A survey on artificial intelligence techniques for security event correlation: models, challenges, and opportunities

Diana Levshun
St. Petersburg Federal Research Center of the Russian Academy of Sciences
Igor Kotenko (ivkote@comsec.spb.ru)
ITMO University

Research Article

Keywords: Event correlation, Security event, Data mining, Situational awareness, Knowledge representation, Cybersecurity

Posted Date: August 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1975426/v1

A survey on artificial intelligence techniques for security event correlation: models, challenges, and opportunities

Diana Levshun[1] and Igor Kotenko[2*†]

[1]St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg, 199178, Russia.
[2*]ITMO University, St. Petersburg, 197101, Russia.

*Corresponding author(s). E-mail(s): ivkote@comsec.spb.ru;
Contributing authors: gaifulina@comsec.spb.ru;
†These authors contributed equally to this work.

Abstract

Information systems need to process a large amount of event monitoring data. The process of finding the relationships between events is called correlation, which creates a context between independent events and previously collected information in real time and normalizes it for subsequent processing. In cybersecurity, events can determine the steps of attackers and can be analyzed as part of a specific attack strategy. In this survey, we present the systematization of security event correlation models in terms of their representation in AI-based monitoring systems as: rule-based, semantic, graphical and machine learning based-models. We define the main directions of current research in the field of AI-based security event correlation and the methods used for the correlation of both single events and their sequences in attack scenarios. We also describe the prospects for the development of hybrid correlation models. In conclusion, we identify the existing problems in the field and possible ways to overcome them.

Keywords: Event correlation, Security event, Data mining, Situational awareness, Knowledge representation, Cybersecurity

1 Introduction

Modern security tools based on the artificial intelligence (AI) monitor a large number of system events and must identify those that may pose a potential threat or indicate an attack. A security event, also known as an alert, is a message that typically contains information about unusual activity, such as a vulnerability exploitation, or about other aspects of the system security state. An example of an alert generating system is an intrusion detection system (IDS) that detects suspicious activity. An essential element of situational awareness is the event correlation, which makes it possible to identify the relationships between events.

Security event correlation contributes to a better understanding of the attack development, determination of the source and purpose of attack and the identification of the most significant events. In this case, the set of steps of one or more attackers who pursue a specific goal of intrusion is called a multi-step attack or an attack scenario.The steps of the attack are not isolated, but are interconnected by some logical relationships. A number of multi-step attacks are called Advanced Persistent Threats (APTs), which are specifically designed against a single victim and where the attacker's access to the target system is maintained for a long period of time.

In predictive analytics, security event correlation techniques are capable of analyzing both historical data and real-time events, and automatically detect changing thresholds. This allows detecting anomalous events and preventing cyberattacks on early stages. The primary purpose of alert correlation is to identify the most significant events in the security dataset. Also security events correlation has applications in digital forensics when it is necessary to investigate the source of an attack and trigger events.

There is some related surveys devoted to security event correlation approaches (Salah et al, 2013; Mirheidari et al, 2013; Yu Beng et al, 2014). Often, security event correlation techniques match the steps of an attacker as a multi-step or target attack for detection and prediction (Ramaki et al, 2018; Husák et al, 2018; Kovačević et al, 2020; Albasheer et al, 2022). In our last review (Kotenko et al, 2022) , we consider a number of publications on the correlation of security events, including in the field of attack detection.

Most researchers consider approaches to event correlation in terms of implemented methods. As a rule, they distinguish three main groups of event correlation methods: similarity-based, sequence-based and case-based or knowledge-based (Salah et al, 2013; Ramaki et al, 2018). Similarity-based methods analyze the event proximity based on the calculation of a certain similarity measure of event attributes or fields. Sequence-based methods correlate events based on their causal relationship. There are also statistics-base correlation methods (Mirheidari et al, 2013; Albasheer et al, 2022), which can be both a subcategory of methods based on similarity and causal methods. Case- or knowledge-based correlation methods rely on a knowledge base system used to represent well-defined scenarios of events. Sometimes researchers separate data mining as a category of alert correlation techniques (Yu Beng et al, 2014;

Husák et al, 2018; Kovačević et al, 2020). Such methods include a stage of learning from historical events, which creates an intelligent model for further predictions and searches for event patterns.

In our previous work (Kotenko et al, 2022), we present a extended classification of security event correlation approaches based on correlation methods, knowledge extraction, detection type, number of sources, level of analysis, and architecture. We divide correlation methods into three main classes: similarity-based, step-based, and mixed. According to the event knowledge extraction, we separate manual, supervised and unsupervised methods. The type of detection depends on whether the approach examines intrusion detection or anomaly detection. The event data source can be either single or multiple. Depending on the stage, security information can be processed at the raw data, events, and report levels. Architecture of the security event correlation system can be centralized, distributed or hierarchical. In this review, we focus on correlation methods using artificial intelligence. We introduce as a new classification criterion the knowledge representation in intelligent correlation approaches.

The process of representing cause-and-effect relationships of input data in artificial intelligence is closely related to knowledge representation models. As canonical models of knowledge representation, researchers consider rules, logical representation, semantic networks, and frames (Stephan et al, 2007; Tanwar et al, 2010). Knowledge representation in cybersecurity problems can be one of the following types: neural network training weights, rules derived from fuzzy logic, conditional probabilities of Markov models, events from monitoring logs, decision trees, or signature rules (Hamed et al, 2018; Sarker et al, 2021). Models for representing knowledge about security events are most often described in the literature in the form of ontologies (Sikos, 2021) and attack graphs (Zeng et al, 2019; Lallie et al, 2020).

In this survey, we suggest and analyse the taxonomy of security event correlation models based on the ways of event knowledge representation in AI-based monitoring systems. To our best knowledge, our paper is one of the first surveys that focuses on the AI-based security event correlation according to the ways of knowledge representation and usage. Our review contains many recently proposed approaches that were not included in the papers, that have been published in the scientific literature in the last years. At the same time, we include a large number of new scientific publications of recent years that have not been reviewed in our previous work. In our survey, the methods from related fields of event detection are mentioned to demonstrate the existing methods for intelligent analysis of event sequences. As a result, we also outline the prospects of developing combined correlation models.

The paper is organized as follows. Section 2 enumerates and explains the terms and notation used in the paper. Section 3 considers AI-based security event correlation models as rule-based, semantic, graphical, shallow and deep learning, and hybrid. We analyse how artificial intelligence methods are applied for each category and provide examples of using intelligent event correlation models. Section 4 contains a summary and discussion of the models under

consideration. Section 5 offers some promising ways to create combined event correlation models. Finally, section 6 discusses a number of open challenges and considers perspectives for future research that could inspire researchers and developers in this field. Section 7 concludes the paper.

2 Background and notations for security event correlation

First, we define a few important notations for better reading and understanding of the review. Table 1 contains a description of common abbreviations used in the paper. A number of specific abbreviations (names of correlation methods or techniques) are marked with an asterisk (*). We also give explanations of abbreviations directly in the text of the paper. Table 2 contains a description of the main notations and symbols used in the paper.

Table 1 List of abbreviations

Notation	Description	Notation	Description
ABE*	Automaton Based Engine	IoT	Internet of Things
AE	AutoEncoder	IP	Internet Protocol
AI	Artificial Intelligence	IPS	Intrusion Prevention System
ANN	Artificial Neural Network	k-NN	K-Nearest Neighbors
AOI-FIM*	Attribute-Oriented Induction-based Frequent-Item Mining algorithm	KPG	Knowledge Provenance Graph
API	Application Programming Interface	LMS	Log Message Strings
APT	Advanced Persistent Threat	LR	Logistic Regression
ARM	Association Rule Mining	LSTM	Long Short-Term Memory
ASTD*	Algebraic State Transition Diagram	LSWE*	Log-Specific Word Embedding
BERT	Bidirectional Encoder Representations from Transformers	LWE*	Lexical information Word Embedding
BN	Bayesian Network	MAAC*	Multi-step Attack detection by Alert Correlation
CNN	Convolutional Neural Network	MIF*	Multi-Information Fusion system
CVE	Common Vulnerabilities and Exposures	ML	Machine Learning
DL	Deep Learning	MLP	Multilayer Perceptron
DNN	Deep Neural Network	MM	Markov Model
DOMCA*	Detection Of Multi-stage Coordinated Attacks	NLP	Natural Language Processing
DST	Dempster-Schafer Theory	OL	Ontology Learning
DT	Decision Tree	PCA	Principal Component Analysis
ECTBN*	Event-driven Continuous-Time Bayesian Networks	PGM	Probabilistic Graphical Model
EDL*	Event Description Language	RACC*	Real-time Alert Correlation based on Codebooks
ELMo	Embeddings from Language Model	REGNN*	Real-time Event Graph Neural Network
FRM	Frequent Rule Mining	RF	Random Forest
FSA	Finite-state Automata	RNN	Recurrent Neural Network
GA	Genetic Algorithm	RTMA*	Real Time Mining Algorithm
GAN	Generative Adversarial Network	SHEDEL*	Simple Hierarchical Event Description Language
GCN	Graph Convolutional Network	SIEM	Security Information and Event Management
GDN*	Graphical Deviation Network	SIRUS*	Stable and Interpretable RUle Set
GM	Graphical Model	SL	Shallow Learning
GRU	Gated Recurrent Units	SMOTE	Synthetic Minority Over-sampling Technique
HFSN*	Hierarchical Fuzzy Situational Networks	SOAAPR*	Streaming Outlier Analysis and Attack Pattern Recognition
HMM	Hidden Markov Model	STATL*	State Transition Analysis Technique Language
HTTP	Hypertext Transfer Protocol	SVM	Support Vector Machine
IDMEF	Intrusion Detection Messaging Format	SWE*	Semantic Word Embedding
IDS	Intrusion Detection System	ZSEE*	Zero-Shot transfer learning for Event Extraction

Table 2 List of notations and symbols

Symbols	Descriptions	Symbols	Descriptions
E	Event set	Υ^E	Event semanic model
e	Single event	Σ	Semantic model alphabet
$e(t)$	Event with timestamp	δ	State transition function
e^{pre}/e^{sub}	Subset of previous/subsequent events	f_θ	Event embedding algoriphm
F^E	Set of event features	θ	Event embedding space
f	Event feature	$v_(e)$	Event embedding vector
D_k^E	Range of event feature values	$\mathcal{D}(E)$	Event corpus
d	Event feature value	$\mathcal{V}$	Event vocabulary
$\boldsymbol{\alpha}(e)$	Event feature vector	Ω	Event ontology
eS	Event sequence	C_ω	Event concept set
R^E	Event relationship set	D_ω	Event ontology domain
r	Event correlation function	G	Event graph
$Corr$	Event sequence correlation function	w	Graph edge weight function
eS_a	Path of the attacker	$P(e)$	Event probability
eS_c	Current event sequence	$P(R)$	State transition probability
eS_n	Normal event sequence	Q	HMM states
sim	Similarity measure	V	Observation HMM variables per state
$Pr(e)$	Prerequisite set for event	S	Individual HMM states
$Cs(e)$	Consequence set for event	O	Observation sequence of HMM
ES	Variety of event sets	λ	Hidden Markov model
E'	Event vector	π	Initial state distribution
$\sigma(E')$	Support count of event vector	A	State transition matrix
$s(E')$	Support of event vector	B	Confusion matrix
$c(E')$	Confidence of event vector	μ	Event input/output function

Let us define the definitions of event, event type, security event and event correlation, as they are important terms for the area of event correlation.

Definition 1. Event can be individual or cumulative message relating to actions on a system or network that result in a state change. The state change information typically includes a timestamp and a topological label identifying the location of the occurrence.

Definition 2. Event type is a specification for a set of events that have a similar purpose and the same structure.

Definition 3. Security event is a message that contains information about unusual activity, possible threats and vulnerabilities. Such events are usually generated by security systems such as IDS, intrusion prevention system (IPS), antiviruses, firewalls, and others.

Let the set of events, for example in log or capture file, is denoted as:

$$E = \{e_n\},$$

where e is a single event, N is the number of events in the record, $n = 1...N$.

Traditionally, a set of events is viewed as sequences ordered in time:

$$E(t) = \{e(t_i) \mid t_i \leq t_{i+1}.\}$$

Otherwise, for any event $e \in E$, the subset of previous events e^{pre} and the subset of subsequent events e^{sub} are defined as:

$$e^{pre} = \{c \mid (c, e) \in E\},$$
$$e^{sub} = \{c \mid (e, c) \in E\}.$$

The set of events attributes (or features) is denoted as:

$$F^E = \{f_k\},$$

where f is a single feature name, K is a number of features, $k = 1...K$.

Each feature has a range of acceptable values:

$$D_k^E = \{d_{kl}\},$$

where L is the number of possible k feature values, $k = 1...K, l = 1...L$.

The event feature is defined as a mapping:

$$f_k : E \to D_k^E.$$

As a result, each event $e \in E$ can be represented as a feature vector:

$$\boldsymbol{\alpha}(e) = \{(f_1, d_1), ..., (f_k, d_k)\},$$

where the pair (f_i, d_i) corresponds to the i-th feature of the event ($f_i \in F$) with the value $d_i \in D_k^E$.

Definition 4. Event correlation is the creation of context between independent events. The primary purpose of security event correlation is to identify the most significant events in the security dataset.

As mentioned earlier, there are three main categories of correlation methods: similarity-based, step-based, and mixed. Mixed correlation methods use a combination of different correlation algorithms without the obvious predominance of one over the other.

Similarity-based correlation methods compare multiple events based on their attribute similarity. A measure of similarity between attributes can be calculated using Euclidean, Mahalanobis or Manhattan distance functions, correlation coefficients and other mathematical tools. In cybersecurity the basic principle of this correlation type is that a group of similar events can correspond to the same type of attack.

Step-based correlation methods create chains of events, reconstruct a user's actions, and analyze connections between several events. These types of approaches can both match security events based on specific sequence signatures, and define event chains based on their statistical relationships without predetermined knowledge. In cybersecurity, step-based correation methods often use attack scenarios and vulnerability knowledge bases as sources of

knowledge. Step-based correlation methods, in turn, we can divide into causal-based methods and data mining methods. *Cause-based* correlation methods analyze the causal structure of events and obtain a sequence where previous steps determine subsequent ones. *Data mining* correlation methods search for patterns in event datasets using statistical analysis. Note that in this study, we distinguish between the terms "data mining" and "machine learning" not as equivalent. Data mining is a broader field of artificial intelligence research that includes mechanisms such as sequence mining, time series analysis, Bayesian networks, classification, regression analysis and others. So machine learning is one of the areas of data mining.

Let us introduce general notation for the event correlation process. We denote the sequence of events as a pair:

$$eS = (E, R^E),$$

where E is the event set, and R^E is the set of relationships between these events. So, the set R^E can describe several causal conditions for event pairs.

Event correlation can be represented as a mapping:

$$r : e_1 \rightarrow e_2,$$

where the symbol $(\rightarrow)$ denotes a functional relationship between events $e_1 \in E$ and $e_2 \in E$, $r \in R^E$.

In this case, the e_2 is called correlating event for the event e_1 if $e_2 = r(e_1)$. A pair of events can belong both to the same sequence of events $(e_1, e_2 \in E)$ and to different ones $(e_1 \in E_1, e_2 \in E_2)$. A correspondence of events between two subsequences $eS_1 = (E_1, R_1)$ and $eS_2 = (E_2, R_2)$ is a mapping:

$$Corr : eS_1 \rightarrow eS_2.$$

From the point of view of cybersecurity, a sequence of certain events eS_a can be considered as an attack scenario, where each event is an attacker step. In this case, the path of the attacker can be described as:

$$eS_a = \{a_0 ... a_n \mid a_i \in E\}$$

where i=1...n, a_0 is a source of the attack, a_n is a target of the attack, n is a length of the path.

Then matching the current sequence of eS_c events with the sequence of attacking actions eS_a as $Corr : eS_c \rightarrow eS_a$ is important to multi-step attack detection. In this case, we are talking about the so called intrusion or misuse detection. If we only know the sequence of events in normal operation eS_n, the mapping is calculated as $Corr : eS_c \rightarrow eS_n$. This is called anomaly detection.

3 Artificial intelligence models for security event correlation: the state-of-the-art

We will take the canonical methods as the basis for knowledge representation models: rules, logical representation, semantic networks, and frames. Various semantic models can be used as a logical representation of knowledge about security events, and graph models can be used as a semantic networks representation. We also consider frames as a set of features and labels used in machine and deep learning models.

Thus, we highlight the following AI models for event correlation:

- *Rule-based* models (knowledge representation is a set of conditions to compare and aggregate security events);
- *Semantic* models (knowledge representation is some language with specific syntax and semantics);
- *Graphical* models (knowledge representation is in the form of graphical networks);
- *Machine learning* models (knowledge representation is a data structure that consists of a collection of event features and their values).

Figure 1 shows the classification of AI correlation models by event knowledge representation. Further, in this section, we will consider AI-based event correlation approaches, systematizing them by knowledge representation. We also note the existence of *hybrid* models that combine several forms of knowledge representation.

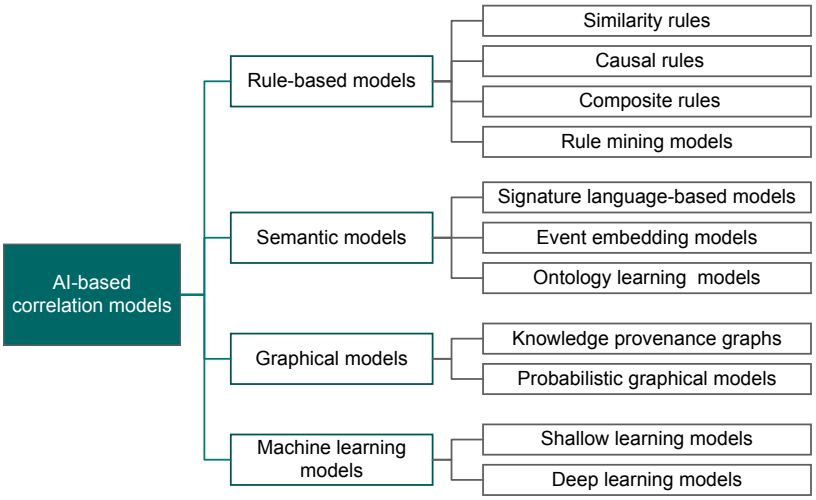

Fig. 1 Classification of AI correlation models by event knowledge representation

3.1 Rule-based models

Rule-based correlation models most often use knowledge about the causal relationships of security events, which are presented in the form of conditional sentences. In this case, event correlation approaches often use a knowledge base that contains rules for matching event attributes.

We distinguish two subcategories of rules: similarity rules and causal rules.

Similarity rules describe the conditions for the similarity of events in terms of their attributes (features), they often use the threshold values of correlation coefficients and similarity measures sim:

$$\begin{gathered} sim(e_i, e_j) = sim(\alpha(e_i), \alpha(e_j)), \\ \text{IF } sim(e_i, e_j) > threshold \text{ THEN } e_i \rightarrow e_j. \end{gathered} \tag{1}$$

A threshold can be set as a fixed value of the correlation coefficient, or calculated from the average value of the feature correlation (Hostiadi et al, 2019). The security event similarity approach can also take into account event attribute weights assigned depending on the attack class (Sun et al, 2020). In addition, similarity can be defined both between event attributes of the same type, and between attributes of different types (Kotenko et al, 2018a, 2020).

Causal rules describe the conditions for the causal relationship of events, an they often use knowledge bases. These rules of event correlation can be described using models of prerequisites and consequences. Those models connect events in such a way that the consequences of early events coincide with the prerequisites of subsequent events. For event e the prerequisite set is denoted as $Pr(e)$ and the consequence set as $Cs(e)$. Then:

$$\text{IF } Pr(e_j) \subseteq Cs(e_i) \text{ THEN } e_i \rightarrow e_j.$$

Let the event be described as a triple $e = (time, type, host)$. Then the conditions for linking events e_1 and e_2 can be as follows (Khosravi and Ladani, 2020):

1. The prerequisites-consequences events occur on the same host: $host_1 = host_2$, $host_1 \in Cs(e_1)$, $host_2 \in Pr(e_2)$;
2. The prerequisite event and the consequence event belong to the same class of events: $type_1 = type_2$, $type_1 \in Cs(e_1)$, $type_2 \in Pr(e_2)$;
3. The prerequisite event precedes the consequence event: $time_1 < time_2$, $time_1 \in Cs(e_1)$, $time_2 \in Pr(e_2)$.

Knowledge of prerequisites and consequences can be also represented in the form of a codebook as binary matrix, where "1" is the presence of a causal relationship between events, and "0" is the absence. In RACC (Mahdavi et al, 2020) (Real-time Alert Correlation based on Codebooks) codebooks correspond to attack scenarios that are mapped to incoming real-time alerts using matrix operations. TempoCode-IoT (Siddiqui and Boukerche, 2021) uses a flow function representation based on unsupervised learning of a temporal

codebook that captures key patterns in data across different time windows. Cluster centers from each time window data are stored as codewords.

The two subcategories of rules listed above can be combined into composite rules. The approach by Tao et al (2021) uses data affinity propagation (AP) clustering, identifying similar alerts, and then a prerequisite and consequence method to recover the full attack process in IoT networks.

Traditionally, security event analysis is provided by default by expert rules, such as rules provided by SIEM systems like Open Source SIEM or Sigma, and programmed rules, for example in IDS Bro. AI methods allow the use of automatic extraction of correlation rules, and reducing the cost of manual specification. Thus, the correlation can be built without requiring predetermined knowledge, so that the system allows finding new correlations between events. In ABE (Lanoe et al, 2018) (Automaton Based Engine) the correlation rules are first represented by a correlation tree based on historical data analysis. In this tree, the nodes are logical nodes (operators AND and OR), and the leafs are the attacker's actions. The correlation tree is then transformed into an automaton that is able to recognize sequences of security events.

An example of rule extraction methods is association rule mining (ARM) based on the frequent rule mining (FRM) paradigm. ARM algorithms allow finding relationships between event attributes in the form of consequence rules. In this case, possible event sets $ES = \{E_1, ..., E_h\}$ are considered, where h is the number of sets. A repeating ordered subset of events, or a vector of events, is denoted as E'. Let this vector have a set $ES' = \{E_k \mid E' \subseteq E_k\}$.

Correlation in this case is a mapping of two consecutive event vectors $Corr : E'_i \rightarrow E'_j$. To assess the correlation, two metrics are introduced – support and confidence. Denote the frequency of occurrence of E' as support count $\sigma(E') =\mid ES' \mid$, where $||$ is the size of the set. The support measure means the proportion of event sets containing the E' subset. Then for vector $E'_i \cup E'_j$:

$$s(E'_i \cup E'_j) = \frac{\sigma(E'_i \cup E'_j)}{h}. \tag{2}$$

Confidence measures how often events of E'_j appear in the vector $E'i$:

$$c(E'_i \cup E'_j) = \frac{\sigma(E'_i \cup E'_j)}{\sigma(E'_i)}. \tag{3}$$

A frequent event vector is the vector whose support and confidence are greater than or equal to the given thresholds $min_{support}$ and min_{conf}:

$$\text{IF } s(E'_i \cup E'_j) \geq min_{support} \text{ AND } c(E'_i \cup E'_j) \geq min_{conf} \text{ THEN } E'_i \rightarrow E'_j.$$

Examples of event rule mining are presented in a number of publications. SIRUS (Bénard et al, 2021) (Stable and Interpretable RUle Set) extracts interpretable rules from the random forest classifier by searching for frequent

patterns in trees. Case-crossover APriori algorithm (Dhaou et al, 2021) provides association and causal rules explaining the occurrences of flooding events. The extraction of association rules can be based on temporal event characteristics, when event segmentation is performed using a sliding window. This analysis is based on the calculation of the frequency characteristics of attributes in accordance with the minimum support threshold (Xie et al, 2018).

3.2 Semantics models

Semantics models use languages with some specific rules, syntax and semantics to establish a relationship between input and output. In this case, events can be represented by sequences of characters that can be considered as "words" of a formal language, specified by some formal grammar.

Denote the semantic model of the event as:

$$\Upsilon^E = (E, \Sigma, \delta), \tag{4}$$

where

1. The event set in a sequence is defined as a set $E = (e_0, E_i, E_f)$ consisting of a sequence, including start event e_0, a set of intermediate state events E_i, and a set of final events E_f;
2. Σ is the model input alphabet (non-empty symbol set);
3. δ is the state transition function $\delta : E \times \Sigma \to E$.

In cybersecurity the signature language allows describing system penetrations as sequences of actions that an attacker performs to compromise. Examples of such languages are STATL (Eckmann et al, 2002) (State Transition Analysis Technique Language) or ASTD (Tidjon et al, 2020) (Algebraic State Transition Diagram), which represent an event sequence in the form of state machines with actions and state variables. Other languages like SHEDEL (Meier et al, 2002) (Simple Hierarchical Event Description Language) and EDL (Jaeger et al, 2015) (Event Description Language) introduce a colored Petri net where nodes are system states, and transitions are current system events.

Transitions between system states in attack scenarios can be also defined using a fuzzy declarative language. Thus, in a fuzzy state machine, the events are represented as fuzzy sets. Transitions from one state to another are described by a fuzzy transition function (Almseidin et al, 2019). Possible input values are determined by the types of attacks in the form of sets of events, and transitions from one state to another are described by fuzzy rules.

For semantic models of security event correlation, researchers often use natural language processing methods. So, by analogy with word embedding, correlation approaches include event embedding. Event embedding is a learned representation for raw event logs as text where the events are words and words with the same meaning have a similar representation. There are research papers on analyzing log events using word embedding methods. Event embedding

can be represented as a parametrized function $f_\theta : E \rightarrow \theta$, where θ is the embedding event space. The embedding algorithm f_θ learns the space θ to preserve the linguistic structure in the reference event "text" corpus $\mathcal{D}(E)$ based on the vocabulary $\mathcal{V}$. The structure in $\mathcal{D}(E)$ is analyzed in terms of the relationship between events caused by their co-occurrence, according to the context definition. So this function maps each event e_i to a vector $v(e_i) \in \mathbb{R}^d$.

The most widely used for word embedding models are Word2Vec (Mikolov et al, 2013b,a) and GloVe (Pennington et al, 2014) both of which are based on unsupervised learning. LogEvent2vec (Wang et al, 2020) and LogUAD (Wang et al, 2022) use Word2Vec to generate word vectors and generates weighted log sequence feature vectors. Doc2Vec (Le and Mikolov, 2014) is similar to the Word2vec algorithm, but instead of vectorizing words, it creates a vector embedding of text snippets. For approach by Liu et al (2020), the training corpus is comprised of rows that are transformed from the raw events of security logs. Each row is treated as a paragraph in training the Doc2Vec model. LogTransfer (Chen et al, 2020b) represents each event log template using Glove, which takes into account both global word matching and local context information. Thus, the presentation of templates minimizes the impact of word order (i.e. syntax) while preserving semantic information. This helps to solve the problem that the log syntax of systems of various types is different, while the log semantics should be reserved.

The big disadvantage of Word2Vec and GloVe is the inability to encode unknown events and attributes. To solve this, the fastText model (Bojanowski et al, 2017), an extension of Word2Vec, splits words into several sub-words (or n-grams), and then transmits them to a neural network. So the improved LogEvent2Vec model by Ryciak et al (2022) uses the fastText algorithm instead of Word2Vec. Ring et al (2021) analyze four different approaches to presenting audit log data: one-hot-ecoding, Word2Vec, fastText, and GloVe. As a result, the study recommends using FastText, which showed the most significant latent space, has the ability to generalize previously unknown values.

Sentence embedding is similar to word embedding, but instead of words, they encode the whole sentence into a vector representation. Some of the most modern models for sentence embedding are ELMo (Peters et al, 2018) and BERT (Devlin et al, 2019). Such models create context-sensitive representations of a word instead of creating one value for each word. ELMo (embeddings from language model) considers the context in which words are used, rather than creating a dictionary of words with its vector form. One of experiments by Zhan and Haddadi (2019) compares how differently the Word2Vec embedding and the ELMo representation perform in prediction network. ELMo representation is used to define each event embedding matrix. The results illustrate that the ELMo representation can better illustrate the contextual-temporal dynamics in event prediction. BERT (Bidirectional Encoder Representations from Transformers) is actively used not only in the processing of natural, but also synthetic languages, such as HTTP/HTTPS

for attack detection in network traffic (Seyyar et al, 2022). BERT models also allow learning the context of event log keys in anomaly detection systems LAnoBERT (Lee et al, 2021) and LogBERT (Guo et al, 2021).

Another common formal language for describing events is the ontology language. We can represent a formal event ontology as:

$$\Omega = (E, C_\omega, F^E, R^E, D_\omega), \tag{5}$$

where E is event instances, C_ω is the concept set (event types, etc.), F^E is the property set, R^E is the relation set, and D_ω is the ontology domain. Ontology learning (OL) allows one to automatically or semi-automatically create ontologies by extracting terms for describing events. During event detection, the OL system attempts to extract complex relationships from the sequence of events by representing them as a natural language text. Examples of such linguistic tools for detecting events and matching their types with target events in the ontology are ZSEE (Huang et al, 2018) (Zero-Shot transfer learning for Event Extraction) and OntoED (Deng et al, 2021). We can say, that event ontology learning aims to get event ontology embedding with the correlation of events, based on the relations among event types.

Within the security framework, the ontology can be based on a hierarchy of concepts that determine the actions of attackers to implement attacks of various classes with varying degrees of detail. Intrusion ontology by Barzegar and Shajari (2018) is based on the Intrusion Detection Messaging Format (IDMEF). IDMEF is a data model for representing information exported by an IDS. An ontology for attack detection can be created using neural networks to learn text embedding as a latent representation of raw security event logs (Zheng et al, 2018). The attack patterns for the ontology by Wang et al (2018) are extracted from the normalized datasets using an Attribute-Oriented Induction-based Frequent-Item Mining algorithm (AOI-FIM). This algorithm includes event aggregation and pattern search using data mining. This correlation approach is based on a machine learning paradigm, such as learning by example, which extracts generalized data and frequently occurring items.

3.3 Graphical models

Graphical models allow one to represent knowledge about events in the form of graphical networks. This network consists of nodes depicting objects and arcs describing those object's relationships. Graphical models (GM) allow one to represent the sequence of events in the form of directed graphs: $G = (E, R^E, w)$, where an event set E is a set of vertices or nodes, relationships R^E is a set of edges $R^E \subset E \times E$, and w is a function mapping edges to their weights, $w : E \to \mathbb{R}$. If the steps of the attacker are considered as events, then such a graph is called an attack graph.

Intelligent methods carry out automated construction of graphs based on event data. NoDoze (Hassan et al, 2019), OmegaLog (Hassan et al, 2020),

UNICORN (Han et al, 2020), HOLMES (Milajerdi et al, 2019), and WATSON (Zeng et al, 2021) analyze the semantic information of the logs and model event knowledge provenance graphs (KPG). NoDoze (Hassan et al, 2019) is based on the understanding that the suspiciousness of each event on the provenance graph must be adjusted based on the suspiciousness of neighboring events on the graph. To assign anomaly scores to events, NoDoze creates an event frequency database and then aggregates the integral anomaly score across neighboring graph events. OmegaLog (Hassan et al, 2020) performs static analysis on log message strings (LMS) and determines their timing relationships, creating a set of all valid LMS control flow paths that may occur at run time. Once the attack is investigated, OmegaLog can use the LMS control flow paths to analyze the flow of events in a cause-and-effect manner.

UNICORN (Han et al, 2020) creates a block graph representing the entire history of system calls and builds a normal evolutionary model of system behavior to detect anomalous actions without knowledge of attacks. HOLMES (Milajerdi et al, 2019) compares tactics, methods and procedures that can be used to perform each stage of APT and creates a high-level graph that summarizes the actions of the attacker in real-time. Host-based intrusion detection system WATSON (Zeng et al, 2021) abstracts behaviors as embeddings (numeric vectors) based on contextual information and provides a vector representation of behavior semantics.

The uncertainty of event values in a graph can be expressed using hierarchical fuzzy situational networks (HFSN) (Kotenko et al, 2019) based on fuzzy inference and multi-agent implementation. This approach allows one to make decisions quickly in dynamic operating conditions.

Graph embedding, or graph representation learning, is a machine learning approach, capable to convert nodes (log entries) in the heterogeneous graph into low-dimension vectors. Approaches like CoRelatE (Huang et al, 2021) study correlations between entities, facts, and relationships from instances in sequences in the form of natural language, and then build knowledge graphs.

Probabilistic graphical model (PGM) is a model in which dependencies between random variables are represented as a graph. In addition to vertices and edges, Bayesian networks (BN) contain a quantitative assessment of relationships based on the conditional probability distributions of each node in the context of its parents. Probability of event e_1, provided that event e_2 has occurred (posterior probability) as:

$$P(e_2 \mid e_1) = \frac{P(e_1 \mid e_2)P(e_2)}{P(e_1)} \tag{6}$$

where $P(e_1 \mid e_2)$ is the probability of e_1, provided that e_2 has occurred, $P(e_1)$ and $P(e_2)$ are the probabilities of e_1 and e_2.

Then the probability of a certain event sequence is:

$$P(e_1 ... e_n) = \prod_{i=1}^{n} p(e_i \mid e_{pa(i)}), \tag{7}$$

where $e_{pa(i)}$ are parents of node i. This model can be used to calculate the probability of a certain security violation or an attacker's action (Kim et al, 2020).

The Bayesian graph can reflect possible attack paths with the probability of transition between the attacker's steps. So at any moment, the attacker is in node e_i and moves on to the next node e_j only if a certain vulnerability exists and can be exploited. The probability of this state transition is given by:

$$P(R_{ij}) = P^V(R_{ij}) \cdot P^V_{exp}(R_{ij}), \tag{8}$$

where R_{ij} is state transition between events e_i and e_j, P^V is the probability presented by the vulnerability j, and P^V_{exp} is the probability that such a vulnerability is exploitable if present. Algorithms for finding the paths of an attacker with the highest transition probability allow us to detect possible multi-step attacks, including APT attacks (Zimba et al, 2019) and zero-day attacks (Sun et al, 2018).

To reflect the impact of events on state variables, ECTBN (Bhattacharjya et al, 2020) (Event-driven Continuous-Time Bayesian Networks) can be used, in which, in addition to state variables, a history of events with a timestamp can affect the time and probability of transition of state variables. DOMCA (Sen et al, 2022) (Detection Of Multi-stage Coordinated Attacks) presents attack scenarios using the Dempster-Schafer Theory (DST) (Dempster, 2008). DST is a generalization of traditional Bayesian probability that allows you to assign probabilities to sets of statements. This allows you to combine events from several sources without a priori knowledge, i.e. prior probability distributions about the states of the system.

The Markov model (MM) or chain is similar to the Bayesian network in terms of dependencies. The difference is that Bayesian networks are directional and acyclic, whereas Markov chain are undirected and can be circular. Approaches such as RTMA (Zhang et al, 2019) (Real Time Mining Algorithm) and Third Eye (Hossain and Xie, 2020) monitor how well the observed sequence of events corresponds to the established model of normal or malicious behavior. For multi-step attack scenario reconstruction the RTMA (Zhang et al, 2019) uses the concept of MM to facilitate alerts analysis. In this case, the correlation is carried out between event types and event attributes. The Markov state in Third Eye (Hossain and Xie, 2020) denotes the state of the node under test (NUT) for current operating IoT channel at the end of a time-slot. The state transition diagram of the proposed Markov model depicts the interaction between the primary user, the NUT, and the external node (an external terminal used by an attacker).

In a hidden Markov model (HMM), states are not observable, and we can only keep track of the variables (or symbols) that are affected by the state:

$Q = \{q_1, q_2, ..., q_n\}$ – set of hidden Markov process states,
$V = \{v_1, v_2, ..., v_m\}$ – set of observation variables per state,
$S = \{s_1, s_2, ..., s_n\}$ – set of individual states,

$O = \{o_1, o_2, ..., o_t\}$ – state observation sequence,

where n is a number of model states, m is a number of distinct observation variables per state, t is an observation sequence length.

HMM model is a triplet:

$$\lambda = (\pi, A, B), \tag{9}$$

where

1. π is an initial state distribution, $\pi = \{\pi_i\}$, where $\pi_i = P(q_1 = s_i)$ are the probabilities that s_i is the first state in the sequence of states;
2. $A = \{a_{ij}\}$ is a state transition matrix, where $a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$ is the probability of transition from state s_i to state s_j at time t;
3. $B = \{b_{ik}\}$ is a confusion matrix, where t $b_i(k) = P(o_t = v_j \mid q_t = s_i)$ is the probability of observing state s_k given $q_t = s_i$ at time t.

For a given model, the observation probability of O is determined as:

$$P(O \mid \lambda) = \sum_{Q} P(O \mid Q, \lambda) P(Q \mid \lambda). \tag{10}$$

An important step in model training is to tune the model parameters to maximize the probability of observation. Parameter optimization can be achieved using the Baum-Welch algorithm (Welch, 2003), as well as others. In (Khan and Abuhasel, 2021) a hidden Markov model is explored to model serial data that is generated by IoT devices. To optimize the HMM parameters, a genetic algorithm is used that maps the search space to the genetic space. Each gene has a mean and variance for each state in the HMM.

Often, hidden Markov models are trained for each event sequence type. So for cybersecurity the HMM is created for each type of attack, for example using a training set of alerts generated by an IDS (Zegeye et al, 2018; Shawly et al, 2019) or based on common vulnerabilities and impacts (CVE) (Holgado et al, 2017; Ma et al, 2022). In the first case, the HMM parameters are extracted from the IDS alert training dataset, and can be adjusted online. In the second case, the possible observations (V) are based on different tags in the CVE repository and the severity of the alerts. For a multi-stage attack pattern k, the HMM includes the number of attack stages (Markov chain states), the number of associated observations, and the above probability matrices A and B. Subsequently, given observations associated with attack k, the HMM estimates the probability of being in each state of the model using Viterbi algorithm (Viterbi, 1967).

3.4 Machine learning models

Machine learning models, such as shallow and deep, use frames as a representation of data. A frame is the AI data structure that includes a collection of attributes and values. It consists of a collection of slots and slot values of

any type and size. This structure allows one to use large amounts of knowledge about events and analyze them using intelligent methods such as cluster analysis and machine learning.

If there is a knowledge base about all normal (eS_n) and abnormal (eS_a) events and their attributes, the problem of event detection and prediction of event sequences (eS_{out}) can be reduced to the problem of multiclass classification in the form of some function (μ) that matches input values with output values: $eS_{out} = \mu(eS_a \cup eS_n)$. The detection of anomalies in the event sequence is considered as an unsupervised or semi-supervised learning task $eS_{out} = \mu(eS_n)$. In both cases, the following main stages of solving the problem are distinguished: (1) identification of informative event features; (2) selection and training of a model (algorithm) capable of assigning the current event sequence to a certain class; (3) calculation of reliability and accuracy. As a rule, all three steps are repeated iteratively until a set of features and a model are found that meet the specified reliability and accuracy criteria.

Shallow learning models are traditional machine learning techniques which can be used for alert correlation by mapping alert features, such as alert attribute values, event rates, etc. The number of alerts per day, the frequency of event occurrence, relational functions obtained from the social graph analysis are used as features for training. Chang and Wang (2016) profile malware data to extract attack scenarios using k nearest neighbor (k-NN), decision tree (DT) and support vector machine (SVM) algorithms. For Big Data monitoring in IoT system, the approach by Kotenko et al (2018b) uses a structure involving principal component analysis (PCA), DT, SVM, k-NN, the gaussian naïve Bayes (GNB) and the artificial neural network (ANN). The SMOTE-RF model (Li et al, 2021) combines SMOTE and random forest (RF) algorithms to solve the problem of unbalanced classification and multiclassification in APT datasets. The SMOTE increases the number of minority samples through k nearest neighbor interpolation to improve the distribution of an imbalanced dataset. Then, multiclass learning is performed based on the RF.

Recurrent neural network (RNN) allows the analysis of sequential data such as time series or natural language texts. In the latter case, tools like SAM-Net (Lv et al, 2019) are common, which model relationships between events in a text corpus and represent them as scripts. Approaches such as Tiresias (Shen et al, 2018), DeepLog (Du et al, 2017), OC4Seq (Wang et al, 2021b) use RNN to predict future events based on previous observations to track anomaly behavior. Tiresias (Shen et al, 2018) calculates a probability distribution of possible events $e^{pre} = \{e_{k+1}...e_n\}$ given historical observed events $e^{sub} = \{e_1...e_k\}$, where k refers to the rollback window size, to predict the specific steps that will be taken by an adversary when performing an attack. Similarly, OC4Seq (Wang et al, 2021b) uses Gated Recurrent Units (GRU) for anomaly detection in event sequences. DeepLog (Du et al, 2017) uses Long Short-Term Memory (LSTM) and learns the correlations and patterns embedded in a sequence of log entries produced by normal system execution paths.

Convolutional neural network (CNN) is often used to process arrays of input data to look for specific patterns. For example, Chen et al (2020a) encode the system call sequence into a two-dimensional fixed-length "picture", which is very suitable for CNN analysis. For this, N-Gram is used, the main idea of which is to cut the sliding window and get the sequence segments of length *N*. CNN is used in DeepCorr (Nasr et al, 2018) to study the stream correlation function adapted to the complex Tor network.

Autoencoder (AE) is a symmetric neural network and usually studies the features of events in an unsupervised manner. Abdullayeva (2021) uses a deep autoencoder model to automatically extract event features and encode APT attack vectors. Network Anomaly Detection (Min et al, 2021) with MemAE (Memory-augmented deep Auto-Encoder) solves the problem of over-generalization when normal sampling and attack sampling have common features. So APT signature templates have problems with low generalization performance. MemAE approximates the attack input reconstruction to a normal pattern by using a memory module.

Generative Adversarial Network (GAN) uses an adversarial mechanism to extract implicit relationships between events. The teacher network by Liu et al (2019b) encodes event data into vectorized knowledge representations for feature learning. The student network processes raw texts for event detection and requires no extra toolkits, naturally eliminating the error propagation problem faced by pipeline approaches.

Model ensembles combine several different models of learning, including both supervised and unsupervised, or shallow and deep. The approach by Oki et al (2018) uses an ensemble of RF, logistic regression (LR) and AE models to detect and predict mobile network outages using multiple sets of user activity data. Ghafouri et al (2018) describe an ensemble predictor that contains a deep neural network (DNN) and a linear regression model to detect anomalous cyber-physical sensor readings, where each sensor's measurement is predicted as a function of other sensors. Model combination studies often involve evaluating and determining the optimal combinations of AI models and their parameters to most effectively achieve results. Joloudari et al (2020) use three AI-based classification models to early detect and classify APT attacks, including Bayesian network, C5.0 decision tree, and multilayer perceptron (MLP). Also, a combination of CNN and LSTM models is often used to detect and predict APT (Cheng et al, 2019; Do Xuan and Dao, 2021). The advantage of the CNN-LSTM model in security event analysis is that such an architecture works well with tasks where the raw data has an explicit structure and has temporal properties.

3.5 Hybrid models

Some approaches and systems combine several event correlation methods, without the obvious predominance of one over the other. As a rule, systems that use a combination of similarity-based and casual-based correlation assume

that the most similar events may be associated with the same attack scenario. Multi-step attack scenario reconstruction consists of three main aspects: (1) identifying related security events, (2) matching a subset to the appropriate scenario, and (3) ordering the sequence of events.

Correlation can be based on the similarity of events by parameters (for example, source and destination IP addresses and ports), and the scenario can be represented as a graph (Haas and Fischer, 2019; Bajtoš et al, 2020). So SOAAPR (Heigl et al, 2021) (Streaming Outlier Analysis and Attack Pattern Recognition) matches and groups alerts in streaming mode, and the resulting clusters are converted into a graphical representation. The result is an attack signature that represents the attack scenario in terms of communication behavior, cause in data features, and time sequence of associated alerts.

Another way of correlation is the analysis of semantically similar events. MAAC (Wang et al, 2021a) (Multi-step Attack detection by Alert Correlation) uses Doc2vec to get the semantic representation of the alert description and calculates the cosine distance of the generated vector. MAAC matches the alerts and creates a graph first for alerts generated on the same host and then between hosts. The approach by Zhang et al (2022) uses the Word2Vec model to convert alerts to low-dimensional continuous values and match semantically similar alerts. Then the distance of the alert vector to each attack stage is converted into the probability of generating alerts at each attack stage, replacing the initial Baum-Welch value, to build the attack HMM. The Log2Vec method (Liu et al, 2019a) uses Log-Specific Word Embedding (LSWE) Word2Vec for word representation that enhances domain-specific semantic and relational information. LSWE uses two methods of word embedding: Lexical information Word Embedding (LWE) and Semantic Word Embedding (SWE). LWE predicts the target word so that its vector representation distance is as close as possible to its synonyms and as far as possible from its antonyms. SWE defines associative word relationships.

The extraction of security event attributes can also be performed using machine learning methods. MIF (Mao et al, 2021) (Multi-Information Fusion system) extracts anomalous alert streams using a CNN called Convolution and agent decision Tree network (CTnet) and then reconstructs the attack scenario using graph-based fusion module. CTnet evaluates attack risks, which, together with information about attack nodes and attack time, are used to build a graph-based fusion module. The high-risk attack chain is retrieved using a Time-Weighted Depth-First Search (TW-DFS) algorithm. Weight information determines the path through nodes with a higher risk of attack, and time information helps to remove non-temporal correlations of attacks.

Recently, AI-based methods are often used to analyze event graphs. REGNN (Luo et al, 2020) (Real-time Event Graph Neural Network) is used to embed and predict real-time events by building dynamic heterogeneous graphs. This model creates event provenance graphs for user behavior and then uses recurrent neural networks to model the time dependence of past events and embed real-time events. In turn, CNN can be applied to graph convolutional

network (GCN) (Nguyen and Grishman, 2018), in which the convolution vector for each node is computed from nearest neighbor representation vectors. To detect and predict events, the GNN uses the current event vector in the graph. The GDN (Deng and Hooi, 2021) (Graphical Deviation Network based approach) examines the graph of relationships between sensors and detects deviations from these patterns. This approach makes it possible to detect anomalies without preliminary data on the structure of graphs.

4 Summary of AI-based security event correlation models

The main part of the described security event correlation approaches has a common goal: to detect and predict security breaches that are step-by-step in nature – multi-step or targeted attacks or cause-and-effect violations of the system stability. In this section, we provide a summary of the review results. We classify the considered correlation approaches according to the following main criteria: application, AI-model used, and correlation method.

We can select the following main directions for **application** of AI methods to correlate security events:

- Clustering of similar events to reduce the volume of processed information and classify security events to event detection (ED) (Liu et al, 2019b; Deng and Hooi, 2021), event grouping (EG) (Hostiadi et al, 2019; Sun et al, 2020), and event pattern extraction (EPE) (Dhaou et al, 2021; Zeng et al, 2021).
- Intrusion detection (ID), which deals with multi-stage and targeted attacks (Joloudari et al, 2020; Sen et al, 2022), or anomaly detection (AD) (Han et al, 2020; Wang et al, 2022) to notify the security administrator about misuses and deviations from normal behavior, respectively.
- Intrusion prediction (IP) (Holgado et al, 2017; Oki et al, 2018) based on incoming events, which allows early detection of intruder targets.

As the main event correlation **AI-models**, we can note the following:

- Rule-based correlation models – similarity rules (SimR) (Kotenko et al, 2020), causal rules (CauR) (Mahdavi et al, 2020; Siddiqui and Boukerche, 2021), composite rules (ComR) (Tao et al, 2021) and rule mining models (RM) (Xie et al, 2018; Bénard et al, 2021).
- Semantic correlation models – signature language-based (SigL) (Almseidin et al, 2019; Tidjon et al, 2020), event embedding (EE) (Lee et al, 2021; Seyyar et al, 2022), and ontology learning (OL) (Zheng et al, 2018; Deng and Hooi, 2021) models.
- Graphical correlation models – knowledge provenance graphs (KPG) (Milajerdi et al, 2019; Zeng et al, 2021), and probabilistic graphical models (PGM) (Shawly et al, 2019; Ma et al, 2022).
- Machine learning correlation models – shallow (SL) (Kotenko et al, 2018b; Li et al, 2021) and deep learning (DL) (Du et al, 2017; Min et al, 2021) models.

We will also highlight three main areas of event **correlation methods**:

- Similarity-based (SB) methods are based on the idea that similar events can have the same root cause or the same type, and the found links depend on the inherent similarity between attributes of each event (Kotenko et al, 2018a; Heigl et al, 2021).
- Causal-based (CB) methods focus on the causal structure of a event sequence, when previous steps determine the ones that follow (Zegeye et al, 2018; Hossain and Xie, 2020).
- Data mining (DM) is a process of discovering significant patterns especially in a large amount of data (Abdullayeva, 2021; Zhang et al, 2022).

Tables 3 to 7 provide an overview of the considered security event correlation approaches. For each approach (the name of the approach, if any, is indicated), we also define the main application area (App.), the AI-model (Model) and their basis (Basis), the correlation method (Corr.), the type of data under study (Data type), and used dataset (Dataset).

We can distinguish the following main types of data for security event correlation: event logs, IDS alerts, network traffic, vulnerability databases (CVE), and malware. We designate the datasets created by researchers and not made publicly available as Generated. The following open datasets are used as in the reviewed studies:

- *as system logs*: BlueGene/L Supercomputer System (BGL) and Thunderbird (Oliner and Stearley, 2007), Hadoop Distributed File System (HDFS) (Xu et al, 2009), Rice University Bidding System (RUBiS) (Amza et al, 2002), CERT Insider Threat (Glasser and Lindauer, 2013), DARPA TRACE (DARPA TC, 2020), Los Alamos National Laboratory (LANL) (Kent, 2016), Secure Water Treatment (SWaT) (Goh et al, 2016) and UCI datasets (Asuncion and Newman, 2007);
- *as log text annotated corpora*: ACE 2005 (Walker et al, 2006), FB15K (Bollacker et al, 2008), WN18 (Miller, 1995), JF17K (Wen et al, 2016), WikiPeople (Guan et al, 2019);
- *as IDS alert datasets*: DARPA 1998 (LL-MIT, 1998), DARPA 2000 (LL-MIT, 2000), CIC-IDS2017 and CIC-IDS2018 (Sharafaldin et al, 2018), N-BaIoT (Meidan et al, 2018), CTU-13 (Garcia et al, 2014), Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC) 2012 (NETRESEC, 2000), Vast Challenge 2012 (Cook et al, 2012), ADFA-LD (Creech and Hu, 2013), ISCXIDS 2012 (Shiravi et al, 2012), NDSec (Beer and Bühler, 2017) and DEFCON 21 (DEF CON Communications, Inc., 2021);
- *as network traffic*: CSIC 2010 (Giménez et al, 2010), FWAF (Ahmad, 2017), HttpParams (Morzeux, 2020), NSL-KDD (Tavallaee et al, 2009), UNSW-NB15 (Moustafa and Slay, 2015), 4ICS Geek Lounge (NETRESEC, 2015);
- *as malware*: ContagioDump (Contagio Mobile, 2011) and MalwareTrainingSets (Ramilli, 2016).

Table 3 Overview of rule-based security event correlation

Ref.	Approach	App.	Model	Basis	Corr.	Data type	Dataset
Kotenko et al (2018a), Kotenko et al (2020)	N/A	EPE	SimR	Correlation coefficients	SB	Logs	Generated
Hostiadi et al (2019)	N/A	EG	SimR	Correlation coefficients	SB	IDS alerts	Generated
Sun et al (2020)	N/A	EG	SimR	Rough entropy	SB	IDS alerts	DARPA 1998, CICIDS 2018
Khosravi and Ladani (2020)	N/A	EPE, ID	CauR	Prerequisites-consequences	CB	IDS alerts	Generated
Mahdavi et al (2020)	RACC	ID	CauR	Codebook	CB	IDS alerts	DARPA 2000
Siddiqui and Boukerche (2021)	TempoCode-IoT	ID	CauR	Codebook	CB	IDS alerts	CICIDS 2017, NBaIoT
Tao et al (2021)	N/A	ID	ComR	AP clustering	DM	Net. traffic	Generated
Lanoe et al (2018)	ABE	ID, IP	RM	Correlation tree	DM	Net. traffic	Generated
Xie et al (2018)	N/A	EPE	RM	Temporal association tree	DM	Logs	Generated
Dhaou et al (2021)	CAP	EPE	RM	Frequent rule mining	DM	Logs	Generated
Bénard et al (2021)	SIRUS	EPE	RM	Random forest	DM	Logs	UCI datasets

Table 4 Overview of semantic model-based security event correlation

Ref.	Approach	App.	Model	Basis	Corr.	Data type	Dataset
Jaeger et al (2015)	EDL	ID	SigL	Colored Petri net	DM	Logs	Generated
Tidjon et al (2020)	ASTD	ID	SigL	FSA	DM	IDS alerts	CICIDS 2018, CTU-13
Almseidin et al (2019)	N/A	ID	SigL	Fuzzy logic	CB	IDS alerts	DARPA 2000
Zhan and Haddadi (2019)	N/A	IP	EE	ELMo	DM	Logs	Generated
Wang et al (2020)	LogEvent2vec	AD	EE	Word2Vec	DM	Logs	BGL
Liu et al (2020)	N/A	ID	EE	Doc2Vec	DM	Logs	CERT Insider Threat
Chen et al (2020b)	LogTransfer	AD	EE	Glove	DM	Logs	Generated
Wang et al (2022)	LogUAD	AD	EE	Word2Vec	DM	Logs	BGL
Lee et al (2021)	LAnoBERT	AD	EE	BERT	DM	Logs	BGL , HDFS
Guo et al (2021)	LogBERT	AD	EE	BERT	DM	Logs	BGL, HDFS, Thunderbird
Ryciak et al (2022)	N/A	AD	EE	fastText	DM	Logs	BGL
Seyyar et al (2022)	N/A	ID	EE	BERT	DM	Net. traffic	CSIC 2010, FWAF, HttpParams
Zheng et al (2018)	N/A	ID	OL	ANN	DM	Logs	Generated
Barzegar and Shajari (2018)	N/A	ID	OL	SimRank	DM	IDS alerts	DARPA 2000, MACCDC 2012
Huang et al (2018)	ZSEE	ED	OL	Abstract Meaning Representation	DM	Logs	ACE05 data
Wang et al (2018)	N/A	ID	OL	AOI-FIM	DM	IDS alerts	DARPA 2000, Vast Challenge 2012
Deng et al (2021)	OntoED	ED	OL	NLP	DM	Logs	Generated

For rule-based correlation models, we can see a clear correspondence between rule type and correlation method. Semantic correlation models and machine learning models use data mining, while graphical models explore causal correlation. Event grouping and pattern searching are characteristic of rule-based and graphical models, while intrusion and anomaly detection and prediction are characteristic of graphical and machine learning models.

Table 5 Overview of graphical model-based security event correlation

Ref.	Approach	App.	Model	Basis	Corr.	Data type	Dataset
Milajerdi et al (2019)	HOLMES	ID	KPG	GM	CB	Logs	DARPA TRACE
Hassan et al (2019)	NoDoze	EPE	KPG	GM	CB	Logs	Generated
Hassan et al (2020)	OmegaLog	EPE	KPG	GM	CB	Logs	Generated
Han et al (2020)	UNICORN	AD	KPG	GM	CB	IDS alerts	Generated
Zeng et al (2021)	WATSON	EPE	KPG	GM	CB	Logs	DARPA TRACE
Huang et al (2021)	CoRelatE	EPE	KPG	Graph embedding	CB	Logs	FB15K, WN18, JF17K, WikiPeople
Holgado et al (2017)	N/A	IP	PGM	HMM	CB	IDS alerts, CVE	DARPA 2000
Zegeye et al (2018)	N/A	ID	PGM	HMM	CB	IDS alerts	CIC-IDS2017
Sun et al (2018)	ZePro	EPE	PGM	BN	CB	Logs, CVE	Generated
Zimba et al (2019)	N/A	EPE	PGM	BN	CB	IDS alerts, CVE	Generated
Shawly et al (2019)	N/A	ID	PGM	HMM	CB	IDS alerts	DARPA 2000
Zhang et al (2019)	RTMA	EPE	PGM	MM	CB	IDS alerts	DARPA 2000
Kim et al (2020)	N/A	EPE	PGM	BN	CB	IDS alerts	Generated
Bhattacharjya et al (2020)	ECTBN	EPE	PGM	BN	CB	Logs	Generated
Hossain and Xie (2020)	Third Eye	ID	PGM	MM	CB	Net. traffic	Generated
Khan and Abuhasel (2021)	N/A	EPE	PGM	HMM, GA	CB	Net. traffic	Generated
Sen et al (2022)	DOMCA	ID	PGM	BN	CB	IDS alerts	Generated
Ma et al (2022)	N/A	EPE	PGM	HMM	CB	IDS alerts, CVE	Generated

Table 6 Overview of machine learning-based security event correlation

Ref.	Approach	App.	Model	Basis	Corr.	Data type	Dataset
Chang and Wang (2016)	N/A	ID	SL	DT, k-NN, SVM	DM	Malware	ContagioDump
Kotenko et al (2018b)	N/A	ID	SL	DT, SVM, ANN	DM	IDS alerts	N-BaIoT
Li et al (2021)	N/A	ID	SL	SMOTE-RF	DM	Malware	Generated
Du et al (2017)	DeepLog	AD	DL	LSTM	DM	Logs	HDFS
Shen et al (2018)	Tiresias	IP	DL	RNN	DM	Logs	Generated
Wang et al (2021b)	OC4Seq	AD	DL	GRU	DM	Logs	HDFS, RUBiS, BGL
Nasr et al (2018)	DeepCorr	EPE	DL	CNN	DM	Net. traffic	Generated
Chen et al (2020a)	N/A	AD	DL	CNN	DM	IDS alerts	ADFA-LD
Abdullayeva (2021)	N/A	IP, AD	DL	AE	DM	Malware	MalwareTrainingSets
Min et al (2021)	N/A	AD	DL	AE	DM	Net. traffic, IDS alerts	NSL-KDD, UNSW-NB15, CIC-IDS2017
Liu et al (2019b)	N/A	ED	DL	GAN	DM	Logs	ACE 2005
Cheng et al (2019)	N/A	IP	DL	CNN-LSTM	DM	Net. traffic	Generated
Do Xuan and Dao (2021)	N/A	ID	DL	CNN-LSTM	DM	Net. traffic	CTU-13
Oki et al (2018)	N/A	ID, IP	SL, DL	RF, LR, AE	DM	Logs	Generated
Ghafouri et al (2018)	N/A	AD	SL, DL	LR-DNN	DM	Logs	Generated
Joloudari et al (2020)	N/A	ID	SL, DL	BN, DT, MLP	DM	Net. traffic	NSL-KDD

5 Prospects for combining AI-based security event correlation models

Recent trends in security event correlation are leading to an increasing use of hybrid AI-models (Haas and Fischer, 2019; Bajtoš et al, 2020; Deng and Hooi, 2021). Such models allow researchers to use the advantages of various correlation methods and offset the disadvantages.

Similarity-based methods have the simplicity of implementation to determine the relationship between a pair of events. However, the difficulty consists in choosing the most efficient way to calculate the event connection (Hostiadi et al, 2019; Bajtoš et al, 2020). Simple matching of event attributes can give

Table 7 Overview of hybrid model-based security event correlation

Ref.	Approach	App.	Model	Basis	Corr.	Data type	Dataset
Haas and Fischer (2019)	GAC	ID	SimR, KPG	Clustering, GM	SB, CB	IDS alerts	Generated
Bajtoš et al (2020)	N/A	ID	SimR, KPG	Corr. Coeff., GM	SB, CB	Net. traffic	4ICS Geek Lounge
Heigl et al (2021)	SOAAPR	EPE, ID	CauR, KPG	GM	SB, CB	IDS alerts	CIC-IDS2017, CIC-IDS2018
Liu et al (2019a)	Log2Vec	ID	EE, KPG	Word2Vec, GM	DM	Logs	CERT Insider Threat, LANL
Wang et al (2021a)	MAAC	ID	EE, KPG	Doc2Vec, GM	CB, DM	IDS alerts	DARPA 2000, ISCXIDS 2012, NDSec
Zhang et al (2022)	N/A	ID	EE, PGM	Word2Vec, HMM	CB, DM	IDS alerts	DARPA 2000, DEFCON 21, ISCXIDS 2012
Mao et al (2021)	MIF	EPE	DL, KPG	CNN, GM	DM, CB	IDS alerts	DARPA 2000
Luo et al (2020)	REGNN	ED	KPG, DL	GM, RNN	CB, DM	Logs	Generated
Nguyen and Grishman (2018)	N/A	ED	KPG, DL	GM, CNN	CB, DM	Logs	ACE 2005
Deng and Hooi (2021)	GDN	AD	KPG, ML	GNN	CB, DM	Logs	SWaT

a lot of false positives. At the same time, complex functions can be too specific, which makes similarity-based methods less flexible and poorly adapted for detecting a wide range of events.

Causal-based correlation methods make it easy to interpret the results of correlation by the operator. Therefore, this category of methods is well suited for visualizing the sequence of events (Heigl et al, 2021; Wang et al, 2021a). The simplicity of implementing such models is reduced, and also requires more computing resources to process a large amount of data.

In turn, data mining correlation methods are easier to deal with. Data mining methods allow one to automatically extract event features for correlation (Nguyen and Grishman, 2018; Liu et al, 2019a). The interpretability of the results, in turn, decreases. In addition, the performance of trained models strongly depends on the training dataset.

Based on the results of the review, we can describe the functional requirements for the most complete event correlation model as follows:

- the model should be able to define sequences or clusters of related events;
- the model should take into account the correlation between the features of events;
- the model should take into account the semantic properties of events;
- the model should be able to automatically process a large amount of event data;
- the model should be able to visually interpret the results of event correlation.

The figure 2 shows the proposed diagram of the application of the hybrid model for the correlation of security events. This model meets all the requirements by combining correlation methods with the corresponding functions. The hybrid model accepts a stream of security events as input. The output of the model is the reconstructed security event scenario (attack graph, anomalous sequence, etc.), as well as the type of scenario and the security events predicted according to the scenario. The semantic model transforms the input data into context-aware security event vectors. The rule-based model demonstrates the proximity of vectors of similar events in a clear way, and then

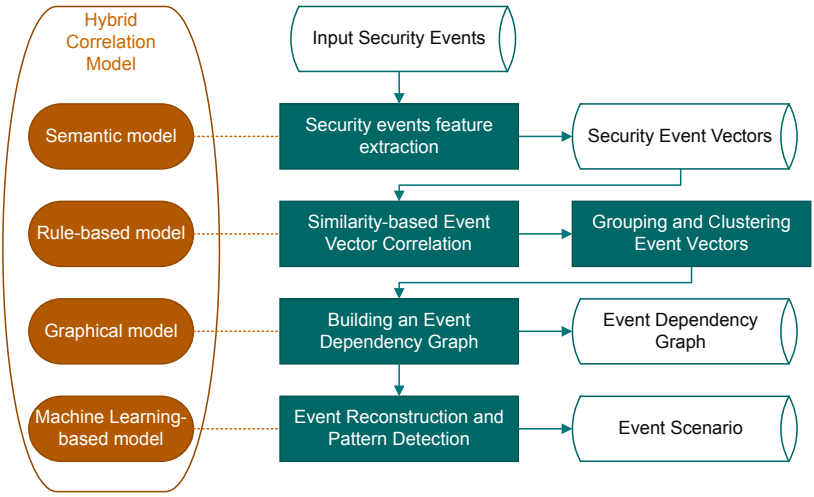

Fig. 2 Diagram of a hybrid correlation model

separates them into clusters. Interpretability and visualization of the results of the correlation model is supported by building dependency graphs of security events. Automation of the process of searching for patterns of security events, their classification, and prediction is achieved through a machine learning model. At the same time, when implementing supervised learning, the hybrid security event correlation model is able to detect known multi-step attacks as scenarios. When implementing unsupervised learning, the model is able to detect anomalies in normal event scenarios. In this way, a wide application coverage is achieved in terms of intelligent models and correlation methods, as well as applications.

It should be noted that this hybrid model can be adjusted depending on the characteristics of the target system and non-functional requirements for this system, such as resource consumption, time costs, and others.

In general, the task of event correlation can be defined in the following three stages: (1) calculation of pairwise similarity of events; (2) compilation of sequences of events as steps of a certain process; (3) determining the correspondence of event sequences. Figure 3 shows these stages for security events. Independent events describe system behavior, including anomalies. Anomalous behavior also can include attacking actions. The first two stages determine the relationship of events and compose their possible sequences. The third stage classifies or predicts event sequences using the knowledge base. So from a security point of view, certain event sequences may correspond to normal system behavior, anomalous behavior, or a specific type of attack. These categories, in turn, determine the target system state in the security assessment.

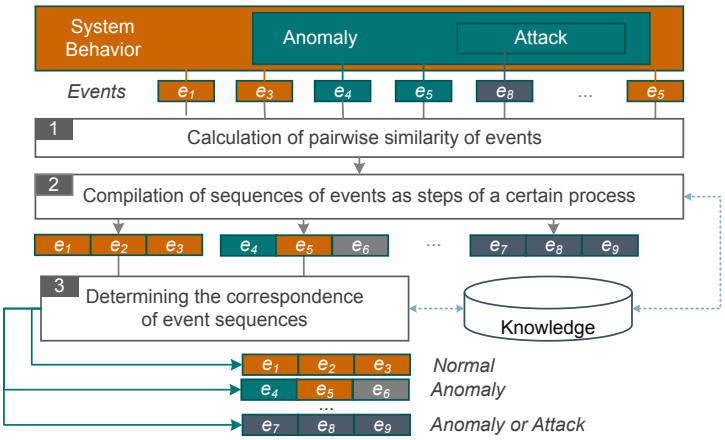

Fig. 3 Main stages of security event correlation

6 Challenges and Opportunities

Below we will present several potential topics in the correlation of security events that seem to us promising areas of research based on AI-based methods and existing challenges.

- **Hiding malicious patterns**. Compared to the tasks of analyzing natural language texts (Huang et al, 2018) and building knowledge bases based on event detection (Deng and Hooi, 2021), the detection of multi-step attacks is complicated by the attacker's intent to hide his trail. In particular, even if attack alerts are detected, it is often difficult to define the entire attacker logic without additional knowledge. So security event correlation methods can encounter missing data in probabilistic models. An attacker can also trick unsupervised anomaly detection methods by disguising some of the events as normal. A suitable solution is to study the attacker's behavior given the semantics and context of security events. Profiling of different types of attackers is also an important and promising area (Oki et al, 2018).
- **Explainability of event semantics**. This challenge follows directly from the previous one. Although AI-based methods such as deep learning methods are quite effective at detecting multi-step attacks, many of them operate in a "black box" manner (Du et al, 2017). In this case, it is more difficult for security operators to understand the semantics of the detected events. In addition, cyber-analysts are faced with a semantic gap between low-level audit events and high-level system behavior (Zeng et al, 2021). In this case, the use of interpreted intelligent algorithms will be useful additions, including in forensic problems.
- **Analysis of highly cardinal events**. In this case, we are talking about the high uniqueness of several significant categorical event features. Correlation methods based on ontologies and knowledge graphs cope well with

the analysis of such features. However, as a rule, shallow and deep learning models use encoding of such functions, for example, one-hot encoding, which, in turn, reduces the efficiency of processing high cardinal values.The development of algorithms for learning ontologies and event databases are a good help in processing of such data (Huang et al, 2021).

- **Analysis of large and/or heterogeneous data**. In this case, we are faced with the need to process events from different sources in various formats and with different semantics. Especially in distributed and complex systems, it becomes necessary to analyze a very large amount of data, which requires large computing resources. This task is particularly difficult when detecting attacks affecting multiple sources. Here there is also the problem of choosing the architecture of the correlation system: distributed, centralized or hierarchical. In the case of a large amount of event data, a good solution is to support parallel computing and use big data processing tools (Kotenko et al, 2018b). Heterogeneous event processing may include event unification algorithms or the use of correlation methods without a clear dependence on the event format.
- **Event knowledge base support**. As knowledge about events, there can be both correlation rules and patterns of known multi-step attacks or patterns of normal behavior. In this direction, we can also add the development of event ontologies. Creating such knowledge bases manually with the involvement of experts is quite laborious. In this case, an important direction is the development of adaptive event correlation methods or online learning algorithms for timely updating of pattern databases and knowledge graphs (Zheng et al, 2018).
- **Few publicly available datasets** to assess the correlation of cybersecurity events. Any learning methods largely depend on the availability of a reliable model-building dataset. Most researchers use private datasets that are not shared to reproduce experiments. For this reason, it is important not only to develop new event correlation mechanisms, but also to publish generated security event datasets with the appropriate use license and citation rules.

We hope that this survey and discussion will provide researchers in related fields with an understanding of the latest approaches to the correlation of security events, as well as interest in the development of intelligent event analysis methods applicable in cybersecurity.

7 Conclusion

This paper provided a review of the security event correlation literature over the last years. The review focuses on how AI-based techniques are applied to detect causal security issues, such as the attack scenario detection and prognosis. We presented the systematization of security event correlation models based on AI knowledge representation such as: rule-based models, semantic models, graphical models, machine learning-based models, and hybrid models. We provided comparison tables of the described AI-based correlation models

by application, model basis, correlation method, and security event data used. One of the results of the paper is also a description of the prospects for the development of hybrid correlation models. We also highlighted the challenges that researchers face when developing security event correlation approaches to stimulate future research. In general, they relate to the complexity of defining the logic of a multi-step attack and the complexity of processing a large amount of heterogeneous data. Hence, there is a need to improve approaches to event correlation, both in terms of semantics and learning capabilities.

Acknowledgments This work was supported by the Analytical Center for the Government of the Russian Federation (IGK 000000D730321P5Q0002), agreement No. 70-2021-00141.

References

Abdullayeva FJ (2021) Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm. Array 10:100,067

Ahmad F (2017) Web application firewall. https://github.com/faizann24/Fwaf-Machine-Learning-driven-Web-Application-Firewall, (Online; accessed 03-July-2022)

Albasheer H, Md Siraj M, Mubarakali A, et al (2022) Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: A survey. Sensors 22(4):1494

Almseidin M, Piller I, Al-Kasassbeh M, et al (2019) Fuzzy automaton as a detection mechanism for the multi-step attack. International Journal on Advanced Science, Engineering and Information Technology 9(2):575–586

Amza C, Cecchet E, Chanda A, et al (2002) Specification and implementation of dynamic web site benchmarks. In: 2002 IEEE International Workshop on Workload Characterization, pp 3–13

Asuncion A, Newman D (2007) Uci machine learning repository. http://archive.ics.uci.edu/ml/index.php, (Online; accessed 03-July-2022)

Bajtoš T, Sokol P, Mézešová T (2020) Multi-stage cyber-attacks detection in the industrial control systems. In: Recent Developments on Industrial Control Systems Resilience. Springer, Cham, p 151–173

Barzegar M, Shajari M (2018) Attack scenario reconstruction using intrusion semantics. Expert Systems with Applications 108:119–133

Beer F, Bühler U (2017) Feature selection for flow-based intrusion detection using rough set theory. In: 2017 IEEE 14th International Conference on

Networking, Sensing and Control (ICNSC), IEEE, pp 617–624

Bénard C, Biau G, Da Veiga S, et al (2021) SIRIUS: Stable and interpretable rule set for classification. Electronic Journal of Statistics 15(1):427–505

Bhattacharjya D, Shanmugam K, Gao T, et al (2020) Event-driven continuous time Bayesian networks. In: Proceedings of the AAAI conference on artificial intelligence, pp 3259–3266

Bojanowski P, Grave E, Joulin A, et al (2017) Enriching word vectors with subword information. Transactions of the association for computational linguistics 5:135–146

Bollacker K, Evans C, Paritosh P, et al (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp 1247–1250

Chang YC, Wang SD (2016) The concept of attack scenarios and its applications in Android malware detection. In: 2016 IEEE 18th International Conference on High Performance Computing and Communications, IEEE, pp 1485–1492

Chen H, Xiao R, Jin S (2020a) Real-time detection of cloud tenant malicious behavior based on CNN. In: 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE, pp 998–1005

Chen R, Zhang S, Li D, et al (2020b) Logtransfer: Cross-system log anomaly detection for software systems with transfer learning. In: 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), IEEE, pp 37–47

Cheng H, Xie Z, Shi Y, et al (2019) Multi-step data prediction in wireless sensor networks based on one-dimensional CNN and bidirectional LSTM. IEEE Access 7:117,883–117,896

Contagio Mobile (2011) Contagiodump. mobile malware sample. http://contagiominidump.blogspot.com/, (Online; accessed 03-July-2022)

Cook K, Grinstein G, Whiting M, et al (2012) Vast challenge 2012: Visual analytics for big data. In: 2012 IEEE conference on visual analytics science and technology (VAST), IEEE, pp 251–255

Creech G, Hu J (2013) A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. IEEE

Transactions on Computers 63(4):807–819

DARPA TC (2020) Transparent computing engagement 3 data release. https://github.com/darpa-i2o/Transparent-Computing, (Online; accessed 03-July-2022)

DEF CON Communications, Inc. (2021) Defcon21 ctf dataset. https://media.defcon.org/DEF%20CON%2021/, (Online; accessed 03-July-2022)

Dempster AP (2008) Upper and lower probabilities induced by a multivalued mapping. In: Classic works of the Dempster-Shafer theory of belief functions. Springer, Berlin, Heidelberg, p 57–72

Deng A, Hooi B (2021) Graph neural network-based anomaly detection in multivariate time series. Proceedings of the AAAI Conference on Artificial Intelligence 35(5):4027–4035

Deng S, Zhang N, Li L, et al (2021) OntoED: Low-resource event detection with ontology embedding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, pp 2828–2839

Devlin J, Chang MW, Lee K, et al (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp 4171–4186

Dhaou A, Bertoncello A, Gourvénec S, et al (2021) Causal and interpretable rules for time series analysis. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp 2764–2772

Do Xuan C, Dao MH (2021) A novel approach for APT attack detection based on combined deep learning model. Neural Computing and Applications 33(20):13,251–13,264

Du M, Li F, Zheng G, et al (2017) Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pp 1285–1298

Eckmann ST, Vigna G, Kemmerer RA (2002) Statl: An attack language for state-based intrusion detection. Journal of computer security 10(1-2):71–103

Garcia S, Grill M, Stiborek J, et al (2014) An empirical comparison of botnet detection methods. Computers & Security 45:100–123

Ghafouri A, Vorobeychik Y, Koutsoukos X (2018) Adversarial regression for detecting attacks in cyber-physical systems. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. AAAI Press, Stockholm, Sweden, IJCAI'18, pp 3769—-3775

Giménez CT, Villegas AP, Marañón GÁ (2010) Http data set csic 2010. https://www.isi.csic.es/dataset/, (Online; accessed 03-July-2022)

Glasser J, Lindauer B (2013) Bridging the gap: A pragmatic approach to generating insider threat data. In: 2013 IEEE Security and Privacy Workshops, IEEE, pp 98–104

Goh J, Adepu S, Junejo KN, et al (2016) A dataset to support research in the design of secure water treatment systems. In: International conference on critical information infrastructures security, Springer, pp 88–99

Guan S, Jin X, Wang Y, et al (2019) Link prediction on n-ary relational data. In: Proceedings of the 28th International Conference on World Wide Web (WWW'19), pp 583–593

Guo H, Yuan S, Wu X (2021) LogBERT: Log anomaly detection via BERT. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp 1–8

Haas S, Fischer M (2019) On the alert correlation process for the detection of multi-step attacks and a graph-based realization. ACM SIGAPP Applied Computing Review 19(1):5–19

Hamed T, Ernst JB, Kremer SC (2018) A survey and taxonomy of classifiers of intrusion detection systems. Computer and network security essentials pp 21–39

Han X, Pasquier T, Bates A, et al (2020) UNICORN: Runtime provenance-based detector for advanced persistent threats. In: Network and Distributed System Security Symposium, pp 1–18

Hassan WU, Guo S, Li D, et al (2019) Nodoze: Combatting threat alert fatigue with automated provenance triage. In: Network and Distributed Systems Security Symposium, pp 1–15

Hassan WU, Noureddine MA, Datta P, et al (2020) OmegaLog: High-fidelity attack investigation via transparent multi-layer log analysis. In: Network and Distributed System Security Symposium, pp 1–16

Heigl M, Weigelt E, Urmann A, et al (2021) Exploiting the outcome of outlier detection for novel attack pattern recognition on streaming data. Electronics 10(17):2160

Holgado P, Villagrá VA, Vazquez L (2017) Real-time multistep attack prediction based on hidden Markov models. IEEE Transactions on Dependable and Secure Computing 17(1):134–147

Hossain M, Xie J (2020) Third eye: Context-aware detection for hidden terminal emulation attacks in cognitive radio-enabled IoT networks. IEEE Transactions on Cognitive Communications and Networking 6(1):214–228

Hostiadi DP, Susila MD, Huizen RR (2019) A new alert correlation model based on similarity approach. In: 2019 1st International Conference on Cybernetics and Intelligent System (ICORIS), IEEE, pp 133–137

Huang L, Ji H, Cho K, et al (2018) Zero-shot transfer learning for event extraction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp 2160–2170

Huang Y, Sun H, Xu K, et al (2021) CoRelatE: Learning the correlation in multi-fold relations for knowledge graph embedding. Knowledge-Based Systems 213:106,601

Husák M, Komárková J, Bou-Harb E, et al (2018) Survey of attack projection, prediction, and forecasting in cyber security. IEEE Communications Surveys & Tutorials 21(1):640–660

Jaeger D, Ussath M, Cheng F, et al (2015) Multi-step attack pattern detection on normalized event logs. In: 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, IEEE, pp 390–398

Joloudari JH, Haderbadi M, Mashmool A, et al (2020) Early detection of the advanced persistent threat attack using performance analysis of deep learning. IEEE Access 8:186,125–186,137

Kent AD (2016) Cyber security data sources for dynamic network research. In: Dynamic Networks and Cyber-Security. WSPC (Europe), p 37–65

Khan MA, Abuhasel KA (2021) An evolutionary multi-hidden Markov model for intelligent threat sensing in industrial Internet of things. The Journal of Supercomputing 77(6):6236–6250

Khosravi M, Ladani BT (2020) Alerts correlation and causal analysis for APT based cyber attack detection. IEEE Access 8:162,642–162,656

Kim M, Park Y, Han I, et al (2020) A fast alert correlation method with Bayesian feature constraints. In: International Conference on Cyber Warfare and Security, Academic Conferences International Limited, pp 277–285

Kotenko I, Fedorchenko A, Saenko I, et al (2018a) Parallelization of security event correlation based on accounting of event type links. In: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), IEEE, pp 462–469

Kotenko I, Saenko I, Branitskiy A (2018b) Framework for mobile Internet of things security monitoring based on Big Data processing and machine learning. IEEE Access 6:72,714–72,723

Kotenko I, Saenko I, Ageev S (2019) Hierarchical fuzzy situational networks for online decision-making: Application to telecommunication systems. Knowledge-Based Systems 185:104,935

Kotenko I, Fedorchenko A, Doynikova E (2020) Data analytics for security management of complex heterogeneous systems: Event correlation and security assessment tasks. In: Advances in Cyber Security Analytics and Decision Systems. Springer, Cham, p 79–116

Kotenko I, Gaifulina D, Zelichenok I (2022) Systematic literature review of security event correlation methods. IEEE Access 10:43,387–43,420

Kovačević I, Groš S, Slovenec K (2020) Systematic review and quantitative comparison of cyberattack scenario detection and projection. Electronics 9(10):1722

Lallie HS, Debattista K, Bal J (2020) A review of attack graph and attack tree visual syntax in cyber security. Computer Science Review 35:100,219

Lanoe D, Hurfin M, Totel E (2018) A scalable and efficient correlation engine to detect multi-step attacks in distributed systems. In: 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS), IEEE, pp 31–40

Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning, PMLR, pp 1188–1196

Lee Y, Kim J, Kang P (2021) LAnoBERT: System log anomaly detection based on BERT masked language model. arXiv preprint arXiv:211109564 pp 1–15

Li S, Zhang Q, Wu X, et al (2021) Attribution classification method of APT malware in iot using machine learning techniques. Security and Communication Networks 2021:1–12

Liu F, Wen Y, Zhang D, et al (2019a) Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp 1777–1794

Liu J, Chen Y, Liu K (2019b) Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. Proceedings of the AAAI Conference on Artificial Intelligence 33(01):6754–6761

Liu L, Chen C, Zhang J, et al (2020) Doc2vec-based insider threat detection through behaviour analysis of multi-source security logs. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, pp 301–309

LL-MIT (1998) 1998 darpa intrusion detection evaluation dataset. https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset, (Online; accessed 03-July-2022)

LL-MIT (2000) 2000 darpa intrusion detection scenario specific dataset. https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets, (Online; accessed 03-July-2022)

Luo W, Zhang H, Yang X, et al (2020) Dynamic heterogeneous graph neural network for real-time event prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, p 3213–3223

Lv S, Qian W, Huang L, et al (2019) Sam-net: Integrating event-level and chain-level attentions to predict what happens next. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 6802–6809

Ma Y, Wu Y, Yu D, et al (2022) Vulnerability association evaluation of internet of thing devices based on attack graph. International Journal of Distributed Sensor Networks 18(5):15501329221097,817

Mahdavi E, Fanian A, Amini F (2020) A real-time alert correlation method based on code-books for intrusion detection systems. Computers & Security 89:101,661

Mao B, Liu J, Lai Y, et al (2021) Mif: A multi-step attack scenario reconstruction and attack chains extraction method based on multi-information fusion. Computer Networks 198:108,340

Meidan Y, Bohadana M, Mathov Y, et al (2018) N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing 17(3):12–22

Meier M, Bischof N, Holz T (2002) SHEDEL—a simple hierarchical event description language for specifying attack signatures. In: Security in the Information Society. Springer US, Boston, MA, p 559–571

Mikolov T, Chen K, Corrado G, et al (2013a) Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, pp 1–12

Mikolov T, Sutskever I, Chen K, et al (2013b) Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 26:1–9

Milajerdi SM, Gjomemo R, Eshete B, et al (2019) Holmes: real-time apt detection through correlation of suspicious information flows. In: 2019 IEEE Symposium on Security and Privacy (SP), IEEE, pp 1137–1152

Miller GA (1995) Wordnet: a lexical database for english. Communications of the ACM 38(11):39–41

Min B, Yoo J, Kim S, et al (2021) Network anomaly detection using memory-augmented deep autoencoder. IEEE Access 9:104,695–104,706

Mirheidari SA, Arshad S, Jalili R (2013) Alert correlation algorithms: A survey and taxonomy. In: International Symposium on Cyberspace Safety and Security, Springer, pp 183–197

Morzeux (2020) Httpparamsdataset. https://github.com/Morzeux/HttpParamsDataset, (Online; accessed 03-July-2022)

Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp 1–6

Nasr M, Bahramali A, Houmansadr A (2018) Deepcorr: Strong flow correlation attacks on tor using deep learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, p 1962–1976

NETRESEC (2000) Mid-atlantic collegiate cyber defense competition. 2012. https://www.netresec.com/?page=MACCDC, (Online; accessed 03-July-2022)

NETRESEC (2015) 4sics geek lounge. https://www.netresec.com/?page=PCAP4SICS, (Online; accessed 03-July-2022)

Nguyen T, Grishman R (2018) Graph convolutional networks with argument-aware pooling for event detection. Proceedings of the AAAI Conference on Artificial Intelligence 32(1):5900–5907

Oki M, Takeuchi K, Uematsu Y (2018) Mobile network failure event detection and forecasting with multiple user activity data sets. Proceedings of the AAAI Conference on Artificial Intelligence 32(1):7786–7792

Oliner A, Stearley J (2007) What supercomputers say: A study of five system logs. In: 37th annual IEEE/IFIP international conference on dependable systems and networks (DSN'07), IEEE, pp 575–584

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

Peters ME, Neumann M, Iyyer M, et al (2018) Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ACL, New Orleans, Louisiana, pp 2227–2237

Ramaki AA, Rasoolzadegan A, Bafghi AG (2018) A systematic mapping study on intrusion alert analysis in intrusion detection systems. ACM Computing Surveys (CSUR) 51(3):1–41

Ramilli M (2016) Malware training sets: a machine learning dataset for everyone. https://marcoramilli.com/2016/12/16/malware-training-sets-a-machine-learning-dataset-for-everyone/, (Online; accessed 03-July-2022)

Ring M, Schlör D, Wunderlich S, et al (2021) Malware detection on windows audit logs using LSTMs. Computers & Security 109:102,389

Ryciak P, Wasielewska K, Janicki A (2022) Anomaly detection in log files using selected natural language processing methods. Applied Sciences 12(10):5089

Salah S, Maciá-Fernández G, Díaz-Verdejo JE (2013) A model-based survey of alert correlation techniques. Computer Networks 57(5):1289–1317

Sarker IH, Furhad MH, Nowrozy R (2021) AI-driven cybersecurity: an overview, security intelligence modeling and research directions. SN Computer Science 2(3):1–18

Sen Ö, van der Velde D, Wehrmeister KA, et al (2022) On using contextual correlation to detect multi-stage cyber attacks in smart grids. Sustainable Energy, Grids and Networks 32:100,821

Seyyar YE, Yavuz AG, Unver HM (2022) An attack detection framework based on BERT and deep learning. IEEE Access Early Access:1–13

Sharafaldin I, Lashkari AH, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), pp 108–116

Shawly T, Elghariani A, Kobes J, et al (2019) Architectures for detecting interleaved multi-stage network attacks using hidden Markov models. IEEE Transactions on Dependable and Secure Computing 18(5):2316–2330

Shen Y, Mariconti E, Vervier PA, et al (2018) Tiresias: Predicting security events through deep learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp 592–605

Shiravi A, Shiravi H, Tavallaee M, et al (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection. computers & security 31(3):357–374

Siddiqui AJ, Boukerche A (2021) TempoCode-IoT: temporal codebook-based encoding of flow features for intrusion detection in internet of things. Cluster Computing 24(1):17–35

Sikos LF (2021) Ai in digital forensics: Ontology engineering for cybercrime investigations. Wiley Interdisciplinary Reviews: Forensic Science 3(3):e1394

Stephan G, Pascal H, Andreas A, et al (2007) Knowledge representation and ontologies. In: Semantic Web Services: Concepts, Technologies, and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg, p 51–105

Sun J, Gu L, Chen K (2020) An efficient alert aggregation method based on conditional rough entropy and knowledge granularity. Entropy 22(3):324

Sun X, Dai J, Liu P, et al (2018) Using Bayesian networks for probabilistic identification of zero-day attack paths. IEEE Transactions on Information Forensics and Security 13(10):2506–2521

Tanwar P, Prasad T, Aswal MS (2010) Comparative study of three declarative knowledge representation techniques. International Journal on Computer Science and Engineering 2(07):2274–2281

Tao Xl, Shi L, Zhao F, et al (2021) A hybrid alarm association method based on AP clustering and causality. Wireless Communications and Mobile Computing 2021(5):1–10

Tavallaee M, Bagheri E, Lu W, et al (2009) A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications, IEEE, pp 1–6

Tidjon LN, Frappier M, Mammar A (2020) Intrusion detection using ASTDs. In: International Conference on Advanced Information Networking and Applications. Springer, Cham, pp 1397–1411

Viterbi A (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13(2):260–269

Walker C, Strassel S, Medero J, et al (2006) Ace 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia 57:45

Wang J, Tang Y, He S, et al (2020) LogEvent2vec: Logevent-to-vector based anomaly detection for large-scale logs in internet of things. Sensors 20(9):2451

Wang J, Zhao C, He S, et al (2022) LogUAD: Log unsupervised anomaly detection based on Word2Vec. Computer Systems Science and Engineering 41(3):1207–1222

Wang Q, Jiang J, Shi Z, et al (2018) A novel multi-source fusion model for known and unknown attack scenarios. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, pp 727–736

Wang X, Gong X, Yu L, et al (2021a) MAAC: Novel alert correlation method to detect multi-step attack. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, pp 726–733

Wang Z, Chen Z, Ni J, et al (2021b) Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp 3726–3734

Welch LR (2003) Hidden Markov models and the Baum-Welch algorithm. IEEE Information Theory Society Newsletter 53(4):10–13

Wen J, Li J, Mao Y, et al (2016) On the representation and embedding of knowledge bases beyond binary relations. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pp 1300–1307

Xie T, Zheng Q, Zhang W (2018) Mining temporal characteristics of behaviors from interval events in e-learning. Information Sciences 447:169–185

Xu W, Huang L, Fox A, et al (2009) Online system problem detection by mining patterns of console logs. In: 2009 ninth IEEE international conference

on data mining, IEEE, pp 588–597

Yu Beng L, Ramadass S, Manickam S, et al (2014) A survey of intrusion alert correlation and its design considerations. IETE Technical Review 31(3):233–240

Zegeye WK, Dean RA, Moazzami F (2018) Multi-layer hidden Markov model based intrusion detection system. Machine Learning and Knowledge Extraction 1(1):265–286

Zeng J, Wu S, Chen Y, et al (2019) Survey of attack graph analysis methods from the perspective of data and knowledge processing. Security and Communication Networks 2019:1–16

Zeng J, Chua ZL, Chen Y, et al (2021) Watson: Abstracting behaviors from audit logs via aggregation of contextual semantics. In: Proceedings of the 28th Annual Network and Distributed System Security Symposium, NDSS, pp 1–18

Zhan Y, Haddadi H (2019) Towards automating smart homes: Contextual and temporal dynamics of activity prediction. In: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, pp 413–417

Zhang X, Wu T, Zheng Q, et al (2022) Multi-step attack detection based on pre-trained hidden Markov models. Sensors 22(8):2874

Zhang Y, Zhao S, Zhang J (2019) RTMA: Real time mining algorithm for multi-step attack scenarios reconstruction. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications, IEEE, pp 2103–2110

Zheng H, Wang Y, Han C, et al (2018) Learning and applying ontology for machine learning in cyber attack detection. In: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, pp 1309–1315

Zimba A, Chen H, Wang Z (2019) Bayesian network based weighted APT attack paths modeling in cloud computing. Future Generation Computer Systems 96:525–537