall type									Σ
	0	1	2	20	42	49	59	90	105	
0	6097	860	0	189	34	14	0	0	1	7195
1	0	0	2592	0	104	0	0	5	0	2701
2	0	0	90	0	0	0	14	0	0	104

Most aforementioned approaches rely on the assumption that their data involves only categorical variables or that these variables have been manually pre-selected. However, log files involve various data types, including discrete, continuous, static, and unique variables. Gupta and Kulariya [9] therefore use a Chi-squared test to select variables with sufficiently distinct value co-occurrences before comparing regression, support vector machines, naive bayes, and decision trees for anomaly detection. Our approach employs a sequence of constraints to limit the search space and then makes use of statistical tests to disclose anomalies. In the following section, we outline an overview of this procedure.

3 Concept

This section outlines the concept of the Variable Correlation Detector (VCD). First, we explain important aspects of correlations of variables. Then, we state definitions relevant for this paper and outline the overall procedure of the VCD.

3.1 Correlations of Variables

Log data are chronological sequences of events. Most log data sets comprise of a certain number of different event types, where each type defines the syntax of the corresponding log lines. Accordingly, simple log data such as comma-separated-values only consist of a single event. In any way, each event type specifies a sequence of variables or features. For example, the syscall event in Audit logs consists of a sequence of key-value pairs, such as “syscall=2” that specifies the syscall type or “items=1” that specifies the number of associated path records.

Some variables are strongly correlated, meaning that the occurrence of a value in one variable indicates the occurrence of a specific value in another variable. Given a sufficiently large time frame, these conditional probabilities should be more or less constant on a system with stable behavior. Any changes to these occurrence patterns indicate potentially malicious activities, i.e., anomalies.

Table 1 shows the number of occurrences of syscall types and items extracted from 10000 Audit logs that are also used in the evaluation in Sect. 5.1. With 7195 total occurrences, the majority of these events involve “items = 0” (sum of first row). However, it is visible that syscall type 2 (“open”) mostly occurs with “items=1” (2592 occurrences) and sometimes “items = 2” (90 occurrences), but never with “items = 0”. Since other value pairs exhibit similar dependencies, it

Table 2. Definitions of symbols used in this paper.

Symbol	Definition
E	Log event type from the set of all event types $\mathcal{E}$, i.e., $E \in \mathcal{E}$.
V_i	Variable of log event type E, with $V_1, \ldots, V_n \in E$.
$\mathcal{V}_i$	Set of distinct values attained by V_i.
$v_{i,j}$	Value j of variable V_i, i.e., $\mathcal{V}_i = \left\{v_{i,1}, \ldots, v_{i,m_i}\right\}$.
$P\left(v_{i,j}\right)$	Probability that value j occurs in V_i.
$P\left(v_{i,j} \mid v_{k,l}\right)$	Probability that value j occurs in V_i given that value l occurs in V_k.
$V_i \rightsquigarrow V_k$	Correlation between variables V_i and V_k.
$v_{i,j} \rightsquigarrow v_{k,l}$	Correlation between values of variables, i.e., occurrence of value j in V_i correlates with value l of V_k.
θ_i	Threshold parameter for correlation selection.
N	Size of the sample for computing correlations during initialization.
M	Size of the sample for updating and testing in online mode

is reasonable to monitor the conditional probability distributions of the variable "items" with respect to "syscall" for improved detection over monitoring the occurrences of "items" alone. The same reasoning applies for the other direction, i.e., monitoring the occurrences of syscall types given the number of items.

Different to existing approaches, we do not only focus on the selection of variables that are suitable for such correlations, but monitor the co-occurrences of their values. Thereby, we are not solely interested in frequent values or value combinations, but instead calculate the conditional probability distributions of all values that are useful for anomaly detection. Consider syscall type 59 ("exec") as an example: Even though the value only occurs in 14 events, it always co-occurs with "items = 2" and thus indicates a strong correlation. Due to the large number of possible combinations of variables and distinct values, a brute-force solution is computationally not feasible in practice, especially for high-volume log data with diverse values. This paper therefore presents an iterative selection strategy for interesting correlations that is presented in the following sections.

3.2 Definitions

As mentioned in the previous section, most log files comprise of several events $\mathcal{E}$, each containing a unique set of variables. For simplicity, we only consider a single event $E \in \mathcal{E}$ in the following and assume that the procedure is applicable to all other events analogously. Moreover, we assume that event E involves n variables $V_1, .., V_n$, each comprising of an arbitrary number of values $v_{i,1}, ..., v_{i,m_i}$ from the unique value set $\mathcal{V}_i$. We compute the estimated value occurrence probability as $P(v_{i,j}) = \left|\{V_i = v_{i,j}\}\right| / N$ in a sample of size N and the conditional probabilities as $P\left(v_{i,j} \mid v_{k,l}\right) = \left|\{V_i = v_{i,j} \wedge V_k = v_{k,l}\}\right| / \left|\{V_k = v_{k,l}\}\right|$. Correlations are denoted using the $\rightsquigarrow$ operator. Table 2 summarizes all symbol definitions.

3.3 Procedure

Our approach selects variable fields of log events and performs statistical tests on value occurrences in these fields for the purpose of anomaly detection. To limit

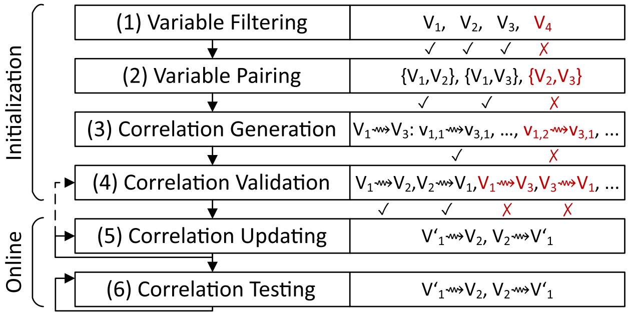

Initialization
Online
(1) Variable Filtering
V_1, V_2, V_3, V_4
✓ ✓ ✓ ✗
(2) Variable Pairing
$\{V_1,V_2\}$, $\{V_1,V_3\}$, $\{V_2,V_3\}$
✓ ✓ ✗
(3) Correlation Generation
$V_1 \rightsquigarrow V_3$: $v_{1,1} \rightsquigarrow v_{3,1}$, …, $v_{1,2} \rightsquigarrow v_{3,1}$, …
✓ ✗
(4) Correlation Validation
$V_1 \rightsquigarrow V_2$, $V_2 \rightsquigarrow V_1$, $V_1 \rightsquigarrow V_3$, $V_3 \rightsquigarrow V_1$, …
✓ ✓ ✗ ✗
(5) Correlation Updating
$V'_1 \rightsquigarrow V_2$, $V_2 \rightsquigarrow V'_1$
(6) Correlation Testing
$V'_1 \rightsquigarrow V_2$, $V_2 \rightsquigarrow V'_1$

Fig. 1. Procedure of the Variable Type Detector. Correlations between variables and values are filtered iteratively.

the search space, we propose several sequential analysis steps that act as filters for all possible variable and value combinations. Figure 1 shows these steps as a state chart, with an in-depth description of each step following in Sects. 4.2-4.6.

For the initialization phase in steps (1)–(4), the VCD first collects a sample of N log lines. We assume that all available variables of a log event are possible choices for correlations and that there is no manual pre-selection. Step (1) *Variable Filtering* sorts out variables that are unlikely to yield useful correlations, such as variables with many unique or static values. This step is exemplarily visualized in the figure by removing variable V_4 for subsequent analyses steps.

Step (2) *Variable Pairing* then generates pairs of the remaining variables V_1, V_2, V_3. This step removes pairs with dissimilar value probability distributions or disjoint value sets. In the figure, the variable pair $\{V_2, V_3\}$ is not considered for correlation. Remaining pairs are transformed to correlation hypotheses in step (3) *Correlation Generation*, where conditional occurrence probabilities of all involved values are computed. Correlations between values denoted by $\rightsquigarrow$ that exhibit weak associations are omitted, e.g., values that occur in many combinations or have similar conditional probabilities to other correlated values. In the figure, value correlation $v_{1,2} \rightsquigarrow v_{3,1}$ of variable correlation $V_1 \rightsquigarrow V_3$ is removed. Note that correlations are directed, i.e., $V_1 \rightsquigarrow V_3$ is different from $V_3 \rightsquigarrow V_1$.

Step (4) *Correlation Validation* then evaluates whether all resulting value correlations indicate a sufficiently strong dependency between the correlated variables, in particular, whether the valid value correlations have independent probability distributions and involve sufficiently many occurring values. For example, assuming that several value correlations such as $v_{1,2} \rightsquigarrow v_{3,1}$ were removed in step (3), the variable correlation $V_1 \rightsquigarrow V_3$ is removed. This marks the end of the initialization phase, which is only executed once for every log event type.

For online anomaly detection, all correlation hypotheses that remain after step (4) are transformed into rules, which are repeatedly evaluated using samples of size M. For this, we perform statistical tests in step (5) *Correlation Updating* and go back to step (3) to re-initialize the correlation rules if value distributions change or new values appear, e.g., V_1 is replaced by V_1' in Fig. 1. Once correlation rules are stable for a sufficiently long time period and should not be updated anymore, they are tested in step (6) for the purpose of anomaly detection.

Table 3. Sample data.

ID	V_1	V_2	V_3	V_4	ID	V_1	V_2	V_3	V_4	ID	V_1	V_2	V_3	V_4
1	1	1	1	1	5	1	1	3	1	9	3	3	3	1
2	1	1	2	1	6	2	2	2	2	10	2	2	1	1
3	1	1	3	3	7	1	2	1	1	11	2	2	1	3
4	1	2	1	1	8	2	3	2	1	12	1	1	2	1

4 Approach

This section presents detailed explanations of all aforementioned steps of the VCD procedure. We also provide examples for the various selection criteria.

4.1 Sample Data

We provide a small sample to make the equations in the following sections easier to understand and to obtain a rough estimate for reasonable choices for threshold parameters θ_i. The data shown in Table 3 comprises of one event with four variables, i.e., $E = \{V_1, V_2, V_3, V_4\}$, and a sample size of $N = 12$. We point out that this data is only for illustrative purposes and that the application of the VCD in practice requires sufficiently large sample sizes for appropriate probability estimation. Each variable involves three possible values, in particular, $\mathcal{V}_1 = \mathcal{V}_2 = \mathcal{V}_3 = \{1, 2, 3\}$. The occurrence probabilities of the values of V_1 are computed as $P(v_{1,1}) = \frac{7}{12}, P(v_{1,2}) = \frac{4}{12}, P(v_{1,3}) = \frac{1}{12}$. The data is set up so that V_1 and V_2 correlate, i.e., the occurrence of any value in V_1 usually co-occurs with the same value in V_2. This is also reflected in the conditional probabilities, e.g., $P(v_{2,1} \mid v_{1,1}) = \frac{5}{7}, P(v_{2,2} \mid v_{1,2}) = \frac{3}{4}$. On the other hand, V_3 and V_4 do not show a strong correlation with any other variable. Accordingly, the following examples usually set the thresholds θ_i so that correlations involving V_3 and V_4 are removed, but $V_1 \rightsquigarrow V_2$ and $V_2 \rightsquigarrow V_1$ are selected as relevant for detection.

4.2 Variable Filtering

This section covers heuristics for variables. The first criterion targets variables with many unique values and the second criterion addresses dominating values.

Diversity of Values. Correlation analysis as it is done by the VCD requires categorical variables to reasonably calculate occurrence probabilities from the sample. Accordingly, we assume that there is a finite number of different values attained by each variable and that the sample size is large enough to obtain an estimate on their occurrence probabilities, i.e., $|\mathcal{V}_i| \ll N$. Variables with a large number of unique values are likely discrete rather than categorical, e.g., event IDs or timestamps, and do not yield stable correlations as described in Sect. 3.1. The reason for this is that they result in a high number of infrequent value co-occurrences that do not represent any actual correlation between the variables, e.g., an event ID is usually a random value. Equation 1 thus defines an upper limit for the number of unique values in V_i, where $\theta_1 \in [0, 1]$. From all available variables, we select all V_i that fulfill Eq. 1, and omit all others.

$$|\mathcal{V}_i| \leq \theta_1 \cdot N \tag{1}$$

The small sample size of the data in Table 3 requires $\theta_1 \geq 0.25$ to retain the variables, e.g., $\theta_1 = 0.25$ yields a critical value of 3 and $|\mathcal{V}_1| \leq 3$ is fulfilled.

Distribution Probabilities. In some variables, one or few values are occurring more often than others and are thus dominating the value probability distribution. These variables usually have weaker correlation with other variables, since most correlated values co-occur with the same dominating value. An extreme case of this situation are static variables, where the same value occurs in every log line and is thus trivially useless for correlation. We therefore use Eq. 2 to select only variables V_i where no occurrence probability of $v_{i,j}$ exceeds a certain limit. To allow more unique values $\theta_2 \in [0, 1]$ should be selected closer to 1.

$$P\left(v_{i,j}\right) < \theta_2 + \frac{1 - \theta_2}{|\mathcal{V}_i|} \tag{2}$$

We point out that this heuristic causes that variables with similarly dominated value probability distributions that may have a strong association between the values are omitted. Since this heuristic is mainly used to efficiently limit the search space, it is possible to set θ_2 to a sufficiently large value to include these variables and use subsequent analysis steps to omit incorrect variable pairings.

The data from Table 3 involves value $v_{4,1}$ which dominates V_4. Setting $\theta_2 = 0.6$ excludes only this variable, since $P\left(v_{4,1}\right) = 0.75$ exceeds $0.6 + \frac{1-0.6}{3} = 0.73$.

4.3 Variable Pairing

This section describes criteria for selecting pairs of variables suitable for correlation. The first criterion matches variables with similar probability distributions and the second criterion addresses common value spaces.

Similarity of Distributions. As pointed out in the previous section, variables with similar value probability distributions are more likely to exhibit associations between their values than other variable pairs. The reason for this is that similar distributions imply that for each value in V_i there exists another value in V_k that occurs roughly the same amount of times and may thus have a direct relationship with the former value. On the other hand, comparing the value occurrences of one dominated distribution and another evenly distributed distribution, there is necessarily at least one value in one variable that co-occurs with more than one value in another variable, which indicates a weaker association.

We therefore generate variable pair $\{V_i, V_k\}$ if the occurrence probabilities $P\left(v_{i,j}\right)$ of all values in V_i do not differ from $P\left(v_{k,l}\right)$ in V_k, where each value is only used once. Equation 3 describes this rule formally, where $\theta_3 \in [0, \infty)$ and $p = 1, ..., min(|\mathcal{V}_i|, |\mathcal{V}_k|)$ is the index of the order statistic so that $v_{i,(1)}$ is the most occurring value of V_i, $v_{i,(2)}$ is the second most occurring value of V_i, etc.

$$\left|P\left(v_{i,(p)}\right) - P\left(v_{k,(p)}\right)\right| \leq \frac{\theta_3}{max(|\mathcal{V}_i|, |\mathcal{V}_k|)} \tag{3}$$

Setting $\theta_3 = 0.6$ yields a critical value of $\frac{0.6}{3} = 0.2$. In this case, variables V_1 and V_2 from Table 3 are correctly paired, since all probability differences $|P(v_{1,1}) - P(v_{2,1})| = 0.16$, $|P(v_{1,2}) - P(v_{2,2})| = 0.08$, and $|P(v_{1,3}) - P(v_{2,3})| = 0.08$ are lower than 0.2, where values are compared in decreasing order of their occurrences. Assuming that V_4 is not removed in the variable filtering phase, the pair $\{V_2, V_4\}$ is omitted since $|P(v_{2,1}) - P(v_{4,1})| = 0.33$ which exceeds 0.2.

Common Values. Another heuristic is that variables sharing common values are likely related in some way. For example, log lines that involve separate variables for source and destination IP addresses often have the same value space, since data is sent and received from the same IP addresses. This also applies to state transitions in logs, such as network logs that contain messages like "inactive –> scanning", "scanning –> authenticating", etc. As an alternative in case that Eq. 3 is not fulfilled, we select pairs $\{V_i, V_k\}$ where both variables share a certain fraction of common values. This corresponds to selecting variable pairs that fulfill Eq. 4, where $\theta_4 \in [0, 1]$. For the sample data displayed in Table 3, this constraint is trivially fulfilled since all variables have the same value space.

$$|\mathcal{V}_i \cap \mathcal{V}_k| \geq \theta_4 \cdot min(|\mathcal{V}_i|, |\mathcal{V}_k|) \tag{4}$$

4.4 Correlation Generation

This section outlines the generation of correlation hypotheses for values of variable pairs. Note that each pair $\{V_i, V_k\}$ is considered as the two hypotheses $V_i \rightsquigarrow V_k$ and $V_k \rightsquigarrow V_i$ that are analyzed separately.

Diversity of Correlations. For optimal variable correlation, each value of one variable only occurs with a particular value of another variable and vice versa. Conversely, values that co-occur with many different values from the correlated variable indicate weak or random associations as pointed out in Sect. 4.2 and should not be considered for correlation hypotheses. We therefore select only value correlations $v_{i,j} \rightsquigarrow v_{k,l}$ for hypothesis $V_i \rightsquigarrow V_k$ if the relative amount of co-occurring values of $v_{i,j}$ does not exceed $\theta_5 \in [0, 1]$, i.e., if Eq. 5 is fulfilled.

$$\frac{|\{v_{k,l} : P\left(v_{k,l} \mid v_{i,j}\right) > 0\}|}{|\mathcal{V}_k|} \leq \theta_5 \tag{5}$$

Selecting $\theta_5 = 0.7$ for the data from Table 3 yields that $v_{1,1} \rightsquigarrow v_{2,l}$ of $V_1 \rightsquigarrow V_2$ are fulfilled for all l, since $v_{1,1}$ only occurs with $v_{2,1}, v_{2,2}$ and $\frac{2}{3} \leq \theta_5$. Similarly, $v_{1,2} \rightsquigarrow v_{2,l}$ yield $\frac{2}{3}$ and $v_{1,3} \rightsquigarrow v_{2,l}$ yield $\frac{1}{3}$, thus all possible value correlations from $V_1 \rightsquigarrow V_2$ are selected. On the other hand, all $v_{1,1} \rightsquigarrow v_{3,l}$ of $V_1 \rightsquigarrow V_3$ are omitted since $v_{1,1}$ co-occurs with three values in V_3 and $\frac{3}{3}$ exceeds θ_5.

Skewness of Distributions. If Eq. 5 from the previous section is not fulfilled, we use an alternative selection constraint to avoid that useful correlations are omitted too easily. In particular, we check the shape of the conditional distributions to identify dependencies between values, i.e., if one of the values in V_k occurs with relatively high frequency given that $v_{i,j}$ occurs, we add $v_{i,j} \rightsquigarrow v_{k,l}, \forall l$ to the hypothesis $V_i \rightsquigarrow V_k$. Equation 7 shows that this constraint is realized by subtracting the highest from the lowest of all conditional probabilities given $v_{i,j}$ (cf. Eq. 6), where $\theta_6 \in [0, \infty)$. The idea behind this is that the difference is large for skewed distributions where some values co-occur frequently and others only rarely, and small for evenly distributed values. Note that this does not take into consideration that dominating values in V_k could incorrectly cause that the constraint is fulfilled, which is addressed in the following section.

$$\mathcal{P}_{i,j,k} = \{P(v_{k,l} \mid v_{i,j}) : P(v_{k,l} \mid v_{i,j}) > 0, \forall l\} \tag{6}$$

$$\max(\mathcal{P}_{i,j,k}) - \min(\mathcal{P}_{i,j,k}) > \frac{\theta_6}{|\{v_{k,l} : P(v_{k,l} \mid v_{i,j}) > 0\}|} \tag{7}$$

We use $\theta_6 = 0.8$ as a sample for the data in Table 3 and assume that $v_{1,2} \rightsquigarrow v_{2,l}, \forall l$ was omitted by the constraint from Eq. 5. Then $P(v_{2,1} \mid v_{1,1}) - P(v_{2,2} \mid v_{1,1}) = 0.42$ and $P(v_{2,2} \mid v_{1,2}) - P(v_{2,3} \mid v_{1,2}) = 0.5$ both exceed the critical value of $\frac{0.8}{2} = 0.4$. However, $v_{1,1} \rightsquigarrow v_{3,l}, \forall l$ is not fulfilled, because $P(v_{3,1} \mid v_{1,1}) - P(v_{3,3} \mid v_{1,1}) = 0.14$ does not exceed the critical value of $\frac{0.8}{3} = 0.27$ and is therefore correctly omitted from hypothesis $V_1 \rightsquigarrow V_3$.

4.5 Validation of Correlations

This section presents hypothesis validation constraints that omit correlations without sufficiently strong dependencies between values or few correlating values.

Dependencies of Distributions. As pointed out in Sect. 4.4, a valid correlation $V_i \rightsquigarrow V_k$ should imply that the conditional value probabilities $P(v_{k,l} \mid v_{i,j})$ differ from each other depending on the value $v_{i,j}$ attained by V_i. Otherwise, the values in V_k are independent from the attained values in V_i, which means that the correlation hypothesis should be discarded. We address this by measuring the variances of all conditional distributions in V_k with respect to the overall distribution of V_k. Equation 8 shows that the variances are added for all value correlations selected by one of the constraints from Sect. 4.4. Since variances of more frequently occurring value correlations are more representative for the variable and should therefore have a higher influence on the result, Eq. 9 with $\theta_7 \in [0, \infty)$ weights all variances by the occurrence probabilities of $v_{i,j}$ and checks whether their sum exceeds a threshold. In this case, the conditional distributions involved in the correlation hypothesis are sufficiently dependent and thus selected, otherwise the correlation is omitted from further analysis.

$$\mathbb{V}_k(v_{i,j}) = \sum_l \left\{ (P(v_{k,l} \mid v_{i,j}) - P(v_{k,l}))^2 : v_{i,j} \rightsquigarrow v_{k,l} \right\} \tag{8}$$

$$\sum_{j} \left\{ \mathbb{V}_k \left(v_{i,j} \right) \cdot P \left(v_{i,j} \right) : v_{i,j} \rightsquigarrow v_{k,l} \right\} \geq \theta_7 \tag{9}$$

We first consider correlation $V_1 \rightsquigarrow V_2$ from Table 3 as an example and use $\theta_7 = 0.2$ as a threshold. The variances $\mathbb{V}_2 \left(v_{1,1} \right) = 0.13$, $\mathbb{V}_2 \left(v_{1,2} \right) = 0.29$, and $\mathbb{V}_2 \left(v_{1,3} \right) = 1.04$ are weighted by probabilities $P \left(v_{1,1} \right) = 0.58$, $P \left(v_{1,2} \right) = 0.33$, and $P \left(v_{1,3} \right) = 0.08$ respectively to yield a total of 0.26 that exceeds $\theta_7 = 0.2$. Accordingly, the conditional value distributions in V_2 sufficiently depend on the attained values in V_1, thus $V_1 \rightsquigarrow V_2$ is selected as a valid correlation. On the other hand, the weighted sum of variances for $V_3 \rightsquigarrow V_1$ yields 0.06, which does not exceed the threshold and thus indicates that the correlation should be omitted.

Value Coverage. The second selection criterion for value correlations from one of the constraints from Sect. 4.4 ensures that only variable correlations supported by sufficiently many correlating values are selected. In other words, a correlation $V_i \rightsquigarrow V_k$ is omitted if only a small fraction of the values in V_i have corresponding correlations. Thereby, we use the occurrence probabilities of $v_{i,j}$ to weight frequent values higher. According to Eq. 10, we only select $V_i \rightsquigarrow V_k$ if the relative amount of correlating values exceeds a threshold $\theta_8 \in [0, 1]$.

$$\sum_{j} \left\{ P \left(v_{i,j} \right) : v_{i,j} \rightsquigarrow v_{k,l} \right\} \geq \theta_8 \tag{10}$$

We use data from Table 3 and consider the variable correlation $V_1 \rightsquigarrow V_3$ with $\theta_8 = 0.5$. We assume that all correlations from $v_{1,1}$ to values from V_3 were removed as outlined in the example in Sect. 4.4, but correlations from $v_{1,2}$ and $v_{1,3}$ to V_3 exist. The sum of probabilities for these values is then $P \left(v_{1,2} \right) + P \left(v_{1,3} \right) = 0.416$. Since this sum does not exceed the threshold of 0.5, correlation $V_1 \rightsquigarrow V_3$ is omitted from further analysis. Assuming that all value correlations were selected for $V_1 \rightsquigarrow V_2$ the constraint is trivially fulfilled since the sum of all occurrence probabilities always equals 1 and thus exceeds the threshold.

4.6 Correlation Updating and Testing

The previous sections outlined the initialization phase of the VCD, where correlations are selected by a sample of N log lines. Afterwards, the VCD switches to online mode, where samples of M log lines are repeatedly collected and tested with respect to the previously generated correlation rules. In the following, we use $\widetilde{P}$ to denote occurrence probabilities of values from these test samples. We use a two-sample Chi-squared test for homogeneity [3] to determine whether a test sample corresponds to the rules. For this, we first compute a test statistic t by comparing the conditional probabilities of the training and test samples with the expected probability P_e based on the mean as shown in Eq. 11 and Eq. 12.

$$P_e = \frac{N \cdot P \left(v_{k,l} \mid v_{i,j} \right) + M \cdot \widetilde{P} \left(v_{k,l} \mid v_{i,j} \right)}{N + M} \tag{11}$$

$$t = \sum_{l} \left(N \cdot \frac{\left(P\left(v_{k,l} \mid v_{i,j}\right) - P_e\right)^2}{P_e} + M \cdot \frac{\left(\widetilde{P}\left(v_{k,l} \mid v_{i,j}\right) - P_e\right)^2}{P_e} \right) \tag{12}$$

For a given $v_{i,j}$, we then define an indicator function $I_k(v_{i,j})$ in Eq. 13 that is 1 if the test statistic does not exceed a critical value given by the Chi-squared distribution with confidence $\alpha_1 \in [0,1]$, i.e., there is no significant difference between the conditional distributions of the training and test samples, and is 0 otherwise. We then store these indicators for all $v_{i,j} \in \mathcal{V}_i$ in a list $r_{i,j}$, so that $r_{i,j}^{(t-d)}, ..., r_{i,j}^{(t)}$ are the d most recent indicators after t tests of $v_{i,j}$, and compute another test statistic $s_{i,j}^t = \sum_{x=t-d}^{t} r_{i,j}^{(x)}$ on these values. The purpose of this is to reduce the number of false positives, i.e., anomalies are only reported when a certain number of Chi-squared tests fail. Since r is a binomial process, we use Eq. 14 to compute a critical value λ, where $\alpha_2 \in [0,1]$ is the confidence of the binomial test and α_1 is reused as the success probability of the Chi-squared test. If $s_{i,j}^t \geq \lambda$ holds, there is no significant change of the conditional probabilities of $v_{i,j} \rightsquigarrow v_{k,l}, \forall l$ at test t, and vice versa. Note that the runtime can be reduced by computing λ a single time in advance when d, α_1, and α_2 remain constant.

$$I_k(v_{i,j}) = \begin{cases} 1 & \text{if} \quad t < \chi^2_{\alpha_1, |\mathcal{V}_k|-1} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\lambda = min \left\{ k_{max} : \sum_{k=0}^{k_{max}} \frac{d!}{k! \cdot (d-k)!} \cdot \alpha_1^k \left(1 - \alpha_1\right)^{d-k} > 1 - \alpha_2 \right\} \tag{14}$$

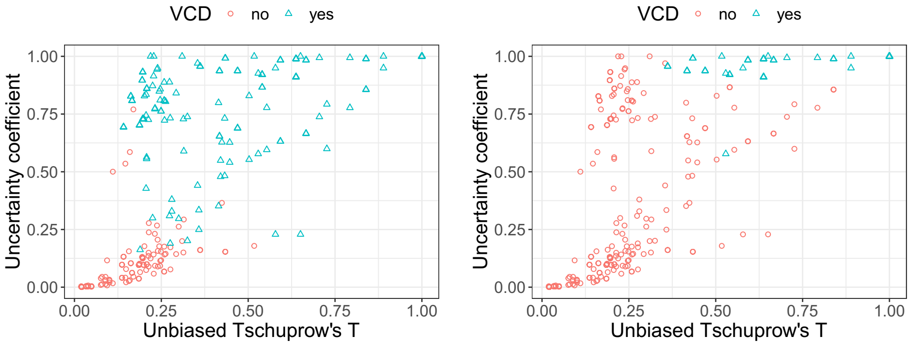
The aforementioned computations are carried out for updating as well as testing correlations. The main difference between both phases is that step (6) *Correlation Testing* only reports anomalies when tests fail, i.e., $s < \lambda$, meaning that all changes of correlations are reported every time after processing the test samples as long as they persist. On the other hand, step (5) *Correlation Updating* adjusts the base line for comparison by updating distributions with newly observed values, removing correlations if the binomial test fails, and periodically repeating steps (3)–(4) to identify new correlations. Accordingly, this phase is seen as an extended training phase that is essential for online learning.

5 Evaluation

This section outlines the evaluation of our approach. We first compare variable correlations selected from a real data set with two well-known correlation metrics. We then showcase the detection capabilities of the VCD.

5.1 Comparison with Association Metrics

This section compares the selected correlations of the VCD with well-known association metrics. We first describe the data and then show the results.



(a) Default thresholds.

(b) $\theta_1 = 0.01$ and $\theta_7 = 0.2$.

Fig. 2. Comparison of VCD selection with association metrics.

Data. We use 10000 Audit logs of type syscall from the publicly available AIT-LDSv1.1 [13] for this evaluation. We select Audit logs, because they are a common source for log analysis and are sufficiently complex to be representative for all kinds of log data. In addition, they contain many categorical variables that correlate with varying strength and have diverse value occurrence distributions. Out of all 27 variables, we remove the timestamp as well as 6 static variables that only attain one value, because it is not possible to generate useful correlations with them. The VCD always omits these variables due to Eq. 1 and Eq. 2.

We use the remaining 20 variables to generate all 380 possible variable pairs and measure their association strength. For this, we employ two association metrics for nominal values with arbitrary many categories, (i) the Uncertainty Coefficient U [17] based on conditional entropy, and (ii) the Unbiased Tschuprow's T [2] based on the Pearson Chi-squared statistic. Both metrics are in the range $[0, 1]$, where 0 indicates no association between the variables, and 1 indicates the highest possible dependency. However, while T is symmetrical, U is non-symmetrical and measures how well the dependent variable is predictable by the given variable. For example, the data in Table 1 yields $T(\{\text{syscall}, \text{items}\}) = 0.53$ as well as $U(\text{syscall} \mid \text{items}) = 0.58$ and $U(\text{items} \mid \text{syscall}) = 0.93$.

Results. We run the sequential selection steps of the VCD on the raw event logs and analyze the variable correlations that remain after the initialization phase. In Figs. 2a and 2b, these remaining correlations are marked "yes" (blue triangles), while all variable pairs that are omitted by one of the selection constraints are marked "no" (red circles). Each point in the scatter plots represents one of the 380 variable pairs displayed by their respective U and T, i.e., points closer to the top right corner of the plot indicate stronger association between the two involved variables, while points closer to the bottom left indicate weaker association.

The VCD was used with default settings (cf. Appendix A.1) to classify the variable pairs in Fig. 2a. From all variable pairs, 222 were selected as interesting after the initialization phase and 158 were omitted. Since all of the omitted pairs received a relatively low association score by at least one of the metrics, we conclude that the VCD achieved to correctly omit irrelevant correlations. For example, among the omitted correlations is "syscall" $\rightsquigarrow$ "pid", which is reasonable as the process id "pid" is mostly random and independent from syscall types.

It is possible to further narrow down the set of tracked variable correlations by adjusting the thresholds. In particular, some of the variables involve large numbers of distinct values, which means that the number of monitored value correlations for pairs of these variables is immense. The default value $\theta_1 = 0.3$ allows 3.000 unique values in each variable, which is limited to 100 by setting $\theta_1 = 0.01$. This causes that the number of remaining correlations drops from 222 to 126, where most of the rejected pairs are located close to the top left corner of the plot. Closer examination of these rejected pairs shows that they involve variables with many distinct values on the left side of the correlation and thus achieve a high U score, e.g., syscall arguments such as "a1" $\rightsquigarrow$ "items" with around 1000 unique "a1" values. Since their prediction strengths merely emerge from the large value space, adjusting θ_1 successfully omits these correlations.

In addition to adjusting θ_1, we increase θ_7 from 0.05 to 0.2 in Fig. 2b so that only variable pairs with strong dependency remain. This further reduces the amount of monitored correlations to 97 and omits correlations involving IDs such as "ppid" $\rightsquigarrow$ "exe", while more interesting correlations such as the sample correlation between "syscall" and "items" from Table 1 remains in both directions. We conclude that these experiments show the VCD is capable of selecting useful and strong correlations based on user-defined thresholds.

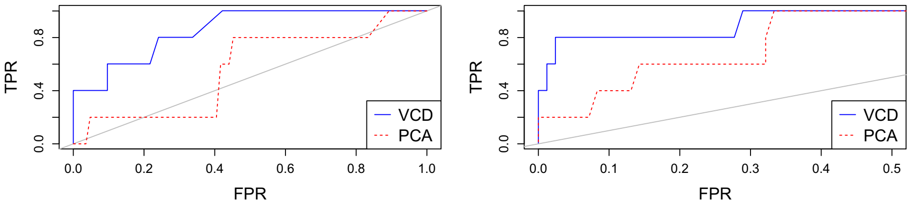
5.2 Anomaly Detection

This part of the evaluation validates the anomaly detection capabilities of the VCD. We first provide information on the log data and then present the results.

Data. We use Apache access logs from the AIT-LDSv1.1 [13] for this part of the evaluation. These logs are relevant, because they involve several categorical variables, including IP addresses, request methods (e.g., "GET", "POST"), resource names, status codes, etc. In addition, web-based attacks frequently manifest themselves as changes of multiple sequential values in these variables. In particular, we select (i) a brute-force login attack using Hydra[1] that repeatedly requests the login web page with arbitrary user data, and (ii) a Nikto vulnerability scan[2] that requests non-available resources and thereby causes multiple redirects that correspond to status code 302. To evaluate detection accuracy with respect to different attack executions, we simulate varying intensities by injecting only a certain amount of events at particular times. Precisely, we inject batches of 5, 10, 20, 50, and 200 events for each attack in intervals of 10000 lines (around 12 h). We label log line samples containing these batches as anomalous to measure the detection accuracy of the VCD in the following.

[1] https://tools.kali.org/password-attacks/hydra, accessed: 2021-04-21.

[2] https://cirt.net/Nikto2, accessed: 2021-04-21.



(a) Hydra brute-force login attack.

(b) Nikto vulnerability scan.

Fig. 3. Anomaly detection ROC plots for two attack scenarios.

Results. For both attack cases, we configure the VCD to use the first $N = 10000$ lines of the Apache Access log files for initialization of the correlations. Thereby, we set $\theta_3 = 0.7$ and $\theta_7 = 0.005$ since the involved variables usually have different distributions and are relatively independent. All other parameters are used with default values (cf. Appendix A.1). After initialization, we use a test sample size of $M = 1000$ to update the correlations on the remaining lines of the first day (20000 lines) using empirically determined confidences $\alpha_1 = 0.001$ and $\alpha_2 = 0.05$, and an indicator list size $d = 30$. This phase omits correlations that appear interesting during initialization, but are too unstable for anomaly detection. With the beginning of the second day, we switch the VCD from updating to testing mode, i.e., correlations that fail tests are no longer changed or omitted. We experiment with different values for α_1 in the test phase and count true positives (TP) as detected samples containing injected lines, false positives (FP) as detected normal samples, false negatives (FN) as undetected samples containing injected lines, and true negatives (TN) as undetected normal samples.

For comparison, we select Principal Component Analysis (PCA) as a baseline, because it allows to handle categorical data through one-hot encoding of values. Similar to the VCD, we use samples of 1000 lines to generate value count vectors and use the first 30000 lines for model building. In the subsequent detection phase, we measure the squared prediction error of test samples and mark them as anomalies if the error exceeds threshold Q_α, where confidence α is varied [10].

Figure 3a shows the trade-off between true positive rate ($TPR = \frac{TP}{TP+FN}$) and false positive rate ($FPR = \frac{FP}{FP+TN}$) of VCD and PCA in the first attack scenario. The results indicate that the VCD successfully detects the attack and yields $TPR = 60\%$ (corresponding to the detection of the samples containing 20, 50, and 200 injected lines) at only $FPR = 10\%$. Closer inspection of the anomalies shows that involved variables are mainly "request" and "referer". In the training phase, the request to the login page "/login.php" occurs in 1.2% of all lines, half of these times with referer "http://mail.insect.com/login.php" and with "–" otherwise. However, requests to the login page made by the Hydra attack always have referer "–" and thus distort this distribution within the test sample, which is detected by the VCD. On the other hand, the PCA ROC curve indicates that it is only slightly better than random guessing. The reason for this is that the one-hot encoded data becomes very high-dimensional and PCA is thus unable to detect slight changes of single values in such complex models.

For the second attack scenario, relevant variables include the request method, where values “GET”, “POST”, and “OPTIONS” occur with 74%, 21%, and 5% in the training data respectively, as well as the status code, where 200 occurs in 96% and 302 in 4% of these lines. The Nikto scan generates lines with request method “GET” and status code “302”, a combination that only occurs in 0.5% of all lines. Since the VCD is better suited to detect changes of occurrences conditioned by infrequent values such as “302” $\rightsquigarrow$ “GET” of correlation “status” $\rightsquigarrow$ “method”, it performs better than PCA as visible in Fig. 3b.

6 Discussion

The evaluation in the previous section ascertains that the VCD selects appropriate variables for correlation analysis and detects anomalies by monitoring co-occurrences of correlated values over time. Thereby, the VCD makes use of a sequence of filtering steps that are separately configured by thresholds. We recognize that such a large number of parameters usually complicates practical application [19], however, we argue that this is not the case for the VCD since the thresholds are set relatively independent and specific to certain properties of the data (cf. Appendix A.1). In addition, it is possible to omit single selection steps and iteratively refine the limits of the search space as we show in Sect. 5.1.

This paper focuses on the correlation between pairs of variables rather than correlations where more than two variables are involved, e.g., $V_1 \rightsquigarrow \{V_2, V_3\}$ or $\{V_1, V_2\} \rightsquigarrow V_3$. However, we argue that this is trivial to achieve, since our selection criteria work analogously with combined occurrences of values. In fact, our implementation [21] supports correlation analysis of specific subsets of variables.

Finally, we suggest to develop selection strategies similar to the one presented in this paper, but with a focus on mixes of categorical and continuous variables, i.e., categorical values indicate that values of another variable origin from a particular continuous distribution. For example, logged measurement data such as memory usage could follow a normal distribution with mean and variance depending on an active user. We leave this task for future research.

7 Conclusion

This paper presents the Variable Correlation Detector (VCD), a novel approach for anomaly detection based on value co-occurrences in categorical variables of log events. The VCD comprises two modes. First, an initialization mode where variable and value correlations are iteratively selected by multiple factors, such as skewness, similarity, and dependency of value occurrence probabilities as well as diversity and coverage of values. Second, an online learning and detection mode that continuously updates the identified correlations and reports anomalies based on deviations of the conditional occurrences. Other than state-of-the-art approaches, the VCD also analyzes infrequent values and recognizes system behavior changes that occur over long time spans. We foresee several extensions for future work, including an anomaly score and automatic threshold selection.

Acknowledgements. This work was partly funded by the FFG projects INDICAETING (868306) and DECEPT (873980), and the EU H2020 project GUARD (833456).

A Appendix

A.1 Threshold Parameter Selection

The filtering steps for correlations between variables and values presented in Sect. 4 make use of threshold parameters θ_1-θ_8 to narrow down the search space and select only those correlations that are likely to positively contribute to the detection of anomalies. This section investigates the influence of these threshold parameters on the resulting correlations and thereby supports the manual parameter selection process, in particular, by relating each parameter to specific properties of the data at hand. In the following, we first explain the generation of synthetic data for this evaluation and then describe our experiments.

Data. To measure the influence of thresholds on the correlation selection, it is necessary to control properties of the input data. Therefore, we generate synthetic data for our experiments. We use three variables V_1, V_2, and V_3, of which only V_1 and V_2 correlate with varying strength, and monitor the correlations found by the VCD for different threshold settings. We use values $\mathcal{V}_i = \{0, 1, ..., x\}, x \in \mathbb{N}$ for each variable and compute their occurrence probabilities as normalized geometric series. Equation 15 shows how the probabilities for values in V_1 and V_3 are computed, where $p_i = 1$ means that all values are equally likely to occur, and lower values mean that one or more values are dominating the probability distribution. Equation 16 shows how the conditional probabilities of values in V_2 given values from V_1 are computed. Thereby, ρ specifies the correlation strength, i.e., larger values for ρ indicate that the same values co-occur more frequently with each other, and ζ is a damping factor that reduces the correlation strength for larger $v_{i,j}$, i.e., higher values for ζ cause more co-occurrences between different values.

$$P\left(v_{i,j}\right) = \frac{p_i^j}{\sum_{j'=0}^{|\mathcal{V}_i|} p_i^{j'}} \tag{15}$$

$$P\left(v_{k,l} \mid v_{i,j}\right) = \frac{(1-\rho)^{|j-l|} + \zeta^{||\mathcal{V}_i|-j|}}{\sum_{l'=0}^{|\mathcal{V}_k|} (1-\rho)^{|j-l'|} + \zeta^{||\mathcal{V}_i|-j|}} \tag{16}$$

Figure 4 shows the co-occurrences of values from V_1 and V_2 for a sample configuration of $x = 9$, $p_1 = 0.7$, $\rho = 0.9$, and $\zeta = 0.4$. Due to the relatively strong correlation factor, most values in V_1 occur with the same value of V_2. The figure also shows that higher values of V_1 co-occur with more values of V_2 due to the damping factor, e.g., while $v_{1,1}$ only occurs with four different values of V_2, $v_{1,9}$ occurs with each value of V_2 at least once.

To evaluate the accuracy of the correlation selection procedure, we generate a ground truth of expected value correlations that contains all $v_{1,j} \rightsquigarrow v_{2,l}$ and

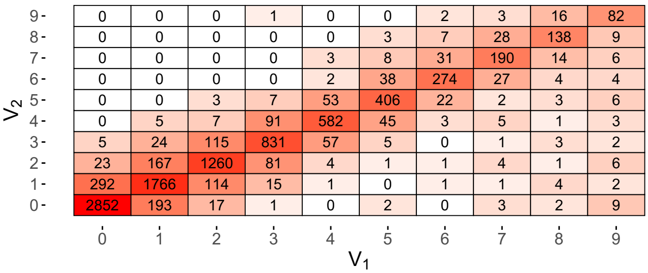
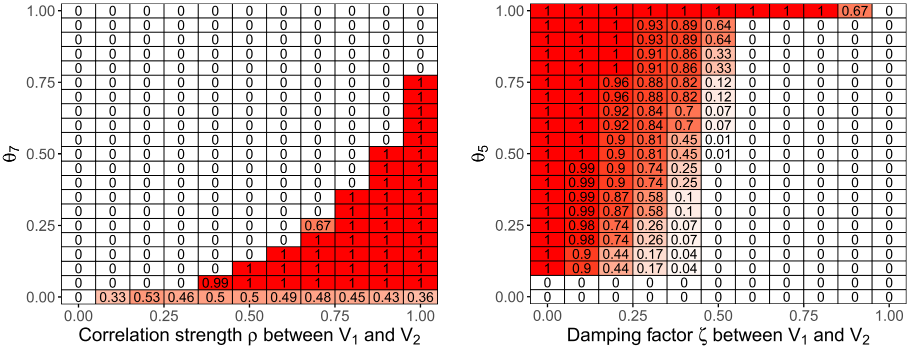


Fig. 4. Value co-occurrences of damped correlation.



(a) Influence of θ_7 for different ρ.

(b) Influence of θ_5 for different ζ.

Fig. 5. Influence of thresholds on accuracy of correlation selection.

$v_{2,l} \rightsquigarrow v_{1,j}$ that occur at least once in the data. We count correlations selected by the VCD and present in the ground truth as true positives (TP), correlations not present in the ground truth as false positives (FP), correlations missed by the VCD as false negatives (FN), and all other correlations as true negatives (TN). We use the F-score $F_1 = TP/(TP + 0.5 \cdot (FP + FN))$ to measure the accuracy in the next section.

Results. We first experiment with θ_7, which is essential for selecting correlations that represent actual dependencies between the values and do not spuriously emerge from skewed value probability distributions. To analyze the relationship between θ_7 and the correlation strength, we increase θ_7 in steps of 0.05 and ρ in steps of 0.1 in the range $[0, 1]$ while leaving $p_1 = 0.7, p_3 = 0.7, \zeta = 0.4$ constant, generate 10 data samples with 10000 events respectively as outlined in the previous section, and then compute the average F-score of these simulation runs. The results visualized in Fig. 5a show that weaker correlation strengths require θ_7 to be sufficiently low to select all correct correlations and achieve the highest possible F-score of 1. However, setting θ_7 to 0 causes a decrease of the F-score independent of the correlation strength. The reason for this is that

Table 4. Dependencies and default values of thresholds.

Thresh.	Infl. by	Default	Thresh.	Infl. by	Default
θ_1	$\lvert\mathcal{V}\rvert, N$	0.3	θ_5	ρ, ζ	0.5
θ_2	p	0.4	θ_6	ρ, ζ	1
θ_3	p, ρ	0.5	θ_7	ρ	0.05
θ_4	$\mathcal{V}$	0	θ_8	θ_5, θ_6	0.7

correlations involving V_3 are not checked for dependency and are thus incorrectly selected, which increases the number of FP. We therefore conclude that θ_7 should be set to a low, but non-zero value, e.g., 0.05. Note that the selection of θ_7 is not affected by ζ, since additional value co-occurrences only have little influence on the sum of variances as long as they are not dominating the distribution.

Threshold θ_5 on the other hand relies on the total number of co-occurrences for a given value and is thus influenced by ζ in addition to ρ. Figure 5b shows the F-score for various combinations of θ_5 and ζ, while $\rho = 1$ is fixed. As expected, increasing values for ζ yield lower F-scores for a given θ_5, because the number of distinct co-occurring values for any given value increases quickly (cf. Fig. 4). Accordingly, it is necessary to set $\theta_5 \geq 1$ for $\zeta > 0.5$ to select any correlations. For $\zeta \leq 0.5$, θ_5 effectively steers the allowed number of distinct co-occurrences, e.g., for $\theta_5 = 0.5$ at most 5 co-occurring values are allowed since $|\mathcal{V}_i| = 10, \forall i$.

We argue that the influence of other thresholds is trivial and therefore omit the plots for brevity. Table 4 shows a summary of all thresholds and the data properties with the highest influence on their selection. Note that θ_8 is most influenced by θ_5 and θ_6 rather than a property of the input data, because these thresholds regulate the generation of value correlations that affect the selection criterion involving θ_8. The table also provides default values that we identified as useful during our experiments and are used in the evaluations in Sect. 5.

These results indicate that the large number of parameters does not impede practical application of the VCD, since the thresholds are mostly independent from each other and allow to configure the correlation selection constraints specifically to counteract otherwise problematic properties of the data. For example, a high number of correlations involving many distinct values (i.e., $|\mathcal{V}|$ is large) or weakly correlated variables (i.e., ρ is low) should be addressed by adjusting θ_1 and θ_7 accordingly to reduce the total number of correlations that are considered for anomaly detection as shown in Sect. 5.1.

References

1. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, vol. 1215, pp. 487–499. Citeseer (1994)
2. Bergsma, W.: A bias-correction for cramér's v and tschuprow's t. J. Kor. Stat. Soc. **42**(3), 323–328 (2013)

3. Bolboacă, S.D., Jäntschi, L., Sestraş, A.F., Sestraş, R.E., Pamfil, D.C.: Pearson-fisher chi-square statistic revisited. Information **2**(3), 528–545 (2011)
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 1–58 (2009)
5. Chen, T., Tang, L.A., Sun, Y., Chen, Z., Zhang, K.: Entity embedding-based anomaly detection for heterogeneous categorical events. arXiv preprint arXiv:1608.07502 (2016)
6. Das, K., Schneider, J.: Detecting anomalous records in categorical datasets. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 220–229 (2007)
7. Djenouri, Y., Belhadi, A., Fournier-Viger, P.: Extracting useful knowledge from event logs: a frequent itemset mining approach. Knowl.-Based Syst. **139**, 132–148 (2018)
8. Eiras-Franco, C., Martinez-Rego, D., Guijarro-Berdinas, B., Alonso-Betanzos, A., Bahamonde, A.: Large scale anomaly detection in mixed numerical and categorical input spaces. Inf. Sci. **487**, 115–127 (2019)
9. Gupta, G.P., Kulariya, M.: A framework for fast and efficient cyber security network intrusion detection using apache spark. Procedia Comput. Sci. **93**, 824–831 (2016)
10. He, S., Zhu, J., He, P., Lyu, M.R.: Experience report: system log analysis for anomaly detection. In: Proceedings of the 27th International Symposium on Software Reliability Engineering, pp. 207–218. IEEE (2016)
11. Ienco, D., Pensa, R.G., Meo, R.: A semisupervised approach to the detection and characterization of outliers in categorical data. IEEE Trans. Neural Netw. Learn. Syst. **28**(5), 1017–1029 (2016)
12. Khalili, A., Sami, A.: Sysdetect: a systematic approach to critical state determination for industrial intrusion detection systems using apriori algorithm. J. Process Control **32**, 154–160 (2015)
13. Landauer, M., Skopik, F., Wurzenberger, M., Hotwagner, W., Rauber, A.: Have it your way: generating customized log datasets with a model-driven simulation testbed. IEEE Trans. Reliab **70**(1), 402–415 (2021)
14. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Inf. Secur. J. Glob. Perspect. **25**(1–3), 18–31 (2016)
15. Narita, K., Kitagawa, H.: Detecting outliers in categorical record databases based on attribute associations. In: Zhang, Y., Yu, G., Bertino, E., Xu, G. (eds.) APWeb 2008. LNCS, vol. 4976, pp. 111–123. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78849-2_13
16. Pande, A., Ahuja, V.: Weac: word embeddings for anomaly classification from event logs. In: Proceedings of the International Conference on Big Data, pp. 1095–1100. IEEE (2017)
17. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes. The Art of Scientific Computing, 3rd edn. Cambridge University Press, Cambridge (2007)
18. Ren, J., Wu, Q., Zhang, J., Hu, C.: Efficient outlier detection algorithm for heterogeneous data streams. In: Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery, vol. 5, pp. 259–264. IEEE (2009)

19. Taha, A., Hadi, A.S.: Anomaly detection methods for categorical data: a review. ACM Comput. Surv. **52**(2), 1–35 (2019)
20. Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., Robinson, S.: Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. arXiv preprint arXiv:1710.00811 (2017)
21. Wurzenberger, M., et al.: Logdata-anomaly-miner. https://github.com/ait-aecid/logdata-anomaly-miner, Accessed 21 Apr 2021