IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 6, 2022, accepted April 10, 2022, date of publication April 20, 2022, date of current version April 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3168976

Systematic Literature Review of Security Event Correlation Methods

IGOR KOTENKO, (Senior Member, IEEE), DIANA GAIFULINA, AND IGOR ZELICHENOK

St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), 199178 St. Petersburg, Russia

Corresponding author: Igor Kotenko (ivkote@comsec.spb.ru)

This work was supported by the grant of Russian Science Foundation (RSF) under Grant #21-71-20078 in SPC RAS.

ABSTRACT Security event correlation approaches are necessary to detect and predict incremental threats such as multi-step or targeted attacks (advanced persistent threats) and other causal sequences of abnormal events. The use of security event correlation techniques also makes it possible to reduce the volume of the original data stream by grouping the events and eliminating their redundancy. The variety of event correlation methods, in turn, requires choosing the most appropriate way to handle security events, depending on the purpose and available resources. This paper presents a systematization of security event correlation methods into several categories, such as publication year, applied correlation methods, knowledge extraction methods, used data sources, architectural solutions, and quality evaluation of correlation methods. The research method is a systematic literature review, which includes the formulation of research questions, the choice of keywords and criteria for inclusion and exclusion. The review corpus is formed by using search queries in Google Scholar, IEEE Xplore, ACM Digital Library, ScienceDirect, and selection criteria. The final review corpus includes 127 publications from the existing literature for 2010-2021 and reflects the current state of research in the security event correlation field. The results of the analysis include the main directions of research in the field of event correlation and methods used for correlation both single events and their sequences in attack scenarios. The review also describes the datasets and metrics used to evaluate security event correlation approaches. In conclusion, the existing problems and possible ways to overcome them are identified. The main contribution of the review is the most complete classification and comparison of existing approaches to the security event correlation, considered not only from the point of view of the algorithm, but also the possibility of unknown attack detection, architectural solutions and the use of event initial data.

INDEX TERMS Advanced persistent threat, alert correlation, attack scenario, information security, intrusion detection, multi-step attacks, network security, security event correlation.

I. INTRODUCTION

Every day, modern systems are becoming more complex in terms of architecture, processed data and tasks. At the same time, the attack vectors and protection methods for these systems are also becoming more complicated. Security analytics tools should collect security information, detect intrusions, and analyze security events in the form of messages or alarms related to activity in a system or a network. Such messages can be information from network packets, system logs, application logs, and other data sources. As part of security, events that are part of a multi-stage attack or an independent threat should be detected. Their prerequisites also should be defined to prevent such events in the future.

To solve the problems of security event management, analysts use various event processing methods. Correlation occupies an important place among these methods and determines connections between heterogeneous events and incidents. The use of security event correlation techniques also allows one to decrease the amount of the original data stream by grouping the events and eliminating their redundancy. Determining the relationships between events from different sources contributes to a better understanding of the attack development and the most significant event identification.

There are a several reviews devoted to security event correlation approaches [1]–[6], as well as to the detection

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

of multi-step [7]–[9] and targeted attacks [10]. These reviews vary in depth and research methods used, including systematic literature review (SLR). This method of review, as a rule, includes the formulation of research questions, the selection of keywords to search for papers and inclusion and exclusion criteria determination to form the final corpus of studies that will be included in the review.

We provide a systematic literature review of the security event correlation approaches published in the scientific literature over the past decade. The difference from some other systematic surveys [6], [8]–[10] is determined by the research questions presented in the proposed research methodology. The questions posed make it possible both to assess the current state of the research area and to identify future actual issues. This review describes the complex state of the security event correlation technology based on the proposed classification, which contains various aspects of event correlation systems, including the methods and architectural solutions. Also, one of the review tasks is to describe the datasets and evaluation metrics used by the authors in scientific papers on this topic. We present a comparison of experimental research results.

The main contribution of the paper:

- summarizing the most relevant papers about security event correlation technologies published in recent years;
- the most complete classification and analysis of existing security event correlation approaches, considered not only from the point of view of correlation methods but also from the point of view of architectural solutions, the possibility of unknown attack detection and using the initial data about events;
- comparison of the security event correlation technologies by the used datasets, methods, and evaluation results;
- identification of the most significant problems in the field of security event correlation and possible ways to overcome them;
- presentation of results in a well-structured and accessible form for a large readership.

The paper is organized as follows. Section II contains a clarification of the main terms used. Section III presents an analysis of relevant reviews in the field of security event correlation. Section IV describes the review methodology, including research questions, search strategy, and criteria for inclusion and exclusion of papers for subsequent analysis. Section V provides the classification and review of selected publications that suggest security event correlation approaches. Section VI describes the results of the presented research issues. Section VII concludes the paper.

II. TERMINOLOGY

This section describes the most important terms for research. This is to avoid confusion arising from the fact that terms may be interpreted differently in diverse sources. Table 1 shows the main abbreviations and designations used in the paper.

TABLE 1. List of acronyms and notations used in this article.

Notation	Description
AE	Autoencoder
AG	Attack Graph
ANN	Artificial Neural Network
API	Application Programming Interface
APT	Advanced Persistent Threat
CNN	Convolutional Neural Network
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DoS	Denial of Service
DDoS	Distributed Denial of Service
DL	Deep Learning
DoS	Denial of Service
EDL	Event Description Language
FSM	Finite-state Machine
FTP	File Transfer Protocol
HIDS	Host-based Intrusion Detection System
HMM	Hidden Markov Model
HTTP	HyperText Transfer Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
IRC	Internet Relay Chat
IT	Information Technology
k-NN	K-Nearest Neighbors
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
NIDS	Network-based Intrusion Detection System
NVD	National Vulnerability Database
OWL	Web Ontology Language
P2P	Point-to-Point
PCA	Principal Component Analysis
R2L	Remote to Local Attack
RF	Random Forest
RNN	Recurrent Neural Network
SCADA	System Control and Data Acquisition
SDN	Software-Defined Networking
SSH	Secure Shell
SIEM	Security Information and Event Management
SLR	Systematic Literature Review
SOM	Self-Organizing Map
SVM	Support Vector Machine
SQL	Structured Query Language
TCP	Transmission Control Protocol
VMM	Variable-order Markov Model
U2R	User to Root Attack

A. EVENTS AND ALERTS

In general terms, events are individual or cumulative messages or alarms related to activity on a system or a network [1]. Thus, a security event can contain information about both the contents of the network packet and a message about the exploit. In the second case, such events are often called alerts.

An alert is a message that an event of interest has been detected, which typically contains information about unusual activity [11]. This term is often used by authors as a synonym for "security event". Therefore, in this review, we will use these two terms interchangeably. The term "alarm" is also sometimes used to denote how the security system responds to suspected malicious activity or errors [5].

B. MULTI-STEP AND TARGETED ATTACKS

A multi-step attack is an aggregate of steps, containing at least two different actions, taken by one or more attackers with one specific target within a network [8]. This action can be known as a simple or single-step attack. At the same time, the steps of the attack are not isolated but are interconnected by some logical relationships. Thus, only with unrelated repetitive similar actions, an attack may not be considered a multi-step attack. The authors also call multi-step attacks using the terms ''multi-stage'', ''attack scenarios'', ''attack strategies'', and others.

Some 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 or network is maintained for a long period of time [10]. APT-based attacks are usually covert, multi-action, long-term, and based on a set of different zero-day vulnerabilities.

C. INTRUSION DETECTION SYSTEMS

An example of an alert generating system is an intrusion detection system (IDS) that detects suspicious activity. By organization principle, IDS is classified into host-based IDS (HIDS) [12] and network-based IDS (NIDS) [13]. HIDS monitors events in the incoming and outgoing traffic of the computer system on one host, and NIDS monitors the entire protected network. The IDS approach can be based on both signature detection of known attacks and anomaly detection when checking for compliance with normal system behavior [14]. If a detection system can further mitigate and prevent identified attacks, it is called an intrusion prevention system (IPS).

D. SECURITY INFORMATION AND EVENT MANAGEMENT

Technologies like Security Information and Event Management (SIEM) enable the analysis of security events. SIEM systems form the central platform of modern security centers and analyze events and alerts from multiple sensors such as IDSs, antiviruses, firewalls, etc. Event analysis includes event aggregation, storage, correlation, and security reporting [15]. Typically, a simple SIEM system consists of separate units responsible for collecting, storing and managing security events, which can also work independently of each other.

E. SECURITY EVENT CORRELATION

The event correlation process is the process of finding the relationships between events. Correlation creates context between individual events and information previously collected in real-time, and also normalizes it for subsequent processing. [1]. The primary purpose of alert correlation is to identify the most significant events in the security dataset. Security event correlation should increase the quality of information about events while decreasing their number and interpreting multiple alarms.

Figures 1-2 show diagrams for a better understanding of the relationships between the terms given in this section.

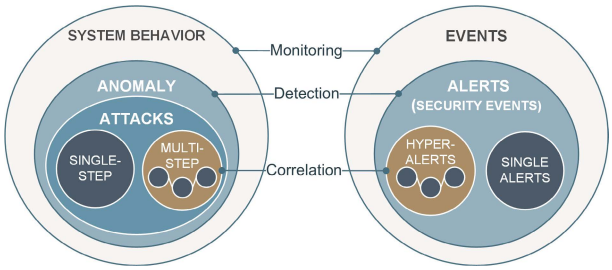

FIGURE 1. Role of correlation in security event management.

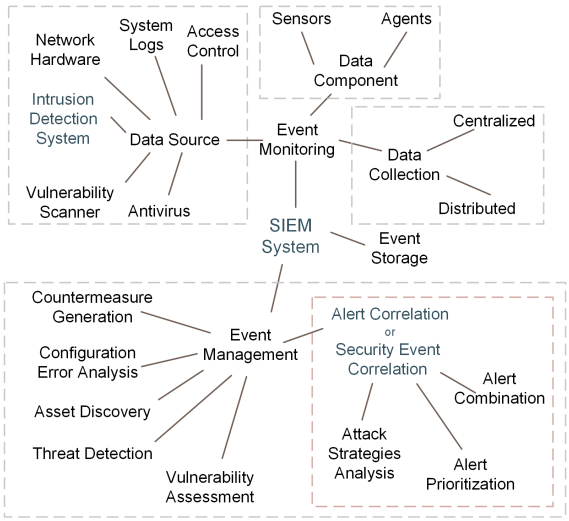

FIGURE 2. Security event management relations.

In Figure 1, events are the result of system behavior monitoring, and alerts or security events are the results of abnormal activity detection, which also include single-step and multi-step attacks. The correspondences between concepts are indicated by color. The security event correlation process, in turn, allows one to define relationships between single alerts, at the same time the related alerts can be combined into a meta-event or a hyper-alert and categorized in different ways.

As mentioned earlier, SIEM performs the functions of monitoring, storing and managing events. The Figure 2 shows the relationships between the main concepts in the security event management. We can trace the relationships between the previously mentioned terms of intrusion detection, SIEM and alert correlation (indicated by color).Intrusion detection is the source of security events. These events are monitored by SIEM system and then correlated in the management process. In addition to the combination and prioritization of alerts, an important task of correlation is the analysis of attack strategies. Related security events are identified as steps of attackers to achieve the goal and are analyzed as part of a specific attack strategy.

For this reason, the security event correlation approaches in this review also include methods for multi-step attack detection, since they consist in finding relationships between events belonging to the same attack scenario.

TABLE 2. Relevant surveys.

Authors	Year	Period	Number of papers	Research method	Search strategy	Paper's analysis
Limmer & Dressler [1]	2008	1998-2007	52	LR	–	–
Elshoush & Osman [2]	2011	2001-2009	18	LR	–	–
Mirheidari et al. [4]	2013	2001-2013	49	LR	–	–
Salah et al. [5]	2013	1996-2011	83	LR	–	+
Yu Beng et al. [3]	2014	2005-2012	22	LR	–	+
Luh et al. [10]	2017	2001-2014	60	SLR	+	+
Ramaki et al. [6]	2018	2000-2017	136	SLR	+	+
Husák et al. [7]	2018	2003-2018	63	LR	–	+
Navarro et al. [8]	2018	2000-2017	181	SLR	+	+
Kovačević et al. [9]	2020	1995-2020	57	SLR	+	+
Pavlov & Voloshina [16]	2020	2001-2019	27	LR	–	–
Kotenko et al. (this survey)	**2022**	**2010-2021**	**127**	**SLR**	**+**	**+**

III. RELEVANT SURVEYS

In this section, we look at the existing relevant surveys in the security event correlation area. The Table 2 presents a brief description of the selected surveys in terms of the year of publication, the number of reviewed papers and their period, field, methods, and research parameters.

There are two types of review:

- literature review (LR) – a simple description of studies, there is no strategy for searching and selecting studies;
- systematic literature review (SLR) – a survey of studies describing the methodology for search and selection of studies, including the formulation of research questions, selection of keywords and criteria for inclusion and exclusion [17].

Limmer and Dressler [1] categorize correlation methods by layer of data processing. Each level tries to filter out as many irrelevant events as possible, match the relevant events and aggregate them accordingly. Raw data level performs packet sampling, probabilistic analysis, anomaly detection, detection of port scans, application identification and payload analysis. The main purpose of the event layer is to collect as much information as possible, aggregating multiple events through local or distributed correlation. The report layer generates possible active countermeasures and checks for security events. The authors also classify the used correlation algorithms for signature matching or anomaly detection.

Elshoush and Osman [2] classify approaches to correlating collaborative IDS alerts into five classes. The similarity-based methods provide alert correlation using the similarity between some of their features; events, in which the similarity value is large enough, are grouped. The attack scenario-based methods analyze alerts using predefined attack scenarios defined by experts or obtained from training datasets. The prerequisites and consequences methods reconstruct some complex attack scenarios by linking the individual steps of the attacker in such a way that one of the stages of the attack is a prerequisite for the other. Methods based on multiple information sources combine different types of information. Methods based on filtering algorithms remove and prioritize security events according to predefined rules.

Similar methods are used to classify alert correlation approaches by Yu Beng *et al.* [3], additionally including expert systems and data mining. A similar division of approaches is also presented in the study by Pavlov and Voloshina [16]. At the same time, the authors consider attribute-based algorithms in terms of property-based, timestamp-based and statistical-based relationships.

Mirheidari *et al.* [4] classify correlation algorithms into three groups. Similarity-based algorithms include simple rules, hierarchical rules, and machine learning. Knowledge-based algorithms rely on known data about attack scenarios, such as the prerequisites and consequences of events. Statistical-based algorithms contain three groups of algorithms. The first group of statistical algorithms detects regularly repeated alerts and determines their patterns. The second group finds out repeated sequences of alerts. The third group evaluates the priority of an alert based on its approval by other data sources.

Salah *et al.* [5] categorize alert correlation approaches, in addition to the methods used, into other aspects such as the number of data sources, type of application and type of architecture. So, approaches can accept data from only one data source or from several. The authors identify three main areas of application: network management systems, IT security, and process control in manufacturing systems (SCADA). The applied architectural solutions can be centralized, distributed and hierarchical. The architecture is determined by whether the central node or distributed network agents perform event correlation. The authors distinguish three main groups of correlation methods: similarity-based, sequential-based and case-based methods. To sequential-based correlation methods, the authors add codebook, Markov models, Bayesian networks, and neural networks.

Luh *et al.* [10] discuss semantic methods for targeted attack detection. Existing solutions include host-based, network-based and multi-source approaches, and purely semantic-based approaches that cannot be attributed to a specific domain. Multi-source approaches focus on data fusion and correlation. The main applications for these approaches are SIEM-like systems and event correlation systems.

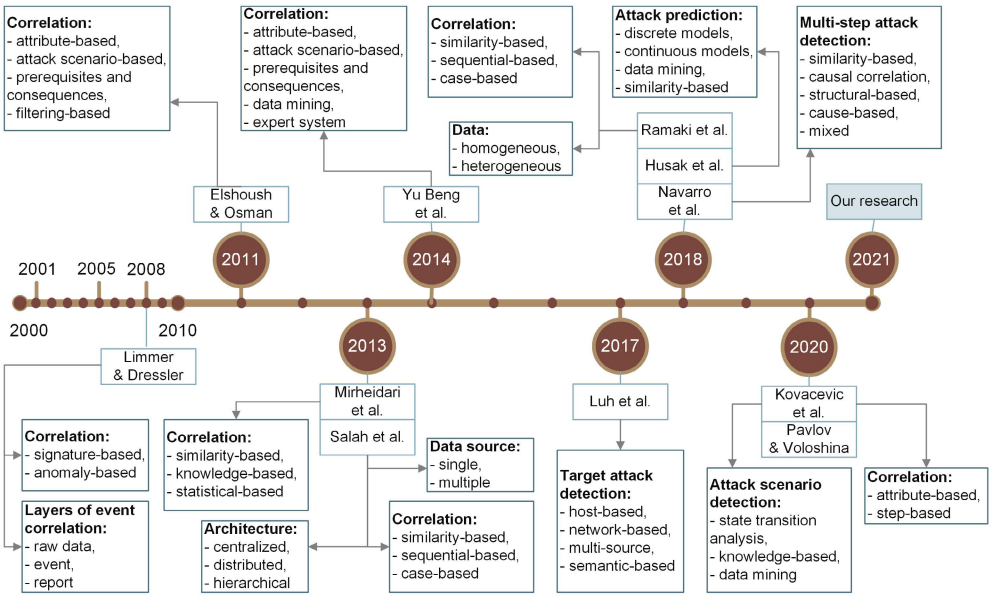

FIGURE 3. Classifications of approaches for security event correlation in relevant surveys.

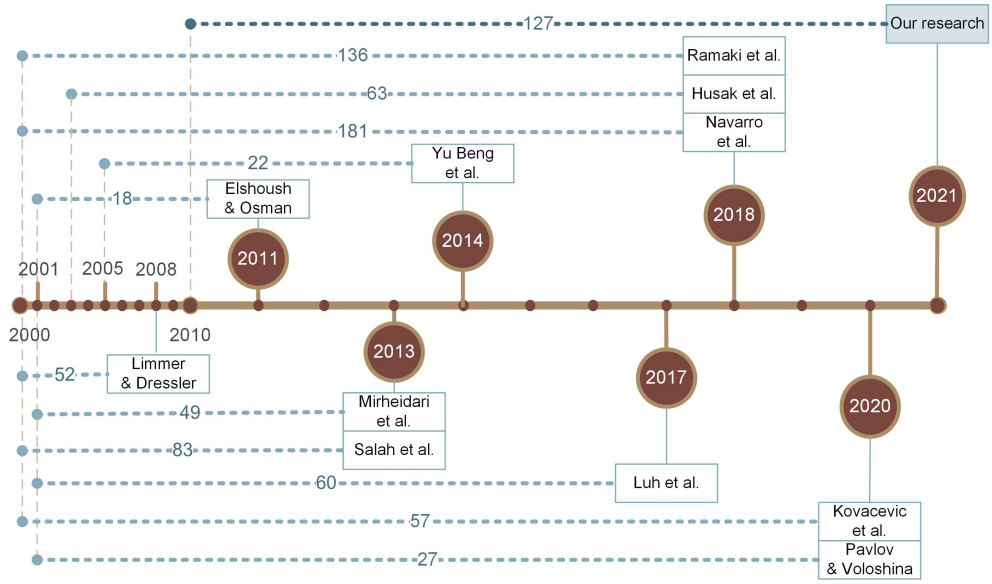

FIGURE 4. Timeline of relevant surveys.

Ramaki *et al.* [6] provide a systematic review and identify ten main sub-processes (functionalities) when analyzing security events. Event pre-processing includes normalization and verification. Event processing is aggregation, correlation, new attack strategy detection and missed attack hypothesizing. Event post-processing is prediction, prioritization, impact analysis and visualization. Correlation is divided into two subclasses: correlation of homogeneous alerts and correlation of heterogeneous alerts. The classification of correlation algorithms is similar to that presented in [5].

Husák *et al.* [7] analyze attack prediction models. The first group of such methods are discrete models, which include attack graphs, Bayesian networks, Markov models and game theory. The second group of methods contains methods based on continuous models, such as time series and gray models. The third group of methods includes methods based on machine learning and data mining. The fourth group comprises similarity-based approaches and evolutionary computing.

Navarro *et al.* [8] consider approaches to multi-step attack detection and classify them into five classes according to the method used. Similarity-based methods include progressive construction by attribute matching or attribute correlation, scenario clustering and anomaly detection. Causal correlation includes prerequisites and consequences models, statistical inference and model matching. Structural-based correlation

uses a network model where future attack paths can be predicted. Case-based approaches detect known attack scenarios as an ensemble of traces. Mixed approaches use several methods, but none of them stands out from the others.

Kovačević *et al.* [9] conduct a systematic review of methods for detecting and preventing attack scenarios, identifying three groups of methods. The first group includes state transition analysis based on explicit attack scenario signatures, and the second group includes alert correlation relying on expert knowledge, which uses rules to construct attack scenarios. The third group includes alert correlation relying on data mining and machine learning.

Figure 3 demonstrates the main classifications of approaches for security event correlation in the presented surveys in a timeline format. We use the criteria identified in relevant studies to compile our own complete classification of security event correlation approaches. Figure 4 shows a timeline of relevant surveys compared to our review. Although we only consider publications from the last decade, we include a larger number of them in the review. This improves the completeness of the description of alert correlation studies.

Though the views vary in depth, they all in one way or another relate to the topic of security event analysis, which is to find the connection between various events and alerts. In this review, we present our classification of security event correlation approaches as part of a systematic analysis, which also includes a classification with significantly wider coverage and a comparison of approaches according to the datasets, methods and evaluating results. The difference from some other SLRs [6], [8]–[10] is determined by the research questions described in the proposed research methodology. Our review contains several recently proposed approaches that were not included in the above papers.

IV. REVIEW METHODOLOGY

This study is based on the recommendations for systematic literature review by Kitchenham *et al.* [17] and Petersen *et al.* [18]. This method of bibliographic review allows one to identify the main directions of research in the security event correlation field and identify existing gaps. This section describes the review questions, search strategy and criteria for including and excluding publications for later analysis.

A. RESEARCH QUESTIONS

The main questions that we aim to answer in the review:

- **RQ1:** What topics are considered by security event correlation researchers?
- **RQ2:** What are the approaches to security event correlation?
- **RQ3:** What datasets are used to evaluate approaches to security event correlation?
- **RQ4:** What metrics are used to evaluate approaches to security event correlation?
- **RQ5:** What security event correlation approaches are promising for future research?

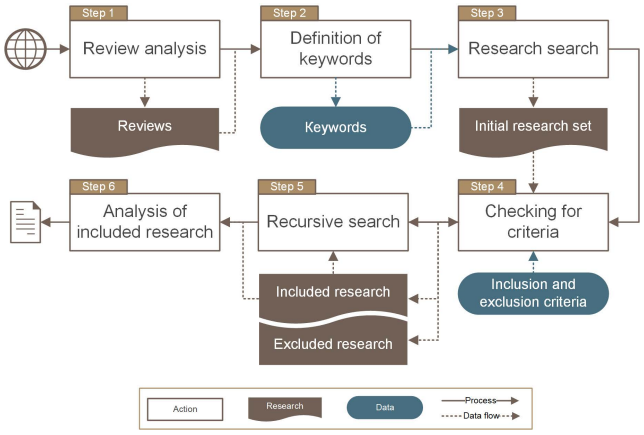

FIGURE 5. Search strategy.

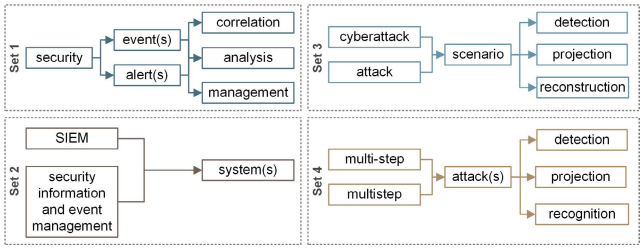

FIGURE 6. Keyword sets.

B. SEARCH STRATEGY

This review examines the research results published in scientific journals. The sources considered do not include articles in non-scientific journals or commercial white papers, presentations, and slides. We also do not consider patents for inventions. The research search strategy according to [17], [18] is shown in Figure 5.

Step 1. Identify and analyze relevant surveys focusing on research in security event correlation and detection of multi-step and targeted attacks. This analysis is presented in III.

Step 2. Keyword extraction. When analyzing relevant reviews, we identify the most popular words and phrases found in the titles of the paper on the selected topic. Based on the results of the analysis, the following sets of keywords were compiled for search queries:

- **Set 1:** security + {event(s), alert(s)} + {correlation, analysis, management}.
- **Set 2:** {SIEM, security information and event management} + system(s).
- **Set 3:** {cyberattack, attack} + scenario + detection, projection, reconstruction.
- **Set 4:** {multi-step, multistep} + attack(s) + {detection, projection, recognition}.

A combination of different keywords in one search is indicated by the "+" symbol. The curly braces show options for selecting parts of the key phrase. Some parts are considered singular and plural, which is indicated by the letter "s" in parentheses. Figure 6 shows the used key strings.

Step 3. Search for publication based on a set of keywords in electronic databases. Google Scholar, IEEE Xplore, ACM Digital Library and ScienceDirect were selected as search engines. The result of this step is an initial set of publications.

Step 4. Check the initial set of publications for inclusion and exclusion criteria. These criteria allow one to assess whether the publication will be included in the final sample for subsequent review.

Step 5. Recursive search for relevant references. The selected relevant publications are considered similar to the initial set from step 4 and are also added to the final corpus or excluded from the review.

C. INCLUSION AND EXCLUSION CRITERIA

For inclusion or exclusion of scientific publications in the final review corpus, we apply the appropriate criteria. If a published scientific study does not meet any of the inclusion criteria or has at least one of the exclusion criteria, then it will be excluded. Otherwise, it will be added to the final corpus.

Inclusion criteria:

- **IC1.** The authors of the publication propose a security event correlation approach, including attack scenario detection.
- **IC2.** The publication presents an evaluation of the proposed approach in an experimental, mathematical or another way.
- **IC3.** The publication has a clear structure. The methods are presented clearly and visually.
- **IC4.** The publication period of the study is 2010-2021.
- **IC5.** The publication is written in English in compliance with stylistic and grammatical norms.

Exclusion criteria:

- **EC1.** The publication is not about security event analysis or attack scenario detection.
- **EC2.** The publication is about security event or attack scenario analysis but does not contain approaches to their detection and correlation.
- **EC3.** The publication is a review or comparison of correlation methods.
- **EC4.** The presented approach is described without validation on data or in another way.
- **EC5.** The publication contains information duplicated from other papers of its authors.
- **EC6.** The presented approach is poorly understood. The publication has no clear structure and/or is written in unscientific language.
- **EC7.** The publication was not written in English.

D. SEARCH RESULTS

We conduct searches for selected keywords conducted in Google Scholar, IEEE Xplore, ACM Digital Library and ScienceDirect. For each search result, we export the first 50 most relevant publications. Thus, a search by titles of publications in electronic databases of scientific publications give 800 studies. Removing duplicate papers give 512 publications as initial set. Further, checking for inclusion and exclusion criteria resulted in the deletion of the following number of articles: based on title – 266, based on annotation – 73, based on the entire text – 78. Thus, the initial number of included publications was 96 papers. Recursive search for relevant papers added 31 studies according to the criteria. The final review corpus is **127** papers.

Figure 7 shows the distribution of excluded studies according to the main reason for exclusion. Most of the studies are excluded from the review, as they do not offer a specific approach to alert correlations. For example, research might focus on risk analysis, event modelling and attack scenarios, or visualization of security event correlation results. Papers that describe the analysis of multimedia data such as captchas, images, and videos are also excluded from the review corpus.

Figure 8 shows the distribution of studies by years of publication. The prevailing number of publications in the review corpus based on the search results was written in 2010-2014, since the papers of this period are the most cited, and it can be assumed that search engines offer them as the most relevant for queries. Recursive search also allows us to find earlier publications. At the same time, the area of security event correlation does not cease to be relevant, and in recent years no less number of scientific papers have been published.

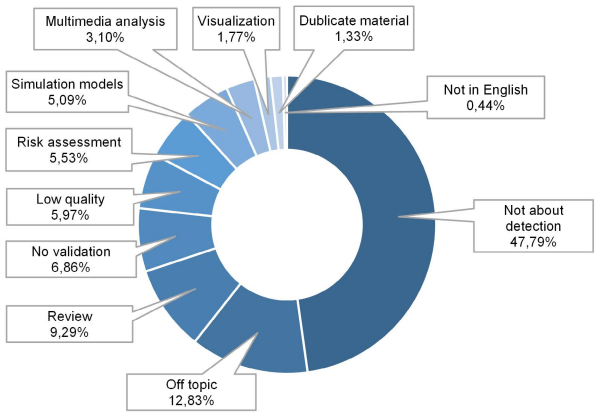

FIGURE 7. Distribution of reasons for excluding studies from the review.

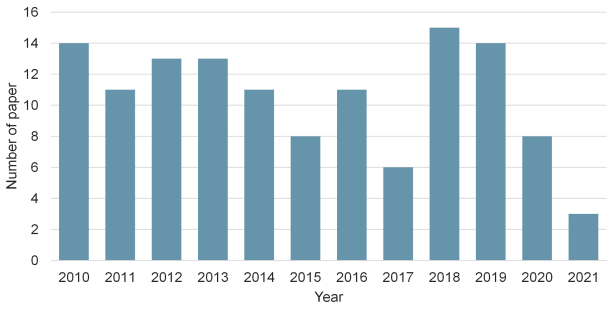

FIGURE 8. Distribution of studies by years of publication.

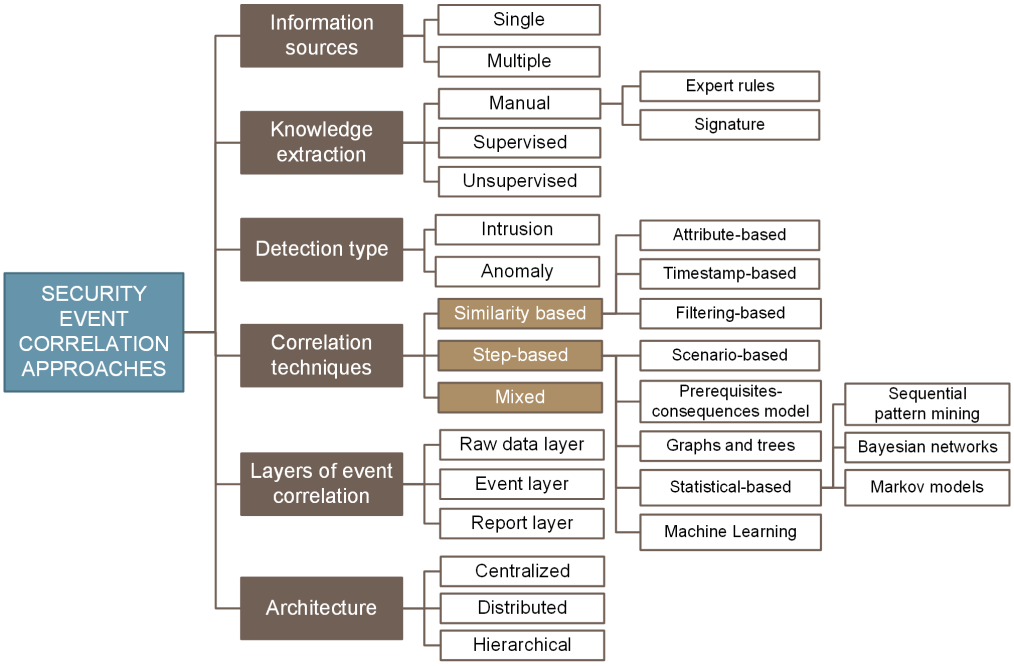

FIGURE 9. Security event correlation approaches classification.

V. CLASSIFICATION OF SECURITY EVENT CORRELATION METHODS

In this section, we present a classification and review of selected publications, in which the authors propose approaches for security event correlation. Figure 9 shows the general classification scheme for correlation approaches. The basis for the purposed classification is the features, identified by researchers in the field of security event correlation, which were considered earlier in the relevant reviews (Table 2).

We define the following main features to classify the approaches and methods of security event correlation:

- **Correlation method.** Security event correlation methods can be divided into three main classes: similarity-based, step-based and mixed. *Similarity-based* methods compare multiple events based on attributes, timestamps, or filters. *Step-based* methods create chains of events, reconstruct an attacker's actions, and analyze connections between several events. This class includes scenario-based methods, prerequisites and consequences models, attack graphs and trees, statistical-based methods and machine learning. *Mixed* methods use a combination of different correlation algorithms.
- **Knowledge extraction**. The approaches to security event correlation can be categorized according to the origin of information about attacks. So, according to the knowledge extraction, we can distinguish *manual, supervised* and *unsupervised* methods. *Manual* methods use knowledge encoded by an expert in the form of rules or attack signatures. *Expert rules* are created by describing the conditions for an attack, for example, using logical expressions, ontologies and other ways. *Signatures* are known and documented attack patterns. Supervised methods use a training dataset containing information about attacks. Unsupervised methods do not use any prior knowledge.
- **Detection type.** Correlation approaches can be classified according to their ability to detect new security issues. It depends on whether the approach explores *intrusion detection* or *anomaly detection*. The first category analyzes events presumably containing specific attack data and does not allow detection of new attacks; this category often is named misuse detection. The second category analyzes deviations from normal behavior, which may indicate a new attack type.
- **Information sources**. The security event data source can be either *single* or *multiple*.
- **Layers of event correlation**. Depending on the stage, alerts can be processed at the *raw data*, *events*, and *report* levels. The raw data layer processes raw sensor and source data. The event layer analyzes IDS alerts and aggregated event data. Report level to carry out post-processing of prioritized and classified events.
- **Architecture** of the security event correlation system can be *centralized*, *distributed* or *hierarchical*. Centralized alert correlation approaches use local data collection using various network agents. Further, alerts are transmitted to the central control server for correlation analysis. Distributed correlation allows each agent to perform partial correlation, which results are aggregated, for that all agents have the same weight. The hierarchical architecture divides management agents into several groups according to various aspects such as geographic location, administrative control, and others.

Correlation levels are allocated similarly to [1], the number of data sources and architecture type is similar to [5]. The classification of correlation methods is based on the generalization of [2], [4], [5], [7]–[9]. The type of attack detection and knowledge extraction is partially mentioned in [1], [8].

In this review, we consider approaches to security event correlation from the selected set of publications in terms of each given feature. The review also takes into account the experimental component of the research. We define existing datasets and how approaches are evaluated. The evaluation describes whether the authors validated their approach on datasets using quantitative performance metrics, reconstruction of attack scenarios, or using examples, use cases, and formal evidence. The metrics used will be discussed in more detail in Section VI.

A. SIMILARITY-BASED METHODS

1) ATTRIBUTE-BASED METHODS

This type of method correlates events based on the similarity of their attributes. Figure 10 shows how this category of methods works. A group of alerts $\{A_1, A_2, A_3 \ldots\}$ is sent to the input, where each i-th alert has a set of attributes $A_i = \{a_{i1}, a_{i2}, \ldots, a_{in}\}$, where n is the number of attributes. A measure of similarity between attributes (sim) can be calculated using Euclidean, Mahalanobis or Manhattan distance functions, Pearson's correlation coefficient, hash functions and other mathematical tools. The obtained correlation coefficients are compared with the threshold value. The selected correlation coefficients are used to determine the similarity of each pair of alerts based on the preset function Φ. This creates a similarity matrix for incoming alerts. The basic principle behind this correlation is that a group of similar alerts can correspond to the same type of attack. Attribute-based correlation is also used effectively to identify similar alerts, which helps reduce the number of events processed for the security administrator.

Different sets of attributes can be used to determine the similarity between events. Most researchers analyze IDS

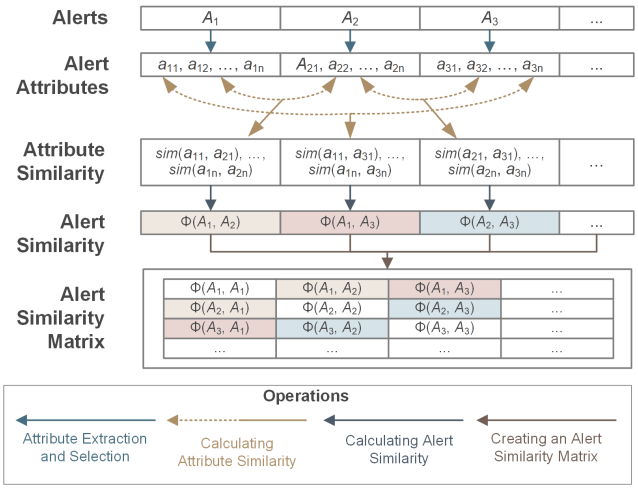

FIGURE 10. Simple attribute-based correlation method.

alerts in terms of signs of network flows, such as IP addresses, ports, protocols, and others. Mohamed *et al.* [19] use the target asset and destination IP address as alert attributes to identify patterns of attacks against specific assets. They combine alert attributes into clusters, and calculate the hash sum of each cluster for later comparison and similarity. Bajtoš *et al.* [20] propose their own formula for combining and correlating security events using attribute values such as IP addresses and ports. Hostiadi *et al.* [21] describe a two-step approach for low and high levels alerts. At a low level, they determine the equality of the attributes of the network flows, and the result is a matrix of similarity values for each alert type. At a high level, the authors measure the similarity of the resulting alert matrices. GhasemiGol and Ghaemi-Bafghi [22], [23] use alert partial entropy. The higher the partial entropy, the higher the probability that alerts indicate the same information. Alerts are grouped using density-based spatial clustering of applications with noise (DBSCAN) [24]. This clustering algorithm examines the density of points: groups closely spaced points, and marks distant points as outliers. Only the specified number of alerts containing the largest amount of information in the cluster is selected.

In the case of cloud environments, attributes are identified using the cloud management portal. So, Meera and Geethakumari [25] perform event correlation in such a way as to group events generated by a specific user or virtual machine using their unique identifiers.

Tan *et al.* [26] propose a multivariate correlation analysis system for attack detection by extracting geometric correlations between characteristics of network traffic. This solution uses anomaly-based detection by examining only legitimate network traffic patterns. The Mahalanobis distance is used to measure the similarity between traffic records, which allows the estimation of multidimensional data objects.

The set of attributes can be determined not only by experts, but also without prior knowledge. So, Liu *et al.* [27] propose a correlation scheme in which unnecessary attributes are removed using rough sets theory. This method allows removing unnecessary attributes by calculating the conditional entropy between them and analyzing the alert sequences to obtain rules in the form of attack behavior patterns. Huang *et al.* [28], [29] introduce an automatic fuzzy logic rule generator to block highly correlated alerts. The module uses attribute adaptation and history rule lifetime to generate temporary blocking rules that can then be applied in the IDS rule database. Attack streams are generated by combining alerts with equivalent source and target attributes that occur over a specific period. Traditionally, security event analysis is provided by default by expert rules, such as rules provided by SIEM systems like OSSIM or Sigma, and programmed rules, for example in IDS Bro. Granadillo *et al.* [30] describe two methods for obtaining attribute-based alert correlation rules. The first method aims to derive correlation rules from the configuration of policy enforcement points (PEPs), such as firewalls, web servers, and others. All PEP attributes are

standardized into a common format by a matching table. The authors describe their function for comparing attributes. The second method aims to derive correlation rules from information security indicators such as the National Vulnerability Database (NVD).

In addition to centralized management, the security event correlation module can have a distributed architecture. Mohamed and Basir. [31] split the managed network into separate management domains, and each domain is assigned an intelligent agent to collect and analyze security events. All agents use a majority rule to determine the root cause of a network failure. To compute the most likely explanation, the agent manager applies the theory of belief functions or Dempster-Shafer theory [32]. This theory makes it possible to combine evidence from different sources and arrive at a degree of belief, in the form of a mathematical object called the belief function. It takes into account all available evidence to calculate the probability of an event.

Rice and Daniels [33] propose a layered hierarchical approach to intrusion detection based on specifications and event correlation at each level of the system. Each individual level of the hierarchy indirectly interacts with neighboring levels, transmitting the results of the analysis to the higher-level system in the form of messages. The event correlation engine compares messages from each layer with messages generated by neighboring layers.

Attribute-based correlation methods are simple and easy to implement, but the ability of these approaches to detect causality between events is very limited. Attribute similarity approaches rely heavily on expert knowledge, and their effectiveness depends on the choice of an appropriate distance function and the threshold value.

2) TIMESTAMP-BASED METHODS

This type of method uses the timestamp of security events as the primary feature of similarity. Finding relationships between security events is based on their temporal relationships, so the correlation occurs within a certain time window.

Wu *et al.* [34] calculate Pearson correlation coefficients between pairs of security event attributes in a time series. The correlation matrix captures the relationships between each pair of event attributes from the start time to the end time. The graphical representation of the current event from the correlation matrix is compared with known events from the signature database to determine the type of the current event.

Bateni and Barani [35] present an ERDTW (Enhanced Random Directed Time Window) alert selection policy based on sliding time windows analysis. ERDTW classifies time intervals into relevant (safe) and irrelevant (dangerous) based on their attributes described in the rules and expressions of mathematical logic. For example, if there are many alerts with the same IP address in a time interval, then it is more likely to be flagged as dangerous.

Raftopoulos and Dimitropoulos [36] describe an Extrusion Detection Guard (EDGe) IDS alert correlator that detects a multi-step malware trail created by infected hosts. The sequence of alerts is discretized over time slots for each host. Next, an entropy-based information theory criterion is used to detect correlated tuples of alerts within a given window. Each tuple is associated with a frequency and a confidence level. The frequency determines the number of occurrences of the first alert. Confidence refers to the proportion of cases where the first alert is followed by another alert within the time window. The ranking of tuples is based on the union of these metrics.

In addition to the specified time windows, an indicator of the time delay between security events can be used. Kotenko *et al.* [37] propose an approach to security event correlation, which reveals the relationships between the types of security events depending on the distribution of events over time. The delay between two events is considered as one of the main indicators and is included in the calculation of the Pearson correlation between these events.

Timestamp-based correlation can significantly reduce the number of alerts and aggregate them into high-level alerts. But its application is limited due to the high determinism in the analysis of events and the dependence on the size of time windows. Since the security event is constantly evolving, the number of selected time steps may not be sufficient to reflect the entire period of the event, resulting in incomplete measurement data.

3) FILTERING-BASED METHODS

Filter-based correlation methods are also used to reduce irrelevant security events. Certain filtering algorithms assign priority to various events according to their criticality for the security of the system or network. Elshoush *et al.* [38] propose an alert correlation model that reduces the number of alerts processed by discarding irrelevant and false alerts in the early stages. The filtering component uses the asset database, identifies high and low-risk alerts, and the low-risk alerts are discarded. Anuar *et al.* [39] propose a security event prioritization model based on a risk assessment and hierarchy analysis process. The metrics criticality, maintainability, replaceability, and dependability are used for assessment.

Filtering algorithms can also be applied to truncate IDS alerts based on the validation of alerts by security operators. The event filtering logic FO-MQCL (First Order – Minimal Qualitative Choice Logic) by Benferhat and Sedki [40] only displays alerts that match the operator's qualifications.

The applied filtering algorithms are highly dependent on the configuration of the protected system or the network topology used. Consequently, this category of approaches has low dynamics and high deployment costs. There is also a high dependence on the described correlation rules. For these reasons, the use of filtering algorithms in security event correlation systems has been declined significantly in recent years, giving way to more adaptive approaches. Also, this category of methods can be called of little interest to a modern researcher, since this relatively well-studied area.

TABLE 3. Similarity-based methods.

№	Paper	Year	Correlation method	Knowledge extraction	Detection type	Info. source	Corr. layer	Architecture	Dataset	Evaluation
1.	Wu et al. [34]	2010	Timestamp-based	Expert rules	Intrusion	Single	Event	Central.	Generated	Recon.
2.	Mohamed and Basi [31]	2010	Attribute-based	Expert rules	Anomaly	Multiple	Report	Distrib.	Generated	Formal
3.	Benferhat and Sedki [40]	2010	Filtering-based	Expert rules	Intrusion	Multiple	Report	Central.	DARPA 1999	Metric
4.	Huang et al. [28]	2010	Attribute-based	Unsupervised	Intrusion	Multiple	Event	Central.	DEFCON 9, DARPA 2000, Treasure Hunt	Metric
5.	Huang et al. [29]	2012	Attribute-based	Unsupervised	Intrusion	Multiple	Event	Central.	DEFCON 9, DARPA 2000, Treasure Hunt	Metric
6.	Anuar et al. [39]	2011	Filtering-based	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Formal
7.	Liu et al. [27]	2012	Attribute-based	Unsupervised	Intrusion	Multiple	Raw	Central.	Generated	Metric
8.	Rice and Daniels [33]	2012	Attribute-based	Expert rules	Intrusion	Multiple	Event	Hierarch.	Generated	Formal
9.	Elshoush et al. [38]	2012	Filtering-based	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
10.	Mohamed et al. [19]	2012	Attribute-based	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Formal
11.	GhasemiGol and Ghaemi-Bafghi [22]	2013	Attribute-based	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Recon.
12.	GhasemiGol and Ghaemi-Bafghi [23]	2015	Attribute-based	Expert rules	Intrusion	Multiple	Raw		DARPA 2000	Metric
13.	Bateni and Baraani [35]	2013	Timestamp-based	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000, netForensics, honeynet, Generated	Metric
14.	Tan et al. [26]	2013	Attribute-based	Expert rules	Anomaly	Single	Event	Central.	KDD Cup 99	Metric
15.	Raftopoulos and Dimitropoulos [36]	2014	Timestamp-based	Expert rules	Intrusion	Multiple	Event	Distrib.	Generated	Formal
16.	Meera and. Geethakumari [25]	2016	Attribute-based	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Formal
17.	Granadillo et al. [30]	2016	Attribute-based	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Formal
18.	Kotenko et al. [37]	2018	Timestamp-based	Unsupervised	Intrusion	Multiple	Raw	Central.	Generated	Formal
19.	Hostiadi et al. [21]	2019	Attribute-based	Expert rules	Intrusion	Single	Event	Central.	Generated	Metric
20.	Bajtoš et al. [20]	2020	Attribute-based	Expert rules	Intrusion	Multiple	Raw	Central.	4SICS Geek Lounge	Recon.

Note: Here and below, the following abbreviations are allowed in the tables: *Corr.* - correlation, *Info.* - information, *Central.* - centralized, *Distrib.* - distributed, *Hierarch.* - hierarchical, *Recon.* - reconstruction.

4) SUMMARY TABLE OF SIMILARITY-BASED METHODS

Table 3 presents the described similarity-based correlation approaches. Here and below, approaches to the security event correlation are considered from the point of view of the distinguished features of the classification: correlation method, knowledge extraction, detection type, information sources, correlation layer and architecture (see Figure 9). We also determine the datasets used and the method of evaluation based on quantitative performance metrics (*Metric*), reconstruction of attack scenarios (*Recon.*), or using examples, use cases, and formal evidence (*Formal*).

Similarity-based correlation approaches often require knowledge to select a threshold or filtering algorithm, which demands the involvement of experts. But at the same time, we can see an increase in the use of methods based on automatic rule generation. Filtering-based approaches, as the least adaptive, are less commonly used in current research, while methods based on similarity of attributes remain relevant due to their ease of implementation and reliability.

B. STEP-BASED METHODS

1) SCENARIO-BASED METHODS

Scenario-based approaches correlate security events based on specific attack scenarios defined by signatures or expert rules. The main application of these approaches is to detect individual stages of multi-step attacks and the sequence of stages. The degree of similarity between events belonging to the same attack scenario should be higher than the degree of similarity between events from different scenarios. Figure 11 provides representations of the multi-step attack signatures that are used in the studies discussed in this paper, as well as methods for comparing alerts.

Various correlation functions of such attributes as the attacker's IP addresses and port numbers can be used to compare the current state of the system with the attack signature, as shown by Wang *et al.* [41].

Cheng *et al.* [42] present the development of the JEAN (Judge Evaluation of Attack intensioN) approach to validate network security alerts. This approach allows predicting a multi-step attack by measuring metrics of the difference between the current session and the attack session. Sessions are represented as two ordered sets of coordinates, and their coincidence is measured by the sum of the weighted squares of the distances between the corresponding coordinate pairs.

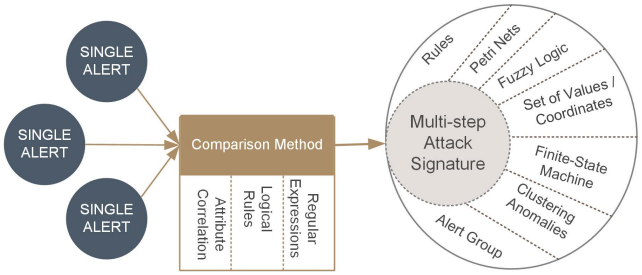

FIGURE 11. Ways to represent signatures of multi-step attacks.

Comparisons between normal and abnormal sessions are also carried out by Das *et al.* [43]. The authors describe detection algorithms for each of three attack scenarios: flooding App-DoS, shrew flooding App-DoS, and flash crowds App-DoS. Díaz López *et al.* [44] present three attack scenarios for the Internet of Things (IoT) networks, based on the main weaknesses of the IoT network, and also constitute the correlation rules used by the SIEM system.

Liu *et al.* [45] group security events based on nondeterministic finite-state machine (FSM). The authors propose three types of attack scenario presentation: process-critical, attacker-critical, and victim-critical. In the process-critical scenario, FSMs are used to simulate an intrusion process between one attacker and one victim. The attacker-critical scenario recovers the intrusion process against the entire target network. The victim-critical scenario simulates intrusion actions for a specific system.

Attack patterns can also be described using ontologies and formal grammars [46], logical rules [47], signature languages and Petri nets [48]. The ontology is based on a hierarchy of concepts that define the actions of intruders to implement attacks of various classes with varying degrees of detail [46]. Besides, attacks are represented by sequences of characters that can be considered as ''words'' of a formal language, specified by some formal grammar [46]. Sadighian *et al.* [47] consider a context-aware and ontology-based alert correlation framework ONTIDS. The proposed ontology contains templates for the basic classes of IT assets, alerts, vulnerabilities, and attacks. The correlation mechanism is implemented using logical rules written in Semantic Web Rule Language (SWRL) [49] and Semantic Query-Enhanced Web Rule Language (SQWRL) [50] based on OWL (Web Ontology Language) description logic (OWL-DL) [51], which uses the model of description ''object-property''. Jaeger *et al.* [52] offer a real-time event analysis and monitoring system (REAMS) to detect multi-step attacks. It is based on analysis of event logs using EDL (Event Description Language) signature language. EDL [53] represents attacks in a colored Petri net, where nodes set the state of the system, and transitions between them are based on current system events. REAMS is based on normalization rules to match and detect the type of the event, and then to extract the event fields from specific groups in a regular expression. Ussath *et al.* [54] propose a method for automatically deriving multi-stage EDL signatures from tagged data graphs.

Using full attack signatures may not be enough to identify variations in different attacks. For this case, Herrerías and Gómez [55] divide security event correlation in logs into two parts: automated detection of atomic attacks from the knowledge base and their use to construct complex multi-step attacks scenarios in real-time. The attacker's actions are linked based on the source IP address and timestamp. Zhang *et al.* [56] propose Intrusion Action Based Correlation Framework (IACF) to detect multi-stage attack scenarios. Alerts are grouped according to the source IP address and the destination IP address. Intrusion actions are extracted from each group of alerts based on intrinsic correlation or atomicity: one attack class can fire the same set of alerts regardless of how they are ordered. The IACF can split a long sequence of actions into different sessions by calculating the time limit for each subsequence. The framework reduces sessions using a sequence-pruning algorithm to remove redundant actions and splits sessions into binary correlations. Binary correlations are then added to correlation graphs for intrusion prediction.

Transitions between system states in attack scenarios can be defined using a fuzzy declarative language. Almseidin *et al.* [57] present a multi-step attack detection mechanism based on a fuzzy automaton. This mechanism allows the use of an incompletely defined rule base for transition between events. In the fuzzy automaton, the system states, inputs, and outputs are represented as fuzzy sets. In the beginning, the initial normal states of the system and possible final states are configured. Possible input values are determined by the attack types in the form of event sets, and transitions from one state to another are described by fuzzy rules.

Attack patterns can be obtained by anomaly analysis. Landauer *et al.* [58] describe an approach that automatically extracts cyber threat information from raw data, creating patterns of complex system behavior. The authors conduct training on log data without anomalies, and use the found deviations in the alert sequences for clustering and recovering attack patterns. Shin *et al.* [59] introduce an approach that automatically generates multi-step attack detection rules based on suspicious traffic. The network flows analysis module clusters the dominant patterns of activity and creates rules for detecting multi-step attacks.

Attacks dispersed across cyberspace can be detected by distributed correlation systems. For this, Nie *et al.* [60] demonstrate an architecture for unlimited network scanning to detect attack scenarios, represented as tuples of attack nodes, directed edges, and their constraints. Control centers in a P2P network manage security agents, and security agents control some peripherals. Each security agent can be used as a task coordinator to correlate security events. Agents are united in groups according to similar attack scenarios. Bhatt *et al.* [61] use Intrusion Kill-Chain (IKC) attack models as a basis for modelling attacker behavior. IKC is a seven-stage model that an attacker inevitably follows during an intrusion: information gathering, weaponizations, delivery, exploitation, installation, command, and control (C2), actions [62]. The authors propose a distributed sensor system to detect each of these stages and search for known attack patterns. The authors propose a system with distributed detection sensors for each of these stages. The sensors are triggered by matching rules with known patterns of malicious behavior.

The described scenario-based correlation approaches are presented in Table 4. Scripting approaches are most effective for known attack detection and have remained relevant over the past decade. But at the same time, both the support of the attack scenario database and the search for new templates can require significant additional resources.

TABLE 4. Scenario-based methods.

№	Paper	Year	Knowledge extraction	Detection type	Info. source	Corr. layer	Architecture	Dataset	Evaluation
21.	Herrerías and Gómez [55]	2010	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Recon.
22.	Liu et al. [45]	2010	Expert rules	Intrusion	Single	Event	Central.	Generated	Recon.
23.	Wang et al. [41]	2010	Signature	Intrusion	Single	Event	Central.	DARPA 2000	Formal
24.	Nie et al. [60]	2011	Signature	Intrusion	Multiple	Event	Distrib.	DEFCON 11	Recon.
25.	Cheng et al. [42]	2011	Signature	Intrusion	Single	Event	Central.	DARPA 2000, DARPA CGC	Formal
26.	Das et al. [43]	2011	Expert rules	Intrusion	Multiple	Raw	Central.	KDD Cup 99, LBNL	Metric
27.	Sadighian et al. [47]	2013	Expert rules	Intrusion	Multiple	Raw	Central.	ISCXIDS2012, DARPA 2000	Metric
28.	Bhatt et al. [61]	2014	Expert rules	Intrusion	Multiple	Event	Distrib.	Generated	Recon.
29.	Jaeger et al. [52]	2015	Signature	Intrusion	Multiple	Event	Central.	Generated	Formal
30.	Ussat et al. [54]	2016	Signature	Intrusion	Multiple	Event	Central.	Generated	Formal
31.	Díaz López et al. [44]	2018	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Recon.
32.	Almseidin et al. [57]	2019	Signature	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
33.	Zhang et al. [56]	2019	Expert rules	Intrusion	Multiple	Raw	Central.	DARPA 2000, CICIDS2017	Metric
34.	Landauer et al. [58]	2019	Signature	Anomaly	Multiple	Raw	Central.	Generated	Metric
35.	Shin et al. [59]	2019	Expert rules	Anomaly	Single	Event	Central.	DARPA 2000, CTU-13	Metric

2) PREREQUISITES AND CONSEQUENCES MODEL

In contrast to the above approaches, methods using prerequisites and consequences allow one to more effectively solve the problem of multi-step attack detection in the absence of specified signatures.

This category of methods compares and links security events to each other on the basis of causality: the consequences of previously executed stages of attacks are compared with the prerequisites for subsequent stages of attacks. Events are correlated if the consequence of one event matches the prerequisite of another (Figure 12).

Relationships between events can be obtained with or without a knowledge base of prerequisites and consequences. A set of defined criteria is used to study the causal relationship between alerts. Modeling prerequisites and consequences can be based on logical expressions, modeling languages, etc. For example, Xuewei *et al.* [63] reconstruct an attack scenario based on a finite-state machine that combines cluster analysis and causal analysis for synchronous security event processing and intrusion process rebuilding. Each FSM node corresponds to an attack scenario rule, which is described by a number of attributes.

Ficco and Romano [64], [65] present a distributed approach to the complex correlation of events and intrusion symptoms. Attack symptoms are classified based on the concepts presented in the author's ontology. The attack scenario is modeled by the causal correlation rule in the form of a set of logical conditions for intermediate attack signals, which are interconnected by time constraints. Based on the developed approach, the authors also implement the intrusion detection and diagnosis system (ID2S) [66], which consists of hierarchically organized logical objects: probes, agents, decision engine, adjudicator, remediators, and monitors. Lin *et al.* [67] describe a real-time intrusion alert correlation system based on prerequisite and consequence (REAC), consisting of distributed agents and a manager. The prerequisites and consequences of alerts in the form of logical predicates are stored in the knowledge base.

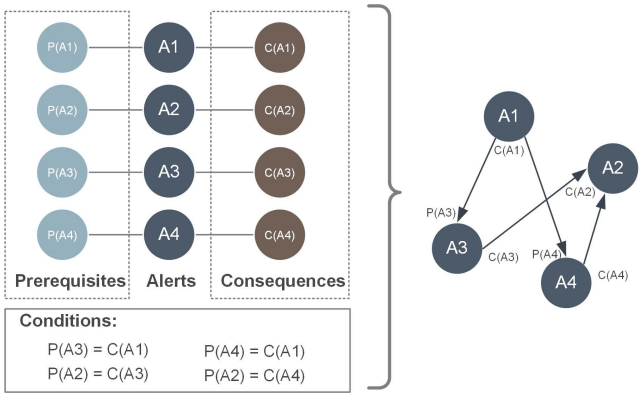

FIGURE 12. Simple prerequisites and consequences correlation model.

Some of the prerequisites-consequences correlation approaches include components that operate in two modes: offline and online. The offline component analyzes historical data and generates rules for security event correlation on a causal basis, while the online component correlates incoming alerts using the created rules and models. Alserhani *et al.* [69], [70] offer multi-stage attack recognition system (MARS). The proposed approach is based on an improved ''requires/provides'' causal model [79] with the addition of two additional consequences parameters: vulnerability and extensional consequences. This model establishes a correspondence between the conditions ''require'' and ''provide'' for elements of attacks. The online component of the system aggregates the raw alerts hyper-alert, while the offline component provides attack detection rules for matching the hyper alert. The authors subsequently demonstrate the capabilities of MARS to detect botnet attacks [68]. Alnas *et al.* [71] use a similar ''requires/provides'' model to recognize botnet attacks, adding new parameters: in addition to vulnerability and extensional conditions, they introduced attack direction and administrator experience.

Saad and Traore [72] propose an attack scenario reconstruction technique that groups semantically relevant IDS alerts into clusters representing possible attack scenarios. The technique identifies relevant alerts and encodes causal information for an attack using an intrusion ontology.

TABLE 5. Prerequisites and consequences model.

№	Paper	Year	Knowledge extraction	Detection type	Info. source	Corr. layer	Architecture	Dataset	Evaluation
36.	Xuewei et al. [63]	2010	Unsupervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
37.	Ficco [65]	2010	Expert rules	Intrusion	Multiple	Event	Distrib.	Generated	Metric
38.	Ficco and Romano [64]	2010	Expert rules	Intrusion	Multiple	Event	Distrib.	Generated	Metric
39.	Ficco and Romano [66]	2011	Expert rules	Intrusion	Multiple	Event	Distrib.	Generated	Metric
40.	Lin et al. [67]	2010	Expert rules	Intrusion	Multiple	Event	Hierarch.	Generated	Metric
41.	Alserhani et al. [68]	2010	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
42.	Alserhani et al. [69]	2010	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
43.	Alserhani [70]	2013	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Recon.
44.	Alnas et al. [71]	2013	Expert rules	Intrusion	Single	Event	Central.	DARPA 2000, Generated	Recon.
45.	Saad and Traore [72]	2013	Expert rules	Intrusion	Multiple	Raw	Central.	DARPA 2000, Treasure Hunt	Metric
46.	Friedberg et al. [73]	2015	Expert rules	Anomaly	Multiple	Event	Central.	Generated	Metric
47.	Liu et al. [74]	2018	Expert rules	Anomaly	Multiple	Event	Central.	Generated	Metric
48.	Barzegar and Shajari [75]	2018	Expert rules	Intrusion	Multiple	Event	Distrib.	DARPA 2000, MACCDC 2012	Metric
49.	Khosravi and Ladani [76]	2020	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Metric
50.	Mahdavi et al. [77]	2020	Expert rules	Intrusion	Single	Event	Central.	DARPA 2000	Formal
51.	Hu et al. [78]	2020	Signature	Intrusion	Multiple	Event	Central.	DEFCON 23, Generated	Recon.

Also, this technique can detect false negative results using prerequisites-consequences information and information about the environment. Similarly, Barzegar and Shajari [75] reconstruct the attack based on the ontology of alerts and search for semantic similarities between them. The most similar alerts have a causal relationship. Hu *et al.* [78] analyze both successful and unsuccessful paths of an attacker, which are grouped into different attack scenarios for comparison with future alerts.

A number of approaches are based on the use of relations of prerequisites and consequences when the system deviates from normal behavior. Friedberg *et al.* [73] use normal system behavior or whitelisting to detect incidents. Their approach analyzes log lines and combinations of information from system logs to determine hypotheses. These hypotheses describe the possible consequences based on the classification of log lines. Events not related to the hypothesis are considered anomalous.

Liu *et al.* [74] propose a tool for tracking and prioritizing causal relationships in events. The authors consider the rarity of a system event and its topological features in the causality graph to assess the priority of this event. The developed reference model records common routines in computer systems. The rarity of each event is quantified to distinguish unusual operations from normal system events.

Khosravi and Ladani [76] describe an approach to real-time APT-based attack detection. The authors conduct a causal analysis of meta-alerts generated by both security sensors and other system recorders. The proposed approach calculates the sensitivity to probable APT-based attacks for each host through dynamic programming that works with alerts from each host separately and conducts a long-term analysis of the attack process. The attack steps are linked if the following conditions are met: (1) prerequisites-consequences alerts occur on the same host; (2) prerequisite alert and consequence alert belong to the same class of events; (3) a prerequisite alert precedes a time consequence alert.

Another way to recover attack scenarios is to use a codebook. The codebook encodes the relationships between security events and their consequences as a matrix of codes. All alerts are grouped into vectors that are mapped to the consequences of security issues and stored in a codebook. Often, this matrix is binary where "1" indicates a causal relationship between the event and the security issue, and "0" indicates no relationship. Mahdavi *et al.* [77] propose a real-time alert correlation method based on code-books (RACC) in which codebooks correspond to attack scenarios. In the offline phase, the method extracts scenarios from the knowledge base and creates indexed codebooks, while in the online phase it accesses the books and performs correlation using matrix operations. Correlation with the codebook quickly identifies the root cause of a security violation. However, dynamically changing the structure of the protected system or network requires a costly update of the codebook. There is also a high dependence on expert knowledge.

The described correlation approaches using the prerequisites and consequences model are presented in Table 5. Prerequisite and consequence methods are more flexible and extensible than scenario-based methods, as they tend to require less initial knowledge of attacks. At the same time, the resource, and computing power requirements increase. Creating a knowledge base that describes each action with its prerequisites and consequences is very labor-intensive. When a previously unknown event appears, it is necessary to check its connections with all existing ones.

3) ATTACK GRAPHS AND TREES

Often, step-based security event correlation approaches use models to specify attacks or network security situations, such as graphical attack evolution models or game-theoretic representation of the interaction between an attacker and a defense system. The most common category of such models are graphs and attack trees. In graph theory [81], a tree is an acyclic graph, in which any two vertices are connected by exactly one path, and the graph is a cyclic graph. At the same time, in the reviewed papers, attack graphs are visually represented by both cyclic and acyclic graphs. Therefore, we need to take into account that the representation of attack

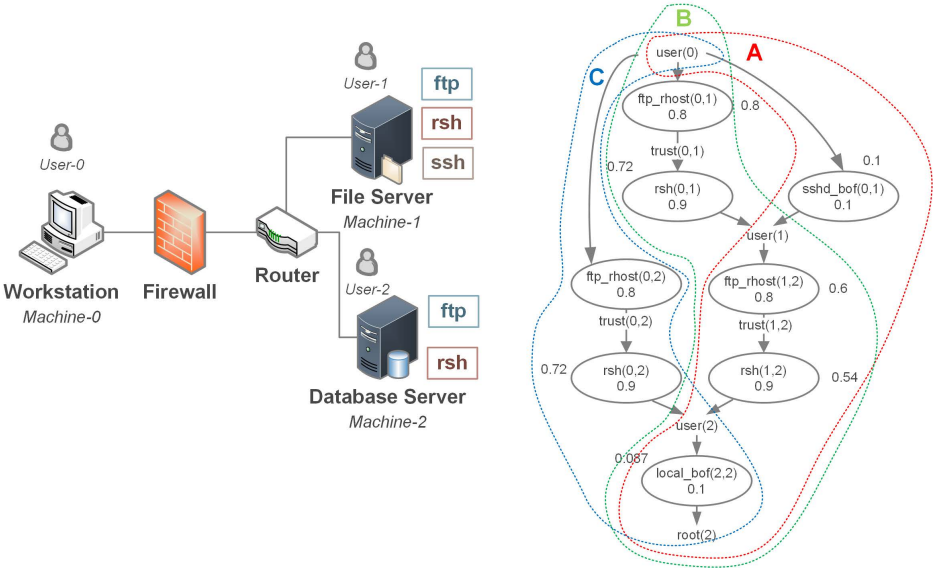

FIGURE 13. Example of network configuration and attack graph *Source*: Adapted from [80].

graphs in security research does not always correspond to the definition of a graph in graph theory [82]. We can establish that attack trees have only one entry point, which is the root of the tree, and there are no closed chains of steps or cycles. At the same time, attack graphs can have one or more input points, and closed chains may or may not be present.

Correlation methods based on graphs and attack trees represent sequential information about security events in the form of directed graphs, where vertices (also called nodes) are alerts or attack stages, and edges are relationships between them. Transitions in such a graph represent attacking actions with some weights, such as probability or criticality. A common way to represent attack graphs is to model the network topology with vulnerabilities for each node.

Figure 13 shows an example of an attack graph described in [80]. The left part of the figure shows the network configuration: machine-1 is a file server that offers file transfer (ftp), secure shell (ssh), and remote shell (rsh) services, machine-2 is an internal database server with ftp and rsh services. The right part of the figure shows an attack graph with two types of vertices: exploits in ovals and conditions in the form of labels without ovals. For example, rsh(0,1) represents a remote shell login from machine-0 to machine-1, and trust(0,1) means that a trust relationship is established from machine-0 to machine-1. Numeric values represent the probability that the exploit will be executed under all required conditions. The figure shows three attack paths (A, B, C) indicated by colored lines.

We will briefly describe attack A as an example (B and C are performed similarly). The attack begins with a ssh buffer overflow from machine-0 to machine-1 (sshd_bof(0,1)), which gives the attacker the ability to execute arbitrary codes on machine-1 as a normal user (user(1)). The attacker exploits the ftp vulnerability on machine-2 (ftp_rhost(1,2)) to download a list of trusted hosts. This trust relationship allows an attacker to remotely execute shell commands on machine-2 without a password. A local buffer overflow is performed on machine-2 (local_bof(2,2)), which increases the attacker privilege to the root server of that machine.

There are two general types of approaches to analyzing attack graphs and trees. The first one includes search technologies to identify possible attack paths and attacker actions. So Roschke *et al.* [83] match alerts to graph nodes and search for suspicious subsets using the Floyd-Warshall algorithm [84], [85] to find the shortest paths in the attack graph. The total weights of the shortest paths between all pairs of vertices are calculated for one execution of the algorithm. At the preparation stage, information about the system is collected, a database with alert classifications is imported, and a graph for the network is loaded. The second type of attack graph analysis approach uses statistical methods, such as sequence frequency analysis, to try to find the relationships of attacks over time. Jinghu *et al.* [86] build attack graphs based on data mining. They first transform the sequence of events into a directed acyclic graph, then look for frequent graphical patterns that define the characteristics of the attack pattern. Shittu *et al.* [87] for a comprehensive system for analyzing intrusion alerts (ACSAnIA) use a frequent pattern analysis algorithm to extract common patterns from each alert cluster. The approach by Navarro-Lara *et al.* [88] is based on Ant Colony Optimization to link sequences of events that could lead to an attack. Agents, like ants, traverse a set of trees representing an attacker's sequence of actions. Additionally, an expert check is introduced to train the system.

The attack graph, which provides information about the progress of an attack, is also often complemented by a service dependency graph that demonstrates the propagation of the attack's impact from a compromised service. The attacker's

actions, in this case, consist of sequential compromise of nodes using their known vulnerabilities. Hossain *et al.* [89] present a real-time approach to reconstructing attack scenarios based on dependency graph analysis of audit log data. The graph vertices represent subjects (processes) and objects (files, sockets), and the edges specify audit events (for example, operations such as read, write, execute, and connect). The approach detects attacks using tags assigned to events based on knowledge. Albanese and Jajodia [90] combine the attack graph and the dependency graph to display all explicit and implicit dependencies between services and hosts. The edges of the joint graph show which services are directly affected by a successful vulnerability exploit, and what losses are caused by this exploit. This helps to calculate the impact of current attacks and assess future impacts. Shameli-Sendi *et al.* [91] propose a model combining attack graphs and service dependency graphs based on LAMBDA functions. These functions determine the attacker knowledge level and the attack impact on security attributes CIA (confidentiality, integrity, and availability). Thus, the attack graph provides accurate information about the attack progress, impact on CIA and the attacker's knowledge. In turn, the service dependency graph demonstrates the propagation of the attack impact from the compromised service.

Angelini *et al.* [92], in turn, purpose an On-line Correlation Engine (OCE), that checks whether the incoming alert is a part of any attack graph and evaluates the next access point. OCE uses two independent metrics to assess the similarity between the current alert and the attack path: Jaccard similarity and cosine similarity. Jaccard Similarity or Jaccard index [93] is defined as the sample intersection size divided by the sample union size. This index allows comparing the set of the attack graph edges and the set of incoming alerts. Cosine similarity is a metric that measures the proximity (distance) of two vectors in space, equal to the cosine of the angle between them. This metric determines the similarity of attack vectors and alerts. Kawakani *et al.* [94] also use the Jaccard index, but to determine the similarity between several attack graphs for further clustering them. The authors calculate the similarity index both for all nodes and for a pair of nodes in the form of edges. Then the average of these two results determines the similarity between the attack graphs.

Attack graphs and trees can also be created offline as baseline models, which are then used in real-time to correlate with incoming alerts. Zali *et al.* [96] in their approach represent the knowledge base of attack patterns in the form of causal relation graphs (CRG). In offline mode, trees are built with probable alert correlations. Correlation is searched online for each received alert in the corresponding tree of previously received alerts. Chien and Ho [95] describe an approach in which an attack plan is created for a specific net in the form of a colored Petri net. The approach allows generating a coverability tree for a security situation assessment and an attack scenario prediction. Attack detection is based on the exploit reliability metric. Kotenko and Chechulin [97] purpose cyber-attack modeling and impact assessment component (CAMIAC) for attack graphs and trees creation, real-time event analysis, future attacker actions prediction, and attack impact assessment. Basic attack graphs are generated for various models of potential attackers. At the same time, a set of tuples is formed for each host of the protected network, including attack action class, access type and attacker knowledge level. The distributed monitoring system tracks events in real-time, correlating them with the basic graphs and updating them as necessary. Godefroy *et al.* [99] automatically generate correlation rules for the attack tree based on previously collected statistics that reflect the relationships between an attacker's actions and potential attack paths.

Security event graphs can be analyzed in terms of looking for abnormal data. Parkinson *et al.* [100] introduce GraphBAD (Graph-Based security Anomaly Detection) for detecting abnormal events without requiring prior knowledge of data structure. GraphBAD converts security configuration data and audit logs into a graphical model. This model is then analyzed to identify abnormal (irregular) subgraphs that may indicate security issues. Melo and Macedo [101] present a distributed agent-based multi-step attack detection system that works with attack graph correlation to reduce false alarm rates to counter-attacks. Agents detect normal and abnormal data in network traffic. When anomalies are detected, each alert is displayed only according to the source and destination data. As a consequence, each alert triggered by an anomalous pattern can be displayed as an attack graph. Marvasti *et al.* [98] propose a data-independent approach to automatically determine the root causes of complex IT systems disruptions based on correlating data from abnormal events. The approach performs statistical processing of virtual directed graphs created based on historical anomalies with probabilistic correlations.

Attack chains can be formed from a graph of correlated alerts by cutting off unrelated events. Zhang *et al.* [102] use negative correlation clipping and non-cascading event noise removal for this purpose. Non-cascading events occur at the same time interval independently of each other, which means they have no causal relationships and can be deleted.

The described approaches to correlation using attack graphs and trees are presented in Table 6. Methods based on attack graphs and trees are popular, especially for visualizing attack scenarios. We can note that these methods are the most used for analyzing generated datasets. The advantage of correlation based on graphs and trees is simplicity, and analysis of graph models allows one to create relationships and associations between events. The ease of updating graphs is also their advantage: elements can be added and removed independently of each other. The main disadvantage of the graph-based methods is data limitation. The graph-based correlation method is not very suitable for analyzing data that cannot be represented in the form of a structure. Implicit and non-linear relationships between events also complicate analysis using graphs and trees. The processing of large complex amounts of data requires more computing resources to maintain the entire graph connections.

TABLE 6. Attack graphs and trees.

№	Paper	Year	Knowledge extraction	Detection type	Info. source	Corr. layer	Architecture	Dataset	Evaluation	
52.	Roschke et al. [83]	2011	Attack graph	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Metric
53.	Chien and Ho [95]	2012	Attack tree	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Recon.
54.	Jinghu et al. [86]	2012	Attack graph	Supervised	Intrusion	Multiple	Event	Central.	DARPA 1999, DARPA 2000	Formal
55.	Zali et al. [96]	2012	Attack graph	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Recon.
56.	Kotenko and Chechulin [97]	2013	Attack graph	Signature	Intrusion	Multiple	Event	Distrib.	Generated	Recon.
57.	Marvasti et al. [98]	2013	Attack graph	Unsupervised	Anomaly	Single	Event	Central.	Generated	Formal
58.	Godefroy et al. [99]	2014	Attack tree	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Recon.
59.	Shittu et al. [87]	2015	Attack graph	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Metric
60.	Navarro-Laraet al. [88]	2016	Attack tree	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Metric
61.	Kawakani et al. [94]	2016	Attack graph	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Formal
62.	Hossain et al. [89]	2017	Attack graph	Expert rules	Intrusion	Single	Event	Central.	Generated	Recon.
63.	Parkinson et al. [100]	2018	Attack graph	Unsupervised	Anomaly	Single	Raw	Central.	KDD dataset, Generated	Metric
64.	Albanese and Jajodia [90]	2018	Attack graph	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Formal
65.	Angelini et al. [92]	2018	Attack graph	Signature	Intrusion	Multiple	Event	Central.	Generated	Metric
66.	Melo and Macedo [101]	2019	Attack graph	Signature	Anomaly	Single	Event	Distrib.	Generated	Metric
67.	Zhang et al. [102]	2020	Attack graph	Expert rules	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric

4) STATISTICAL-BASED METHODS

Statistical correlation methods conclude the distribution of historical data by analyzing sequences of security events in terms of probabilities. Among the popular statistical-based security event correlation methods are probabilistic graphical models (PGMs) in which dependencies between random variables are represented as a graph.

Bayesian networks are structures containing directed acyclic graphs, where vertices correspond to events and edges meet to relationships between them. Another component is quantitative assessment of relationships based on the conditional probability distributions of each node in the context of its parents. So, the Bayesian network consists of the probabilities of the parent node state and a set of conditional probability tables of the child nodes. Figure 14 shows an example of a simple Bayesian attack graph [103] that simulates the activity of an attacker (D) who can use one of the buffer overflows exploits (B, C) to gain access to a server (A). Probability tables are attached to each node. The first two columns (if there are two parents) indicate the necessary conditions for reaching a given node, encoding the execution of parent nodes 0 and 1 (if there is one parent, then one column is used). The other two columns show the probability of execution of a given node $Pr(x)$ or negative probability $Pr(x)$, where x is the designation of the node. This model can be used to calculate the probability of a certain security violation or an attacker's action. For example, Anbarestani *et al.* [104] classify log alerts based on the observed intrusion target using a Bayesian network that extracts the most likely attack strategies.

Statistical approaches can be performed offline and online. The statistical model is trained offline and then used in real-time to correlate security events. In the approach by Ren *et al.* [105] the autonomous system for selection of Bayesian correlation features is used to determine the appropriate features between two alert types and store this information in the correlation and reference tables. The online alert correlation system is designed to determine the relationships between alerts and build attack scenarios on the fly based on the correlation reference table. Wang and Yang [106] implement their algorithm based on [105]. In their approach, the Bayesian network not only calculates the probabilities of relationships between alerts, but also determines the relevance function of alerts. Kavousi and Akbari [107], [108] use a Bayesian network to automatically determine correlation rules in the offline component of their system. This component compares security alerts with accumulated statistics and extracts suspected attack patterns. The online component correlates alerts in real-time, combining them into hyper alerts based on the knowledge obtained by the offline component.

Ramaki *et al.* [109] also purpose a Bayesian network system that operates in two modes. Alerts are merged offline based on similarities in meta-alerts, and the probabilities of transitions between alerts are the basis for creating a

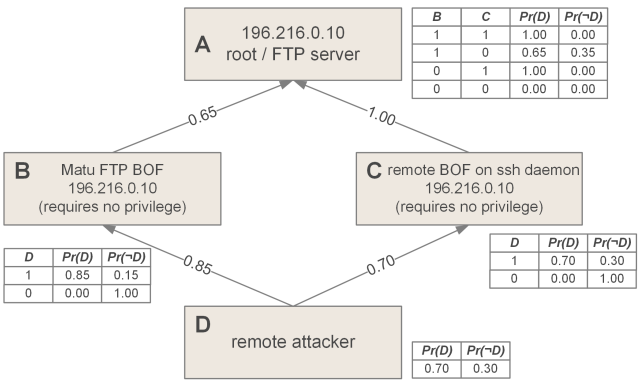

FIGURE 14. Simple Bayesian attack graph *Source*: Adapted from [103].

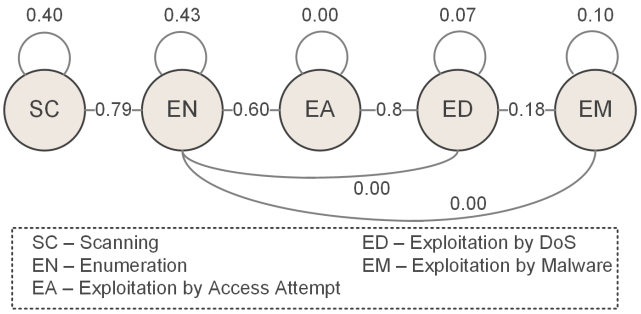

FIGURE 15. Simple Markov model for multi-step attack *Source*: Adapted from [112].

Bayesian attack graph. The most likely next steps of the attacker are predicted online, depending on the incoming alerts, and the values of the probabilities of transitions between the meta-alerts are updated. Based on this system, Pivarníková *et al.* [110] describe a solution that uses a Bayesian network to predict the progress of attacks in the early stages, calculating the probability of a certain alert based on Bayesian inference.

Marchetti *et al.* [111] use the pseudo-Bayesian correlation algorithm to identify correlations between alerts that belong to the same multi-step attack scenario. First, a pseudo-Bayesian probability is determined using Bayes' theorem. This probability reflects the probability of correlating alerts based on their type and timestamps. Correlation results are converted into a weighted graph with nodes representing alerts and edges corresponding to correlation probabilities. The authors then apply a dynamic thresholding algorithm to reduce the graph and remove low-weight vertices.

Markov models describe the probabilities of transition between security events. These models rely on the following property: the subsequent state of the system depends only on the current state, and not on the entire sequence of previous events. The probabilities of state transitions are determined statically or by training on a dataset. In the context of alert correlation, the transition probability is the conditional probability that the next attack step will be executed after the previous step has been completed.

Figure 15 shows an example of the Markov model presented in [112]. This model includes the following stages of a multi-step attack: scanning, enumeration, exploitation by access attempt, exploitation by malware attempt, and exploitation by denial of service. The edges between the attack stages encode the relationships of probability dependence between the variables, and, unlike Bayesian networks, are not directional and can be represented by cycles.

As a rule, events in one attack scenario are always related to each other in the allocation of addresses: the node of the first step of the attack can be the source node of the second step of the attack. For this reason, the IP address correlation property is often used. Xuewei *et al.* [113] propose an automatic search for causal knowledge based on Markov models. The raw alert stream is grouped into multiple sets based on IP similarity. Then a matrix of the probabilities of transition from one alert to another is built, and a knowledge base of attack patterns is created. Similarly, Zhang *et al.* [114] use Markov chains to correlate alerts in their real-time mining algorithm (RTMA). Security alerts are aggregated into hyper alerts and grouped according to IP addresses. RTMA learns attack patterns from these hyper-alerts. In the approach by Hu *et al.* [115] the absorbing Markov chain (AMC) is used to describe the scenario of multi-step attacks. In AMC, each state can reach an absorbing state, which, once entered, cannot be exited. Thus, this model describes the probabilities of transition from one attack action to another before reaching one of the attack targets.

In the usual Markov model, the probabilities of transitions between states are the only parameter, and the states themselves are visible to the observer. In the *hidden Markov model* (HMM), states are not observable and only the variables that are affected by the state can be tracked. Each state has a probability distribution among all possible outputs. After initializing the model, the forward algorithm uses the initial state and transition probabilities to calculate the observation probabilities for each state based on the sequence of observations. For example, Luktarhan *et al.* [116] train hidden Markov models for each attack scenario. The probability of an alert sequence appearing in a particular model is calculated as the intrusion probability.

A series of papers by Kholidy *et al.* [117]–[119] is dedicated to attack prediction in cloud systems. The authors propose the autonomic cloud intrusion detection framework (ACIDF), which integrates models based on the hidden Markov model and the variable-order Markov model (VMM). The observable parts of the hidden Markov model are system events, and the hidden parts are its states. Events are observed with different probability distributions depending on the system state. HMM and VMM define the sequence of events corresponding to the attack signature as a series of transitions between states with a certain probability. The HMM algorithm calculates the risk level of an alert, and then this risk is matched against one of the defined risk levels. Unlike the hidden Markov model, which requires an attack knowledge base, the variable-order Markov model can operate in the absence of such information. In this model, the number of variables, that each state depends on, can vary. VMM is more commonly used to analyze contextual information. Its main role is to compute the conditional probability distribution of the next alert after receiving the previous alert sequence.

Holgado *et al.* [120] add an average alert vector to the HMM in a multi-step attack prediction approach. The HMM is built for each type of attacks using a training set of alerts grouped according to the similarity of the attack description to the content of the Common Vulnerabilities and Exposures (CVE) documents. The Viterbi algorithm [121] is used to find the most likely paths for different scenarios. This algorithm, in the context of Markov chains, obtains the most probable sequence of events that have occurred. Zhang *et al.* [122] use a hidden Markov model to predict multi-step attacks by

training with the Baum-Welch (BW) algorithm [123], which is used to find the maximum likelihood estimate of HMM parameters for a given set of observations. Attack scenarios are recognized using the Forward-backward algorithm [124], which calculates the probability of a specific sequence of HMM observations. Attack prediction is carried out using the Viterbi algorithm.

Researchers also suggest multi-level architectures of hidden Markov models. Zegeye *et al.* [125], [126] use an HMM of top and bottom layers, each with two functional levels. The first level trains and evaluates the HMM parameters using the Baum-Welch algorithm, and the second level searches for hidden states using the Viterbi algorithm. This multi-level model makes it possible to divide the problem space into smaller manageable parts. Shawly *et al.* [127] propose two architectures for detecting multiple interleaved attacks based on a database of known attack patterns represented as an HMM. The design of the first architecture is based on modifying the HMM parameters to detect multi-step attacks in the presence of mixed alerts. The second architecture allows one to eliminate the interleaving of mixed alerts from various attacks for the HMM processing subsystem.

In addition to the Baum-Welch and Viterbi learning algorithms, also called segmental K-means (SKM), the following optimization algorithms can be applied for hidden Markov model: expectation-maximization (EM) [128], spectral algorithms (SAs) [129], differential evolution (DE) [130]. The EM optimization algorithm allows estimating unobservable hidden variables by finding maximum likelihood estimates. Spectral algorithms use the method of moments [131] to determine the hidden Markov model parameters by selecting empirical moments from experimental data and determining the hidden Markov model parameters that give expected values equal to the selected values. Differential evolution is a genetic optimization algorithm with genetic mutation, crossover, and selection. Chadza *et al.* [132] analyze the application of these algorithms in assessing the accuracy of multi-step attack detection and prediction. The results demonstrate the advantage of the Baum-Welch and Viterbi algorithms in conjunction with optimization algorithms, including hybrid approaches with both learning algorithms.

Markov models can also be used to search for anomalies in data. Shin *et al.* [133] use a Markov model, including a matrix of state transition probabilities and an initial probability distribution, to measure the deviation degree of incoming data from the norm in real-time. In this case, the normal states of the system are constructed by K-means cluster analysis on a training dataset. Saaudi *et al.* [134] simulate user behavior using the hidden Markov model with Stochastic Gradient Techniques (SGD-HMM), which is then used to detect any deviations from normal behavior. Such approach adjusts the HMM parameters by training on several iterations. The input data is system logs, detailed based on three attributes such as session, day, and week.

The attacker's attributes can also be used as parameters for hidden states. Katipally *et al.* [112] use a hidden Markov model for attacker's behavior analysis and prediction. To calculate the probabilities for each type of behavior using the HMM, the authors created five alert sets describing the behavior of different categories of attackers based on their goals, intentions, and knowledge. In real-time, the system predicts intrusions based on learned HMM rules.

Bayesian networks and Markov models ensure a high rate of security event correlation and the ability to adapt to new knowledge. Also, these models provide the opportunity to assess the probability of various security events and automatically generate correlation rules. At the same time, training and fine-tuning of such models requires a sufficient amount of data and computing resources. The quality of attack detection also depends on expert knowledge of the threshold.

Another category of statistical correlation methods is *sequential pattern mining*, which looks for statistically relevant patterns between data examples where values are delivered sequentially. Security alerts can be presented as a global event sequence, sorted by time and describing the attacker behavior. Multi-stage attack patterns can be discovered by analyzing sequence data. The approach by Brahmi and Yahia [135] is based on a closed multi-dimensional sequential pattern mining algorithm, called Closed Multi-Dimensional PrefixSpan (CMD PrefixSpan), which is an improved version of the PrefixSpan [136]. The search for frequently encountered alert sequences is performed using a multi-dimensional table with alert attributes.

Similarly, Lv *et al.* [137] propose a TPrefixSpan alert correlation algorithm based on searching for PrefixSpan patterns. The TPrefixSpan algorithm introduces a time interval, saving the time spent rescanning the dataset during sequence mining. Correlation results are expressed in the form of sequence diagrams. Xian and Zhang [138] propose a privacy-preserving multi-step attack correlation (PPMAC) approach to event sequence analysis that also aims to prevent the risk of confidential information disclosure using the k-anonymity method [139] to anonymize attributes and preserve semantic.

Statistical methods for security event correlation provide the ability to use training datasets to tune attack and detection models. Therefore, as it can be noted, they are more often used for intrusion detection, and open datasets are actively used to evaluate this type of methods. At the same time, we note a high dependence of the tuning of statistical correlation models on the quality of training data. Inaccurate tuning of statistical models can lead to frequent false positives when attacks are detected. To prevent this, it is necessary to develop a correlation algorithm for a specific type of data, which often makes statistical models more narrowly focused.

C. MACHINE LEARNING

Correlation based on machine learning and data mining uses automatically generated alert comparison coefficients. Machine learning methods can be used to correlate alerts and gradually recover an attack scenario by matching alert features and determining the probability of their match.

TABLE 7. Statistical-based methods.

№	Paper	Year	Correlation method	Knowledge extraction	Detection type	Info. source	Corr. layer	Architecture	Dataset	Evaluation
68.	Ren et al. [105]	2010	Bayesian networks	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
69.	Marchetti et al. [140]	2011	Bayesian networks	Unsupervised	Intrusion	Single	Event	Central.	DEFCON 18	Recon.
70.	Katipally et al. [112]	2011	Markov model	Supervised	Intrusion	Multiple	Event	Central.	Generated	Recon.
71.	Anbarestani et al. [104]	2012	Bayesian networks	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
72.	Kavousi and Akbari [107]	2012	Bayesian networks	Unsupervised	Intrusion	Multiple	Event	Distrib.	DARPA 2000	Recon.
73.	Kavousi and Akbari [108]	2014	Bayesian networks	Unsupervised	Intrusion	Multiple	Event	Hierarch.	DARPA 2000, SOTM34	Metric
74.	Luktarhan et al. [116]	2012	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
75.	Brahmi and Yahia [135]	2013	Sequential pattern mining	Unsupervised	Intrusion	Single	Event	Central.	NSA	Metric
76.	Wang and Yang [106]	2013	Bayesian networks	Supervised	Intrusion	Multiple	Event	Distrib.	DARPA 2000	Recon.
77.	Shin et al. [133]	2013	Markov model	Supervised	Anomaly	Multiple	Event	Central.	DARPA 2000	Formal
78.	Kholidy et al. [117]	2014	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
79.	Kholidy et al. [118]	2014	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
80.	Kholidy et al [119]	2014	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
81.	Xuewei et al. [113]	2014	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Recon.
82.	Zhang et al. [122]	2014	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Formal
83.	Ramaki et al. [109]	2015	Bayesian networks	Unsupervised	Intrusion	Multiple	Event	Distrib.	DARPA 2000	Metric
84.	Lv et al. [137]	2015	Sequential pattern mining	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Formal
85.	Faraji Daneshgar and Abbaspour [141]	2016	Sequential pattern mining	Unsupervised	Intrusion	Multiple	Event	Central.	DARPA 2000, ISCXIDS2012	Recon.
86.	Xian and Zhang [138]	2016	Sequential pattern mining	Unsupervised	Intrusion	Multiple	Event	Central.	DEFCON 9	Metric
87.	Holgado et al. [120]	2017	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
88.	Hu et al. [115]	2018	Markov model	Supervised	Intrusion	Multiple	Event	Central.	Generated	Metric
89.	Zhang et al. [114]	2019	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
90.	Saaudi et al. [142]	2019	Markov model	Unsupervised	Anomaly	Multiple	Event	Central.	CERT dataset	Metric
91.	Zegeye et al. [125]	2019	Markov model	Unsupervised	Intrusion	Multiple	Event	Central.	CICIDS2017	Metric
92.	Zegeye et al. [126]	2019	Markov model	Unsupervised	Intrusion	Multiple	Event	Central.	NSL-KDD	Metric
93.	Shawly et al. [127]	2019	Markov model	Signature	Intrusion	Multiple	Event	Distrib.	DARPA 2000	Metric
94.	Pivarníková et al. [110]	2020	Bayesian networks	Unsupervised	Intrusion	Multiple	Event	Distrib.	CICIDS2017	Metric
95.	Chadza et al. [132]	2020	Markov model	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric

Cipriano *et al.* [143] introduce Nexat, a tool that uses machine learning to learn attacker behavior based on previous alert histories. This tool groups alerts into sessions by IP address. Nexat then uses the list of attack sessions to determine which alerts are most likely to occur in a single attack session.

Decision trees can be used to correlate alerts and detect multi-step attacks. Decision tree construction starts at the root, where the training dataset is divided into subsets according to the splitting criterion. The latter nodes usually contain objects that mostly belong to only one class. Benferhat *et al.* [144] supplement the decision tree classifier with a polynomial algorithm that adjusts the classifier's predictions based on expert knowledge. Soleimani and Ghorbani [145] collect alerts based on a correlation matrix and then use controlled decision trees to discover possible combinations of alerts. IP addresses, port numbers, attack type, attack severity, and time of occurrence are used to match alerts. The decision tree learning algorithm obtains specifications for alert sequences of different length.

Machine learning techniques are also used for anomaly detection. Pecchia and *et al.* [146], [147] implement a system for filtering and grouping logs and profiling typical node behavior. This system performs various stages of data processing, which ensure the transition from unstructured text alert information to formalization in the decision tree. The authors use the method of weighing terms, which calculates their relevance to determine the normal system behavior. The lower the relevance, the higher the probability that the term will be generated during the regular system operation. The daily scores for each node in the system are analyzed by principal component analysis (PCA) to examine the ability to capture the variability of text alerts. The generated cluster of alerts is added to the decision tree by the security analyst at the first occurrence of this type of alert. Future occurrences of the same alert are automatically assigned to its cluster, and the root cause is established by tree traversal.

To study or improve the correlation effectiveness, authors often use combined machine learning methods. Change and Wang [148] describe the concept of building attack scenarios based on machine learning using the k-nearest neighbors (KNN) and support vector machine (SVM) algorithms. To extract Android malware attack scenarios, each malicious application is profiled in the training dataset using critical APIs. Machine learning system by Feng *et al.* [149] processes big data from various security logs, including alert information and analytical data, to identify a risky user. 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. Training data is extracted through text mining. As machine learning algorithms, the authors use a multi-layer neural network (MNN), Random Forest (RF), SVM, and logistic regression (LR). The extended SIEM implementation by Mauro and Sarno [150] is equipped with ESkyPRO Network Probe, which can detect events in encrypted Skype traffic. It includes a classification engine using the decision tree, logistic classifier and Bayesian Network Classifier.

An artificial neural network (ANN) consists of interconnected neurons and is used to find patterns in data

TABLE 8. Machine learning methods.

№	Paper	Year	Correlation method	Knowledge extraction	Detection type	Info. source	Corr. layer	Architecture	Dataset	Evaluation
96.	Cipriano et al. [143]	2011	ML	Supervised	Intrusion	Multiple	Event	Central.	UCSB 2008	Metric
97.	Manganiello et al. [151]	2011	SOM	Unsupervised	Intrusion	Multiple	Event	Central.	DEFCON 18	Recon.
98.	Benferhat et al. [144]	2012	Decision tree	Supervised	Intrusion	Multiple	Event	Central.	Generated	Metric
99.	Soleimani and Ghorbani [145]	2012	Decision tree	Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
100.	Pecchia et al. [146]	2014	Decision tree, PCA	Supervised	Anomaly	Multiple	Event	Central.	Generated	Metric
101.	Controneo et al. [147]	2016	Decision tree, PCA	Supervised	Anomaly	Multiple	Event	Central.	Generated	Metric
102.	Mauro and Di Sarno [150]	2018	Decision tree, LR, Bayesian	Supervised	Intrusion	Multiple	Event	Central.	Generated	Metric
103.	Chang and Wang [148]	2016	k-NN, SVM	Supervised	Intrusion	Single	Event	Central.	ContagioDump, Google Play	Metric
104.	Feng et al. [149]	2017	MNN, RF, SVM, LR	Supervised	Intrusion	Multiple	Event	Central.	Generated	Metric
105.	Du et al. [152]	2017	RNN-LSTM	Unsupervised	Anomaly	Multiple	Event	Central.	Generated	Metric
106.	Shen et al. [153]	2018	RNN	Unsupervised	Anomaly	Multiple	Event	Central.	Generated	Metric
107.	Saaudi et al. [142]	2018	RNN, CNN	Unsupervised	Anomaly	Multiple	Event	Central.	CERT dataset	Metric
108.	Zhao et al. [154]	2019	RNN, CNN, SVM	Unsupervised	Intrusion	Multiple	Event	Central.	CICIDS2017, DARPA 2000	Metric
109.	Zhou P. et al. [155]	2021	RNN-LSTM	Unsupervised	Intrusion	Multiple	Event	Central.	DARPA 2000, ISCXIDS2012, CICIDS2017, CICIDS2018	Metric
110.	Abdullayeva [156]	2021	AEnc	Unsupervised	Intrusion	Multiple	Event	Central.	MalwareTrainingSets	Metric

with non-linear relationships between inputs and outputs. Manganiello *et al.* [151] propose an approach to automatically analyze security alerts by combining alert classification using self-organizing maps (SOM), a kind of auto-associative neural network, and unsupervised k-means clustering algorithms. A correlation index is calculated for clusters, and attack scenarios are presented in the form of directed graphs.

Among ANNs, there is a class of deep neural networks (DNNs) with multiple layers between input and output layers. DNN training consists of finding the correct method of mathematical transformations to turn input data into output, regardless of linear or nonlinear correlation. A common class of such networks is the recurrent neural network (RNN), which allows one to analyze sequential data such as time series to predict states at subsequent points in time. The Tiresias by Shen *et al.* [153] use RNNs to predict future events based on previous observations. The collection and preprocessing module reconstruct the security events as a sequences ordered by time stamps. If the predicted RNN state is different from the current one, an anomaly is registered. Du *et al.* [152] propose a DeepLog deep neural network model RNN with long short-term memory (LSTM) to model the system log as a natural language sequence. LSTM allows one to increase the amount of processed information for previous points in time. The model is trained on the normal system behavior and records deviations for anomaly detection.

Also, together with RNN, a convolutional neural network (CNN) is often used to process arrays of input data for attack detection models. CNNs use data sequential convolution and pooling to produce a feature map or feature matrix. Saaudi *et al.* [142] use a CNN to process text data from event logs containing information about user behavior. The resulting feature matrix is fed to the input of the LSTM, which detects abnormal activity. In the proposed model, this CNN learns and displays the most important features of the session sequence, while LSTM takes into account the order of user actions in a particular session. Zhao *et al.* [154] use a multi-step deep learning algorithm SMOTE&CNN-SVM and a bidirectional recurrent neural network (Bi-RNN) model to generate attack chains. Raw alerts are preprocessed based on unbalanced training strategies such as Synthetic Minority Oversampling TEchnique (SMOTE) combined with deep CNNs to select dataset features. Then, using hierarchical SVM classifiers, an optimal classifier is constructed. Bi-RNN is a combination of two unidirectional RNNs: forward RNN records attack chain information from cause to result, and reverse RNN saves information from result to cause to ensure maximum preservation of the correlation information.

Another deep learning method used to attack detection is the autoencoder (AEnc). The main idea of this method is to use backpropagation and obtain at the output the response closest to the input. An autoencoder consists of two main parts: an encoder for matching input to a code, and a decoder for matching a code to recover an input. Abdullayeva [156] presents a deep autoencoder approach for automatic selection of informative features and APT classification. In this approach, informative features are first studied using an autoencoder on the training data, and then the softmax regression level is used for APT classification. Zhou P. *et al.* [155] use the principle of autoencoder operation, in which LSTMs act as encoder and decoder. One LSTM encodes a sequence of security alerts into a hidden feature vector, and the other LSTM decodes for attack prediction. At the same time, LSTMs allow one to ''forget'' irrelevant alerts, thus, have more opportunities to ''remember'' the long-term relationships between different attack stages for detection.

Described correlation approaches using machine learning methods are presented in Table 8. The main advantages of machine learning are the high performance and scalability that are required to process the increasing amount of data.

Deep learning algorithms allow one to automatically select informative features and work directly with raw data, extracting general representations from them at different levels of detail. The main disadvantage of supervised learning is the requirement for a large amount of labeled data, the collection of which is labor-intensive, and the disadvantage of unsupervised learning is a high dependence on the quality of the learning data and methods of extracting knowledge from them.

D. MIXED METHODS

Some approaches and systems combine several security event correlation methods, without the obvious predominance of one over the other. For example, in the framework proposed by Marchetti *et al.* [140], two previously described correlation algorithms are combined as modules: Self-Organizing maps [151] and pseudo-Bayesian correlation [111].

Real-time episode correlation algorithm (RTECA) by Ramaki *et al.* [157] combines frequency analysis of attack trees, similarity analysis of alert types through a correlation matrix, and attribute similarity analysis. The algorithm uses a correlation scheme based on a combination of statistical and streaming data analysis methods. In offline mode, alerts are combined based on similarity into hyper alerts to create an offline attack tree. In online mode, the stages of multi-step attacks are determined in real-time based on the updated tree. The authors also propose a three-stage structure for alert correlation [158] based on the detection of causal relationships between alerts described in early works [109], [157]. Additionally, the generation of prediction rules and the study of new attack strategies are introduced.

A combination of different methods can be used to mitigate the disadvantages of each of them. The alert correlation and attack scenario extraction system proposed by Bateni *et al.* [159] includes fuzzy logic and artificial immune recognition system (AIRS) algorithms. For a new alert, the system tries to find a rule in the set of fuzzy rules in accordance with the specified threshold. If no matching rule is found, the AIRS algorithm is trained using predefined fuzzy rules to detect and remember the correlation relationship between each alert type. Hua *et al.* [160] integrate particle swarming optimization (PSO) [161] with a K-Means clustering algorithm for alert correlation. The K-Means algorithm has the advantages of simplicity and the ability to quickly compute a large number of variables, but the final resulting clusters are dependent on their initial partitions. PSO is used to get the optimal cluster center when initializing K-means.

Hybrid approaches often combine similarity-based correlation and step-based methods. As a rule, systems using this combination assume that the most similar events may be related to the same attack scenario. Tao X. *et al.* [162] propose an alarm correlation method based on the affinity propagation (AP) clustering algorithm [163] and the method of prerequisites and consequences to identify an attacker in IoT networks. This method takes into account the high similarity between alarms triggered by the same attack process. The AP algorithm is used to improve correlation efficiency for the following attributes: attack type, IP address, port, and time. The algorithm considers all samples in the dataset as possible cluster centers and concretizes them, determining which values can relate to one attack scenario. Then, the prerequisites and consequences correlation is used to restore the full attack process.

Wang *et al.* [164] consider attacks against cyber-physical systems and describe an approach to the automatic extraction of attack patterns based on the use of Bayesian networks, as well as the temporal and topological correlation between each attack step. Attack events are aggregated according to the alarm log. Physical attack events and attack sequences are correlated using topological time correlation, frequent attack sequence patterns are extracted, and hidden patterns are discovered from alarm logs. Ying Lin *et al.* [165] propose a collaborative alert ranking framework (CAR) that uses both time correlation and alert content correlation. CAR builds a hierarchical Bayesian model to capture short-term and long-term dependencies in alert sequences. The model is then used to examine content correlations between alerts through their heterogeneous categorical attributes.

A number of studies combine filtering algorithms with data mining or statistical methods. Chen *et al.* [166] describe a wireless multi-step attack pattern recognition method (WMAPRM) based on correlation analysis using basic IEEE 802.11 frame attributes. The method consists of six stages: clustering alerts, building an attack database, building possible attack chains, filtering chains, correlating a multi-step attack behavior recognizing multi-step attack patterns. Jasiul *et al.* [167] propose the PRONTO engine that detects malware in two stages. First, system events are identified and filtered, and then, as events arrive from the identification module, they are compared with malware models in the form of colored Petri nets.

Operation of the MLAPT system by Ghafir *et al.* [168] also has two stages. The first stage is to correlate alerts using a filter to identify APT-related alerts. The second stage applies machine learning methods based on historical information about the monitored network for attack prediction. Papataxiarhis and Hadjiefthymiades [169] describe an online schema for correlating multivariate event data, including a Markov model for capturing event sequences and a time-dependent structure for filtering extracted rules over time using aging or decay function. The aging function is a mechanism for gradually forgetting and prioritizing new rules over old rules over time.

Also, intelligent methods can be used to analyze attack graphs. In the approach by Djemaiel *et al.* [170] multi-step attacks are reconstructed based on big data about network activity. Network activity is monitored using temporal conceptual graphs (TCGs) and attack scenario graphs. A hybrid neural network is used to select the intended attack scenario by performing an evidence-based correlation process with TCG over a period of time.

TABLE 9. Mixed correlation methods.

№	Paper	Year	Correlation method	Knowledge extraction	Detection type	Info. source	Corr. layer	Architecture	Dataset	Evaluation
111.	Marchetti et al.	2011	SOM + Bayesian network	Unsupervised	Intrusion	Multiple	Event	Central.	DEFCON 18	Recon.
112.	Bateni et al.	2013	Attribute-based + ML	Expert rules + Supervised	Intrusion	Multiple	Event	Central.	DARPA 2000, netForensics honeynet	Metric
113.	Chen et al.	2014	Filter-based + ML	Expert rules + Supervised	Intrusion	Multiple	Event	Central.	Generated	Metric
114.	Jasiul et al.	2015	Filter-based + Statistical-based	Supervised	Intrusion	Multiple	Event	Central.	Generated	Formal
115	Ramaki et al.	2015	Bayesian network + Attack graph	Unsupervised	Intrusion	Multiple	Event	Central.	DARPA 2000	Metric
116.	Ramaki and Rasoolzadegan	2016	Bayesian network + Attack graph	Unsupervised	Intrusion	Multiple	Event	Central.	DARPA 2000, DARPA CGC, ISCXIDS2012	Metric
117.	Raju and Geethakumari	2016	Attribute-based + Prerequisites and Consequences	Expert rules	Intrusion	Multiple	Event	Central.	Generated	Recon.
118.	Sapegin et al.	2017	Statistical-based + ML	Signature + Unsupervised	Intrusion + Anomaly	Multiple	Raw	Central.	KDD Cup 1999, Generated	Metric
119.	Hua et al.	2017	Statistical-based + ML	Supervised	Intrusion	Multiple	Rep	Central.	DARPA 2000	Metric
120.	Yang et al.	2018	Filter-based + Statistical-based	Expert rules + Unsupervised	Intrusion + Anomaly	Multiple	Raw	Distrib.	CERT dataset	Metric
121.	Ghafir et al.	2018	Filter-based + ML	Expert rules + Supervised	Intrusion	Multiple	Event	Central.	Generated	Metric
122.	Papataxiarhis and Hadjiefthymiades	2018	Filter-based + Markov model	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Metric
123.	Ying Lin et al.	2018	Attribute-based + Bayesian network	Expert rules	Anomaly	Multiple	Event	Central.	Generated	Metric
124.	Wu and Moon	2019	Attribute-based + Time-based	Expert rules	Intrusion	Multiple	Event	Distrib.	Generated	Metric
125.	Djemaiel et al.	2019	Attack graph + ML	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Recon.
126.	Wang et al.	2020	Time-based + Bayesian network	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Metric
127.	Tao X. et al.	2021	Attribute-based + Prerequisites and Consequences	Unsupervised	Intrusion	Multiple	Event	Central.	Generated	Metric

Hybridization approaches can be considered in terms of using a different number of event data sources. Raju and Geethakumari [171] divide the correlation of cloud events into two stages. At the first stage, events are considered from the point of view of a separate source (homogeneous correlation). In the second stage, events are collected from multiple sources (heterogeneous correlation). Heterogeneous correlation uses a codebook and clustering method to detect incidents. Wu and Moon [172] present the alert correlation method for cyber physical systems with timestamp-based and attribute-based similarity analysis. It consists of three steps performed at different levels of the system: cyber alert correlation, physical alert correlation, and cyber-physical alert correlation. IP addresses can be cyber attributes, and target type/manufacturing process, attack assessment/consequence can be used as physical attributes. Cyber-physical alert correlation is about creating a strong meta-alert between alert aggregation results.

Researchers are also integrating signature-based and anomaly detection methods. Sapegin *et al.* [173] present a prototype SIEM system that combines three types of event analysis: using signatures [52], SQL queries and unsupervised anomaly detection. The analytics method based on SQL queries is mainly used to obtain statistics on processed data. For unsupervised anomaly detection, the authors apply a set of approaches: a negative binomial model to real data, a Poisson model, a hybrid approach combining a vector-space model for text field analysis, spherical k-means, and one-class SVM. The approach by Yang [174] is based on the use of two modules: the attack scenario-based module and the alert filtering module. The first module uses adaptive multi-domain behavioral features for anomaly detection. The second module applies the frequency response from the scenario analysis of the first module and an alert filter based on attack initiation detection.

Described correlation approaches using mixed methods are presented in Table 9. We can say that this type of approach is both the most popular and allows one to take advantage of the various methods described earlier. Moreover, mixed methods provide the ability to combine not only alert correlation methods, but also methods of using knowledge and data sources. It also becomes possible to jointly detect both known and new multi-step attacks.

VI. RESULTS AND DISCUSSION

In this section, we will answer the questions posed in the study, based on the material of the review described in the previous sections of the paper.

A. MAJOR RESEARCH AREAS (RQ1)

In the field of security event analysis, the following main areas can be distinguished, dividing them into five groups:

1) BY THE NATURE OF THE TASK

- review, classification and analysis of existing security event correlation methods, aimed at identifying problems in the cybersecurity area and their possible solutions by systematizing existing knowledge (D1);
- development of new security event correlation approaches characterized by novelty of solutions or areas of application (D2);
- improvement of existing security event correlation approaches to increase their effectiveness by expanding the functionality or adjusting their elements (D3);

- automation of the attack pattern generation and the correlation rule creation, which allows one to reduce the time spent by experts and to refuse manual analysis of large amounts of data (D4);
- development of alert ontologies (D5).

2) BY THE MECHANISM USED

- grouping and clustering of similar alerts to reduce the amount of information processed and alert classification (D6);
- detection of multi-step and targeted attacks, allowing timely notification of intrusions and violations to the security administrator (D7);
- prediction of multi-step and targeted attacks based on incoming alerts, allowing early detection of attacks and attackers' targets (D8).

3) BY THE DECISION TIME

- development of attack models that are trained offline based on the history analysis (D9);
- development of attack models that are trained offline based on the history analysis and online (in real-time) based on the analysis of incoming alerts (D10).

4) BY DATA SOURCES

- development of IDS alert correlation approaches (D11);
- development of system log correlation approaches (D12);
- development of system call correlation approaches (D13).

5) BY AREAS OF APPLICATION

- development of correlation approaches for complex distributed systems (cyber-physical systems, Internet of things) (D14);
- development of correlation approaches for computer systems and networks (D15).

The distribution of the studies considered in this review by the listed areas is presented in Table 10. Note that one research can progress in several directions. The links in the table are not duplicated, and the most relevant direction is selected for each paper.

The main part of the approaches developed by researchers in these areas have a common goal: to detect and predict security breaches that are step-by-step in nature. These problems can be considered multi-step or targeted attacks or cause-and-effect violations of the system or network stability.

TABLE 10. Main research areas on alert correlation.

	№	Area	Reference
Task	D1	Review, classification and analysis of existing correlation methods	Elshoush and Osman [2]; Salah et al. [5]; Mirheidari et al. [4]; Yu Beng et al. [3]; Luh et al. [10]; Ramaki et al. [6]; Husák et al. [7]; Navarro et al. [8]; Pavlov and Voloshina [16]; Kovačević et al. [9]
	D2	Development of new correlation approaches	Benferhat and Sedki [40]; Mohamed et al. [19]; Hostiadi et al. [21]; Rice and Daniels [33]; Bateni and Baraani [35]; Kotenko et al. [37]; Hu et al. [78]; Mauro and Sarno [150]
	D3	Improvement of existing correlation approaches	Alnas et al. [71]; Albanese and Jajodia [90]; Wang and Yang [106]; Benferhat et al. [144]; Sapegin et al. [173]
	D4	Automation of attack pattern generation and rule creation	Liu et al. [27]; Huang et al. [28], [29]; Godefroy et al. [99]; Xuewei et al. [113]; Granadillo et al. [30]; Ussath et al. [54]; Shin et al. [59]
	D5	Development of alert ontologies	Sadighian et al. [47]; Ficco and Romano [64], [65]; Lin et al. [67]; Barzegar and Shajari [75]; Khosravi and Ladani [76]
Mechanism	D6	Grouping and clustering	Brahmi and Yahia [135]; Bajtoš et al. [20]; Hostiadi et al. [21]; Meera and Geethakumari [25]; Kawakani et al. [94]; Faraji Daneshgar and Abbaspour [141]
	D7	Detection	Tan et al. [26]; Das et al. [43]; Holgado et al. [120]; Hu et al. [115]; Zhang et al. [114]; Zhou P. et al. [155]; Ramaki et al. [157]
	D8	Prediction	Cheng et al. [42]; Kholidy et al. [117]–[119]; Zhang et al. [56]; Angelini et al. [92]; Shen et al. [153]
Time	D9	Offline	Marchetti et al. [111]; Almseidin et al. [57]; Bhatt et al. [61]; Xuewei et al. [63]; Cipriano et al. [143]; Soleimani and Ghorbani [145]
	D10	Online and offline	Ren et al. [105]; Herrerías and Gómez [55]; Alserhani et al. [69], [70]; Mahdavi et al. [77]; Kavousi and Akbari [107], [108]
Data source	D11	IDS alerts	Mohamed et al. [19]; Raftopoulos and Dimitropoulos [36]; Saad and Traore [72]; Melo and Macedo [101]; Elshoush and Osman [2]
	D12	System logs	Anbarestani et al. [104]; Jaeger et al. [48], [52]; Landauer et al. [58]; Friedberg I. et al. [73]; Parkinson et al. [100]; Pecchia et al. [146], [147]; Feng et al. [149]
	D13	System calls	Chang and Wang [148]; Abdullayeva [156]
Application	D14	Complex distributed systems	Marvasti et al. [98]; Díaz López et al. [44]; Chien and Ho [95]; Tao X. et al. [162]; Wang et al. [164]; Wu and Moon [172]
	D15	Simple computer systems and networks	Liu et al. [74]; Jinghu et al. [86]; Shittu et al. [87]

B. SECURITY EVENT CORRELATION APPROACHES (RQ2)

Classification and overview of existing approaches are presented in Section V. Similarity-based methods have a simple implementation for determining the relationship between a pair of events. The difficulty lies in choosing the most efficient way to calculate the pairwise correlation of events. Simple matching event attributes can give a lot of false positives. At the same time, complex correlation functions are too specific, which makes similarity-based methods less flexible. Step-based methods often have easily interpretable security event correlation results for the operator to understand. Therefore, this category of methods is well suited for visualization. The implementation of such models is more complicated than similarity-based methods, and requires more computing resources for data processing and storage.

Figure 16 shows the distribution of the described security event correlation approaches by the methods used. We can note that step-based correlation methods are more popular than similarity-based methods. This is due to the fact that step-based methods allow one to more effectively restore the security event sequences, and therefore to detect attack scenarios. Such methods can both use known attack signatures and analyze causal event sequences. Also, step-based methods make it possible to reproduce the attacker's actions using graphs and trees. Similarity-based methods are rarely applied independently. They are more common in hybrid correlation approaches to process the input raw events. Mixed methods are the most common because of their increased functionality and performance, as well as the ability to process and analyze events from different points of view.

As methods of knowledge extraction for security event correlation approaches, we distinguish manual methods, including expert rules and signatures, as well as supervised and automatic methods. By type of detection, we determine whether the correlation approach explores intrusion detection or anomaly detection. Figure 17 shows the distribution of approaches by the knowledge extraction method and by the type of attack detection. We can note that most of the methods in this review rely on knowledge like rules and manually encoded signatures. The attack patterns obtained through unsupervised learning are mostly represented by statistical methods such as Markov models and Bayesian networks. Fewer researchers are searching for anomalies, since in this case it is more difficult to study the logic of a multi-step attack, but at the same time it becomes possible to detect unknown attacks. However, the trend shows that researchers are increasingly giving preference to automated knowledge extraction, which saves time and costs for experts. At the same time, automatic signature generation and anomaly detection methods can return many false positives depending on system settings. Any training methods largely depend on the availability of a reliable model-building dataset that accurately represents the traces found in real networks so that they can be effectively detected.

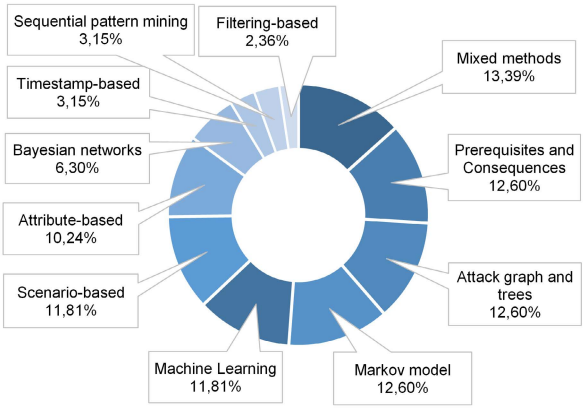

FIGURE 16. Distribution of security event correlation approaches by the methods used.

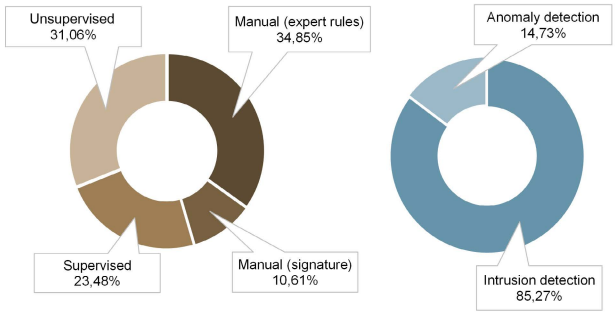

FIGURE 17. Distribution of methods to knowledge extraction (left) and detection type (right).

More researchers are developing security event correlation approaches that can analyze data from various sources (85%), and fewer focus on more specific ones (15%). Modern security systems can include firewalls, authentication services, antiviruses, vulnerability scanners, and IDSs. Data from heterogeneous sources are normalized and analyzed either at a central node (86%), distributed (11.3%) or hierarchically (2.7%). The prevalence of centralized architecture is due to the fact that correlation algorithms in this architecture are easier to implement. However, their scalability is limited, and if the central node fails, the entire system is disrupted. Due to the increasing complexity and size of networks and the increase in volumes of heterogeneous data, the development of distributed and hierarchical correlation systems has become more frequent in recent years. Centralized architectures are no longer beneficial when processing large amounts of data.

Also, depending on the correlation node location, we define systems that operate at the level of raw data (11.2%), events (86.2%) or reports (2.6%). Most of the proposed approaches work directly with incoming events, categorizing them and prioritizing data. At the raw data level, as a rule, data is processed from various network sensors. The report level displays an overall summary of the monitoring data.

C. DATASETS (RQ3)

To evaluate the developed approaches and prove their effectiveness, researchers conduct experiments on open datasets or use datasets obtained as a result of their own system simulation. Thus, the data can be both public and used by any research team, or private. Private data is created using simulated data specially created to validate the approach, and is not available for replication of the experiment by other researchers until the authors make it available. Such datasets are used in the studies reviewed in 47.2% of cases.

The number of experiments performed on the most popular open datasets is shown in Figure 18. The most common public dataset in security event correlation research is the DARPA series, which is used to analyze and test methods for detecting multi-step attacks. This series includes datasets DARPA 1999 [175], DARPA 2000 [176], and DARPA CGP (CGC) [177]. The first two sets are also referred to as

MIT-LL1999 and MIT-LL2000, respectively. The data contains IDS alerts tested using network traffic and audit logs collected from the simulation network. The systems processed this data in batch mode and tried to identify attack sessions during normal operation. The DARPA 1999 dataset consists of 190 samples of 57 attacks which include 8 Probes, 17 DoS attacks, 17 R2L attacks and 15 U2R/Data attack. DARPA 2000 dataset contain two attack scenarios, namely Lincoln Laboratory scenario DDoS (LLDOS) 1.0 and LLDOS 2.0. The first begins with the probing phase, then goes to the infiltration phase and ends with the installation of means for launching DDoS. The second is a complicated version of the first. DARPA CGP is the dataset obtained from the Cyber Grand Challenge project. It provides examples of alerting attacks from sources at different levels: network management systems, firewall, network IDS, and host-based IDS. There are 131 binaries from CGC Qualifying Event (CQE), with various types of bugs injected in them, and 74 binaries from CGC Final Event (CFE).

Many researchers use datasets obtained as a result of hacker competitions, which contain the actions of attackers as close as possible to real ones. DEFCON data [178] comes from the annual Capture the Flag (CTF) competition, in which one team tries to protect a set of services while the other gains unauthorized access to them. DEFCON datasets are built from the footprints left by attackers during the contest and are numbered according to the conference edition number. Also, the datasets resulting from the flag capture competition is the UCSB iCTF datasets [179]. The goal of this competition is to hack the network with a series of attacks without being detected. The Treasure Hunt [180] dataset is the result of a payroll simulator hack competition that aims to execute unauthorized money transfer transactions. Such a set contains a sequence of attacks that are part of an overall plan to achieve a specific goal. The MACCDC [181] datasets include numerous attacks, from scanning to exploitation, resulting from team-to-competition competition.

There are a number of other common datasets that can be used to evaluate approaches for detecting multi-step attacks and reconstructing attack scenarios. The ISCXIDS2012 dataset [182] consists of tagged network traffic collected over 7 days and contains normal traffic (94.42%) and malicious traffic (infiltrating the network from inside (1.26%), HTTP DoS (0.43%), DDoS using an IRC Botnet (2.96%), and Brute Force SSH (0.93%). Attack scenarios can also be explored in the CICIDS2017 dataset [183], which contains network traffic simulating the behaviour of 25 users. Normal traffic is 83.34% of the total number of samples. Implemented attacks include Brute Force FTP/SSH (0.48%), DoS/DDoS (10.4%), Web Attack (0.07%), Port Scan (5.61%), Infiltration (0.001%) and Botnet (0.06%). The CERT dataset [184] contains raw log data, which includes computer events such as logging in, logging out, opening and closing files, visiting websites, using flash drives, and more. The dataset comprises 209 million raw data points and 32 million time-stamped actions. The CTU-13 dataset [185] consists of 13 attack scenarios from various botnets. Botnet attacks include IRC, Spam traffic, Click Fraud, Port Scan, DDoS and Fast Flux (fake domain name system that host malicious contents). This dataset consists of raw data packets, tagged network traffic, the network environment data, and malware.

Security event correlation studies also use the KDD 1999 dataset [187], which contains a large amount of tagged traffic that includes legitimate traffic (19.859%) and four types of attacks: DoS (79.278%), Probe (0.839%), R2L (0.023%) and U2R (0.001%). Although the dataset does not contain pronounced multi-step attacks, it is a labeled reference dataset that is widely used in anomaly detection research. The less common datasets discussed in these studies also include: NSA [188], which contains log data and alerts; ContagioDump [189], which captures scenarios of attacks on user devices using malicious applications; LBNL [190], which contains more than 100 hours of network traffic in packet-based format; netForensics Honeynet (or SOTM34) including IDS alerts (links are out of date). 4SICS Geek Lounge [191] traffic is used to test the approach to attack detection in industrial cyber-physical systems.

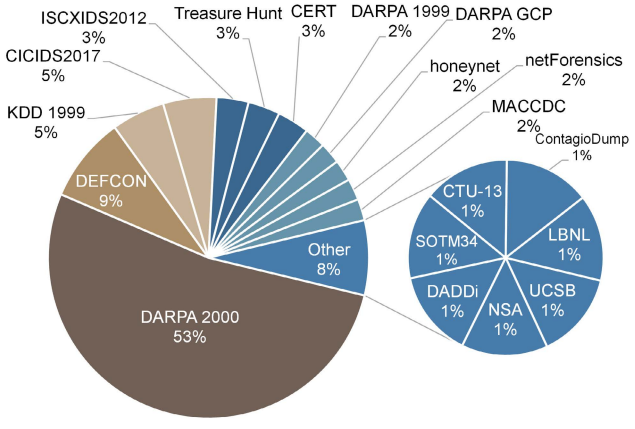

FIGURE 18. Use of open datasets in the publications under review.

D. EVALUATION METRICS (RQ4)

To evaluate the security event correlation, researchers use both different quantitative metrics and more formal approaches. The effectiveness of a correlation approach is assessed by its ability to make correct predictions and properly identify security breaches. Evaluation metrics are also necessary to compare the success of different approaches.

The binary classification of security events into ''normal'' (non-threatening, not part of the attack) and ''attacking'' (threatening, part of the attack) can result in one of the following cases:

- True positive (TP): an attack is identified as an attack;
- False positive (FP): a normal event is incorrectly identified as an attack;
- True negative (TN): a normal event is identified as a normal event;

TABLE 11. Evaluation metrics.

	Metric	Formula	Definition
M1	Accuracy (ACC)	$\frac{TP+TN}{TP+TN+FP+FN}$	1. Correctly Correlated Alerts/All Alert Pairs 2. Correct Attack Detections/All Data Tracks
M2	Precision (P) or Detection rate (DR) or Soundness (Rs)	$\frac{TP}{TP+FP}$	1. Correctly Correlated Alerts/Correlated Alerts [186] 2. Correct Attack Detections/Detected Attack Tracks
M3	Recall (R) or Completeness (Rc) or True positive rate (TPR)	$\frac{TP}{TP+FN}$	1. Correctly Correlated Alerts/Related Alerts [186] 2. Correct Attack Detections/Known Attack Tracks
M4	F1 score	$\frac{2TP}{2TP+FP+FN}$	Harmonic mean of precision and recall
M5	False positive rate (FPR) or Fall-out	$\frac{FP}{FP+TN}$	1. False Correlated Alerts/Unrelated Alerts 2. False Attack Detections/Normal Tracks
M6	False negative rate (FNR) or Miss rate	$\frac{FN}{TP+FN}$	1. False Missed Alerts/Related Alerts 2. False Missed Attack Detections/Known Attack Tracks
M7	Reduction or Recognition rate	$1-\frac{OutputAlerts}{InputAlerts}$	Degree of reduction between input alerts and output alerts

- False negative (FN): an attack is incorrectly identified as a normal event.

For the system to work effectively, FP and FN should be minimized, since otherwise there is a high risk of either false alarm or missed attacks.

A description of the metrics used by researchers in the survey is presented in Table 11. Description of formulas is taken from [192]. The Definition column provides examples of calculating these metrics to (1) similar alert recognition and (2) attack detection. For the first example, the original sample is characterized by the number of All Alert Pairs, among which there are Related Alerts and Unrelated Alerts. The metrics are calculated using Correctly Correlated Alerts, False Correlated Alerts, and All Correlated Alerts. For the second example, the initial data is characterized by all data samples (All Data Tracks), among which there are known attacks (Known Attack Tracks) and normal data (Normal Tracks). To calculate the metrics, Correct Attack Detections, False Attack Detections and all Detected Attack Tracks are determined.

The diagram in Figure 19 shows the frequency with which these metrics are used in the papers included in the corpus of this review. Each scale identifies the number of studies that used a given metric to evaluate the proposed approach. We can note that the most popular metrics are accuracy, precision, recall and FPR.

Table 12 provides M1-M5 metric values for a number of alert correlation approaches presented in this review, which were evaluated using the DARPA 2000. We should note that different authors use common names for metrics that at the same time have the same context, for example recall [59], completeness [75] or TPR [168]. To avoid confusion when comparing the metric values, the metric designation from Table 10 is used in the columns of Table 11, and the name of the metrics is indicated in the cells.

M1-M6 metrics allow one to assess the quality of the correlation or classification of alerts. In addition, the Reduction ratio (or Recognition rate) metric (M7 metric) can be used, which shows the degree of reduction between the original number of alerts and the number of new high-level hyper-alerts. This metric is used by researchers who try to minimize the alert number for analysis. Table 13 shows the values of this metric when evaluating correlation approaches at DARPA 2000.

We can note that while the M1-M6 metrics are mainly used to evaluate step-based methods, the M7 score is more often used for similarity-based correlation methods. A comparison of approaches based on the described metric values will not be entirely correct, since the conditions for conducting experiments may not coincide for different researchers. At the same time, we can note that approaches based on prerequisites and consequences models and using machine learning can provide both greater accuracy and completeness in multi-step attack detection. In general, all the approaches in the above

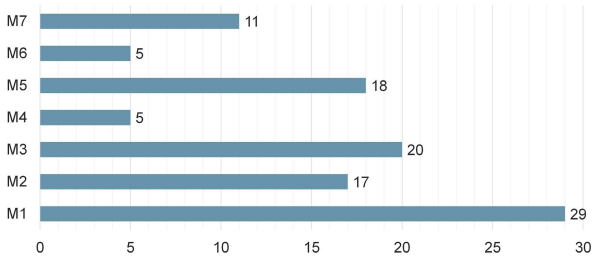

FIGURE 19. Frequency of use of metrics in research evaluation.

TABLE 12. Implications of M1-M5 metrics in evaluating alert correlation approaches using the DARPA 2000 dataset.

Paper	Corr. method	M1, % (ACC)	M2, % (P, DR, Rs)	M3, % (R, Rc, TPR)	M4, % (F1)	M5, % (FPR)
[47]	Scenario-based	–	–	91,08 (TPR)	–	8,92
[57]		97,84	–	–	–	–
[56]		97,78-99,7	93,4-99,7 (DR)	93,4 (R)	–	0,7-7,3
[59]		–	95,18 (P)	92,46 (R)	93,80	–
[69]	Prerequisites and Consequences	–	92,00	–	–	8,1-8,25
[72]		–	99,7 (Rs)	100 (Rc)	–	–
[75]		–	97,22 (Rs)	100 (Rc)	–	–
[105]	Bayesian network	–	–	96,8 (TPR)	–	12,90
[104]		–	72,41-100 (Rs)	91,3-96,72(Rc)	–	–
[108]		–	–	–	–	4,35-11,54
[157]		92,3-99,2	–	–	–	2,4-4,8
[116]	HMM	95,06	–	–	–	2,10
[132]		72-87,3	–	–	–	–
[145]	Decision tree	98,73	–	–	–	1,03
[159]	Attribute-based + ML	–	82-94,8 (Rs)	74,5,7-95,7(Rc)	–	2,45-5,3
[160]	Statistical-based + ML	92,71-99,67	–	–	–	7,30
[154]	RNN, CNN, SVM	96,00	82-99,6 (P)	52,2-99,6 (R)	–	–
[155]	RNN-LSTM	–	87,1-100 (P)	94-100(R)	92,6-99,9	–

TABLE 13. Implications of the M7 metric in evaluating alert correlation approaches using the DARPA 2000 dataset.

Paper	Corr. method	Reduction rate, %
[28]	Attribute-based	58,67-95,29
[29]	Attribute-based	57,88-95,68
[38]	Filtering-based	99
[23]	Attribute-based	99,98
[114]	Markov model	96
[102]	Attack graph	96,68

tables demonstrate high values of metrics, and in the future the research task can be reduced to a decrease in false positives with an increase in accuracy.

In addition to the metrics listed in Table 11, security event correlation approaches can also be evaluated in terms of performance [117], [120], [135], [146], for example, how quickly a multi-step attack will be detected when its early indications appear. The metric can also be defined as the probability of an attack [110], [127] or the probability of its detection [88]. Such metrics also make it possible to evaluate how correctly the attack scenario is restored.

The authors also introduce their own definitions for numerical metrics. Thus, Hu *et al.* [115], based on probabilistic reasoning, suggest using two metrics to evaluate attack scenarios and attackers: the expected number of visits (ENV) and the expected success probability (ESP). ENV gives the expected number of each alert in the current scenario depending on the initial alerts raised by the intruder. ESP is the probability that an attacker will achieve a specific target when starting an attack with a specific initial alert. This metric is equal to the sum of the probabilities of transitions between intermediate alerts to the target of the attack.

E. CHALLENGES AND PROSPECTS (RQ5)

Security event analysts identify the following common challenges when developing approaches to alert correlation:

- *Hiding malicious patterns.* It is necessary to identify not only abnormal events, but also to restore their connections. Detection of individual steps can be skipped due to technical limitations of systems and networks. Even if alerts about an attack are identified, it is rather difficult to define the entire logic of an attacker without additional knowledge, especially in complex and distributed systems. When the attacker uses deceptive maneuvers and concealment of their actions, this is especially problematic. Often the multi-step attacks detection is complicated by the attacker intent to hide his trail.
- *Explainability of event semantics.* Alerts from various sources, such as IDS, may not provide sufficient information about the cause of the problem. In addition, cyber-analysts are faced with a semantic gap between low-level audit events and high-level system behavior.
- *Security event knowledge base support.* When using signature correlation methods and expert rules, it is necessary to constantly update the knowledge base of attack patterns and rule descriptions, respectively. The performance of signature-based methods can be limited by detecting various combinations of single-stage attacks [59]. In this direction, we can also add the development of event ontologies [47]. Creating such knowledge bases manually with the involvement of experts is quite laborious.
- *Analysis of large and/or heterogeneous data.* Especially in distributed and complex systems, it becomes necessary to analyze a very large amount of data, which can

be difficult for systems operating in real-time, and also requires a lot of computing resources. When processing data from several sources, the problem of data unification and standardization may arise, including even for one event, different security systems can generate alerts in different formats.

- *Few publicly available datasets.* As we have seen in this review, there are not many standard datasets for evaluating multi-step intrusion detection systems that are public. Most researchers use generated datasets that are not shared to reproduce experiments and validate their own designs. At the same time, the most popular DARPA 2000 set contains only two attack scenarios, and it can be considered obsolete.

Considering these problems, we propose the following areas of research in the development of approaches to security event correlation, which can help in resolving difficulties encountered:

- *Analysis of semantics of security events.* It is necessary to study the characteristics of multi-step attacks and attacker behavior, taking into account the semantics of alerts. Such studies are conducted both on the basis of the assumption that attacking actions have a strong internal correlation [56], and on the basis of the attribution of attackers [112]. Another solution is to analyze the similarity of alerts based on their semantics [75]. The increasing number of approaches with hybrid methods of security event correlation and their high efficiency allows one to consider the actions of the attacker from different points of view.
- *Development and use of self-learning event knowledge bases.* This can be useful for improving event correlation approaches without known attack signatures and fixed rules. Using unsupervised techniques, multi-step attack rules can be generated automatically without the knowledge of predetermined single-step actions of the attacker. In this case, an important direction is the development of adaptive event correlation methods or online learning algorithms for timely updating of pattern databases, as demonstrated by Shin *et al.* [59], Khosravi and Ladani [76], and others.
- *Security event correlation for predictive analytics* is also an important area of development. Most of the existing forecasting systems are capable of making binary decisions: whether an attack will occur or not. But at the same time, not many systems provide a sequence of actions for an attacker. Attack scenario steps prediction can be based on the use of unsupervised machine learning, such as deep neural networks [153].
- *Support for distributed event correlation and big data processing.* Development of approaches capable of performing distributed security event correlation and analyzing large amounts of data, including those based on current big data processing technologies. This direction is especially relevant for the developing IoT systems and cyber-physical systems [20], [164]. In the case of a large amount of event data, a good solution is to support parallel computing and use big data processing tools [37].
- *Support for normalization and unification of security events.* A number of alert normalization formats have now been introduced and are used by research communities and commercial product developers such as IDMEF, CEE, IODEF, and CEF [6]. At the same time, the existing formats have a limited number of fields, which are sometimes not enough to describe the relationship between events. Unified approaches allow analyzing events without a clear reference to their format.
- *Development of new datasets* for evaluating alert correlation methods. Publication of the experimental data used. Since some data may contain confidential data, authors can make it available on request to allow data sharing, as Hu *et al.* [78]. Evaluation of the developed approaches both on existing and new datasets allow one to demonstrate the effectiveness of the approach to security event correlation, taking into account different types of attacks.

We should note that the above lists do not cover all possible difficulties and directions of research development, and their complete analysis requires a separate in-depth study.

VII. CONCLUSION

This paper provided a systematic review of the security event correlation literature over the past decade, which allowed us to present the current state of research in this area. The review methodology included the formulation of research questions, a keyword search strategy, and a recursive search for scientific publications that were selected according to established inclusion and exclusion criteria.

Researchers' security event correlation approaches are necessary to detect and predict incremental security issues such as multi-step or targeted attacks and other causal sequences of abnormal events. We presented a classification of approaches to security event correlation based on correlation methods, knowledge extraction methods, number of sources, level of analysis, and architecture. The corpus of the review includes 127 research papers characterized in accordance with the proposed classification. We described the datasets and metrics that are used in these researches to evaluate correlation approaches. For a number of papers, we provided a comparative effectiveness analysis of the approaches developed by the authors.

We also presented the challenges that researchers face when developing security event correlation approaches. Among them, we distinguished two main: the complexity of the reconstruction of multi-step attacks and the small number of representative publicly available datasets for evaluation. We also considered the improvement of approaches to the security event correlation with a large amount of heterogeneous data as the main direction of future work in the

area. A systematic literature review has shown that there is still considerable interest among researchers in complex and hidden attack detection. The presented state-of-the-art developments, possible problems and their solutions may be of interest to those who want to get acquainted with this area.

As a direction for future work, we consider the development and evaluation of security event correlation models, taking into account the challenges and perspectives outlined in this survey. The main research will be focused on the security events correlation, which allows us to take into account both the structural relationships of event features and the event semantics. In particular, special attention should be paid to methods for processing a large amount of heterogeneous data in order to ensure the universality of the correlation method and its applicability in complex distributed systems.

REFERENCES

[1] T. Limmer and F. Dressler, "Survey of event correlation techniques for attack detection in early warning systems," Dept. Comput. Sci., Univ. Erlangen, Erlangen, Germany, Tech. Rep. 01/08, Jan. 2008, pp. 1–37.
[2] H. T. Elshoush and I. M. Osman, "Alert correlation in collaborative intelligent intrusion detection systems—A survey," *Appl. Soft Comput.*, vol. 11, no. 7, pp. 4349–4365, Oct. 2011.
[3] L. Y. Beng, S. Ramadass, S. Manickam, and T. S. Fun, "A survey of intrusion alert correlation and its design considerations," *IETE Tech. Rev.*, vol. 31, no. 3, pp. 233–240, May 2014.
[4] S. A. Mirheidari, S. Arshad, and R. Jalili, "Alert correlation algorithms: A survey and taxonomy," in *Proc. Int. Symp. Cyberspace Saf. Secur.*, 2013, pp. 183–197.
[5] S. Salah, G. Maciá-Fernández, and J. E. Díaz-Verdejo, "A model-based survey of alert correlation techniques," *Comput. Netw.*, vol. 57, no. 5, pp. 1289–1317, Apr. 2013.
[6] A. A. Ramaki, A. Rasoolzadegan, and A. G. Bafghi, "A systematic mapping study on intrusion alert analysis in intrusion detection systems," *ACM Comput. Surveys*, vol. 51, no. 3, pp. 1–41, May 2018.
[7] M. Husak, J. Komarkova, E. Bou-Harb, and P. Celeda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 640–660, Sep. 2019.
[8] J. Navarro, A. Deruyver, and P. Parrend, "A systematic survey on multi-step attack detection," *Comput. Secur.*, vol. 76, pp. 214–249, Jul. 2018.
[9] I. Kovačević, S. Groš, and K. Slovenec, "Systematic review and quantitative comparison of cyberattack scenario detection and projection," *Electronics*, vol. 9, no. 10, p. 1722, Oct. 2020.
[10] R. Luh, S. Marschalek, M. Kaiser, H. Janicke, and S. Schrittwieser, "Semantics-aware detection of targeted attacks: A survey," *J. Comput. Virol. Hacking Techn.*, vol. 13, no. 1, pp. 47–85, Feb. 2017.
[11] M. Wood and M. Erlinger, *Intrusion Detection Message Exchange Requirements*, document RFC 4766, IETF, 2007.
[12] R. A. Bridges, T. R. Glass-Vanderlan, M. D. Iannacone, M. S. Vincent, and Q. Chen, "A survey of intrusion detection systems leveraging host data," *ACM Comput. Surveys*, vol. 52, no. 6, pp. 1–35, Nov. 2019.
[13] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer Peer Netw. Appl.*, vol. 12, pp. 1–9, Mar. 2019.
[14] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowl. Syst.*, vol. 189, Feb. 2020, Art. no. 105124.
[15] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures," *Sensors*, vol. 21, no. 14, p. 4759, Jul. 2021.
[16] A. Pavlov and N. Voloshina, "Analysis of IDS alert correlation techniques for attacker group recognition in distributed systems," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Cham, Switzerland: Springer, Dec. 2020, pp. 32–42.
[17] B. Kitchenham, "Procedures for performing systematic reviews," Dept. Comput. Sci., Keele Univ., Keele, U.K., Tech. Rep. 0400011T.1, 2004, pp. 1–26, vol. 33.
[18] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. 12th Int. Conf. Eval. Assessment Softw. Eng.*, vol. 17, Jun. 2008, pp. 1–10.
[19] A. B. Mohamed, N. B. Idris, and B. Shanmugum, "Alert correlation using a novel clustering approach," in *Proc. Int. Conf. Commun. Syst. Netw. Technol.*, May 2012, pp. 720–725.
[20] T. Bajtoš, P. Sokol, and T. Mézešová, "Multi-stage cyber-attacks detection in the industrial control systems," in *Recent Developments on Industrial Control Systems Resilience*. Cham, Switzerland: Springer, Jan. 2020, pp. 151–173.
[21] D. P. Hostiadi, M. D. Susila, and R. R. Huizen, "A new alert correlation model based on similarity approach," in *Proc. 1st Int. Conf. Cybern. Intell. Syst. (ICORIS)*, vol. 1, Aug. 2019, pp. 133–137.
[22] M. Ghasemigol and A. Ghaemi-Bafghi, "A new alert correlation framework based on entropy," in *Proc. ICCKE*, Oct. 2013, pp. 184–189.
[23] M. GhasemiGol and A. Ghaemi-Bafghi, "E-correlator: An entropy-based alert correlation system," *Secur. Commun. Netw.*, vol. 8, no. 5, pp. 822–836, 2015.
[24] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.
[25] G. Meera and G. Geethakumari, "Event correlation for log analysis in the cloud," in *Proc. IEEE 6th Int. Conf. Adv. Comput. (IACC)*, Feb. 2016, pp. 158–162.
[26] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 447–456, Feb. 2013.
[27] J. Liu, L. Gu, G. Xu, and X. Niu, "A correlation analysis method of network security events based on rough set theory," in *Proc. 3rd IEEE Int. Conf. Netw. Infrastruct. Digit. Content*, Sep. 2012, pp. 517–520.
[28] C.-J. Huang, C.-Y. Li, Y.-W. Wang, C.-F. Lin, J.-J. Liao, and K.-W. Hu, "An adaptive rule-based intrusion alert correlation detection method," in *Proc. 1st Int. Conf. Netw. Distrib. Comput.*, Oct. 2010, pp. 222–226.
[29] C.-J. Huang, K.-W. Hu, H. Cheng, T.-K. Chang, Y.-C. Luo, and Y.-J. Lien, "Application of type-2 fuzzy logic to rule-based intrusion alert correlation detection," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 4, pp. 2865–2874, 2012.
[30] G. G. Granadillo, M. El-Barbori, and H. Debar, "New types of alert correlation for security information and event management systems," in *Proc. 8th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Nov. 2016, pp. 1–7.
[31] A. A. Mohamed and O. Basir, "Fusion based approach for distributed alarm correlation in computer networks," in *Proc. 2nd Int. Conf. Commun. Softw. Netw.*, 2010, pp. 318–324.
[32] A. P. Dempster, "The Dempster–Shafer calculus for statisticians," *Int. J. Approx. Reasoning*, vol. 48, no. 2, pp. 365–377, Jun. 2008.
[33] G. Rice and T. Daniels, "A hierarchical approach for detecting system intrusions through event correlation," in *Proc. Int. Conf. Commun., Netw., Inf. Secur.*, Phoenix, AZ, USA, 2012, pp. 1–12.
[34] Q. Wu, Y. Gu, X. Cui, P. Moka, and Y. Lin, "A graph similarity-based approach to security event analysis using correlation techniques," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.
[35] M. Bateni and A. Baraani, "Time window management for alert correlation using context information and classification," *Int. J. Comput. Netw. Inf. Secur.*, vol. 5, no. 11, pp. 9–16, Sep. 2013.
[36] E. Raftopoulos and X. Dimitropoulos, "IDS alert correlation in the wild with EDGe," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 10, pp. 1933–1946, Oct. 2014.
[37] I. Kotenko, A. Fedorchenko, I. Saenko, and A. Kushnerevich, "Parallelization of security event correlation based on accounting of event type links," in *Proc. 26th Euromicro Int. Conf. Parallel, Distrib. Netw. Process. (PDP)*, Mar. 2018, pp. 462–469.
[38] H. T. Elshoush and I. M. Osman, "An improved framework for intrusion alert correlation," in *Proc. World Congr. Eng.*, vol. 1, 2012, pp. 1–6.
[39] N. B. Anuar, S. Furnell, M. Papadaki, and N. Clarke, "A risk index model for security incident prioritisation," in *Proc. 9th Austral. Inf. Secur. Manage. Conf.* Perth, WA, Australia: Edith Cowan Univ., Dec. 2011, pp. 25–39.
[40] S. Benferhat and K. Sedki, "An alert correlation approach based on security operator's knowledge and preferences," *J. Appl. Non-Classical Logics*, vol. 20, nos. 1–2, pp. 7–37, Jan. 2010.
[41] L. Wang, A. Ghorbani, and Y. Li, "Automatic multi-step attack pattern discovering," *Int. J. Netw. Secur.*, vol. 10, no. 2, pp. 142–152, Mar. 2010.

[42] B.-C. Cheng, C.-C. Huang, M.-T. Yu, and G.-T. Liao, "A novel probabilistic matching algorithm for multi-stage attack forecasts," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 7, pp. 1438–1448, Aug. 2011.

[43] D. Das, U. Sharma, and D. K. Bhattacharyya, "Detection of HTTP flooding attacks in multiple scenarios," in *Proc. Int. Conf. Commun., Comput. Secur. (ICCCS)*, New York, NY, USA, 2011, pp. 517–522.

[44] D. Lopez, M. Blanco, C. Santiago, A. Torres, N. Guataquira, S. Castro, P. Nespoli, and F. Gomez Marmol, "Shielding IoT against cyber-attacks: An event-based approach using SIEM," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–16, Oct. 2018.

[45] L. Liu, K. F. Zheng, and Y. X. Yang, "An intrusion alert correlation approach based on finite automata," in *Proc. Int. Conf. Commun. Intell. Inf. Secur.*, Oct. 2010, pp. 80–83.

[46] V. Gorodetsky and I. Kotenko, "Attacks against computer network: Formal grammar-based framework and simulation tool," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, 2002, pp. 219–238.

[47] A. Sadighian, J. Fernandez, A. Lemay, and S. T. Zargar, "ONTIDS: A highly flexible context-aware and ontology-based alert correlation framework," in *Proc. Int. Symp. Found. Pract. Secur.*, 2013, pp. 161–177.

[48] A. Azodi, D. Jaeger, F. Cheng, and C. Meinel, "Pushing the limits in event normalisation to improve attack detection in IDS/SIEM systems," in *Proc. Int. Conf. Adv. Cloud Big Data*, Dec. 2013, pp. 69–76.

[49] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A semantic web rule language combining OWL and RuleML," *W C Member Submission*, vol. 21, no. 79, pp. 1–31, 2004.

[50] M. J. O'Connor and A. K. Das, "SQWRL: A query language for OWL," in *Proc. OWLED*, vol. 529, 2009, pp. 1–8.

[51] D. L. McGuinness and F. Van Harmelen, "OWL web ontology language overview," *W C Recommendation*, vol. 10, no. 10, pp. 1–12, 2004.

[52] D. Jaeger, M. Ussath, F. Cheng, and C. Meinel, "Multi-step attack pattern detection on normalized event logs," in *Proc. IEEE 2nd Int. Conf. Cyber Secur. Cloud Comput.*, Nov. 2015, pp. 390–398.

[53] P. Bates and J. Wileden, "EDL: A basis for distributed system debugging tools," in *Proc. Hawaii Int. Conf. Syst. Sci.*, 1982, pp. 86–93.

[54] M. Ussath, F. Cheng, and C. Meinel, "Automatic multi-step signature derivation from taint graphs," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–8.

[55] J. Herrerías and R. Gómez, "Log analysis towards an automated forensic diagnosis system," in *Proc. Int. Conf. Availability, Rel. Secur.*, Feb. 2010, pp. 659–664.

[56] K. Zhang, F. Zhao, S. Luo, Y. Xin, and H. Zhu, "An intrusion action-based IDS alert correlation analysis and prediction framework," *IEEE Access*, vol. 7, pp. 150540–150551, 2019.

[57] M. Almseidin, I. Piller, M. Alkasassbeh, and K. Szilveszter, "Fuzzy automaton as a detection mechanism for the multi-step attack," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 9, no. 2, pp. 575–586, 2019.

[58] M. Landauer, F. Skopik, M. Wurzenberger, W. Hotwagner, and A. Rauber, "A framework for cyber threat intelligence extraction from raw log data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 3200–3209.

[59] J. Shin, S.-H. Choi, P. Liu, and Y.-H. Choi, "Unsupervised multi-stage attack detection framework without details on single-stage attacks," *Future Gener. Comput. Syst.*, vol. 100, pp. 811–825, Nov. 2019.

[60] C.-J. Nie, D.-G. Feng, Z.-Q. Han, and P.-R. Su, "A distributional attack scenario monitoring system based on dynamic peer-to-peer overlay hierarchy," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Jul. 2011, pp. 348–355.

[61] P. Bhatt, E. T. Yano, and P. Gustavsson, "Towards a framework to detect multi-stage advanced persistent threats attacks," in *Proc. IEEE 8th Int. Symp. Service Oriented Syst. Eng.*, Apr. 2014, pp. 390–395.

[62] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Lead. Inf. Warfare Secur. Res.*, vol. 1, no. 1, pp. 113–125, 2011.

[63] F. Xuewei, W. Dongxia, M. Guoqing, and L. Jin, "Research on the key technology of reconstructing attack scenario based on state machine," in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, vol. 1, Jul. 2010, pp. 42–46.

[64] M. Ficco and L. Romano, "A correlation approach to intrusion detection," in *Proc. Int. Conf. Mobile Lightweight Wireless Syst.*, vol. 45, 2010, pp. 203–215.

[65] M. Ficco, "Achieving security by intrusion-tolerance based on event correlation," *Netw. Protocols Algorithms*, vol. 2, no. 3, pp. 70–84, Oct. 2010.

[66] M. Ficco and L. Romano, "A generic intrusion detection and diagnoser system based on complex event processing," in *Proc. 1st Int. Conf. Data Compress., Commun. Process.*, Jun. 2011, pp. 275–284.

[67] Z. Lin, S. Li, and Y. Ma, "Real-time intrusion alert correlation system based on prerequisites and consequence," in *Proc. 6th Int. Conf. Wireless Commun. Netw. Mobile Comput. (WiCOM)*, Sep. 2010, pp. 1–5.

[68] F. Alserhani, M. Akhlaq, I. U. Awan, and A. J. Cullen, "Detection of coordinated attacks using alert correlation model," in *Proc. IEEE Int. Conf. Prog. Informat. Comput.*, vol. 1, Dec. 2010, pp. 542–546.

[69] F. Alserhani, M. Akhlaq, I. U. Awan, A. J. Cullen, and P. Mirchandani, "MARS: Multi-stage attack recognition system," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 753–759.

[70] F. Alserhani, "A framework for multi-stage attack detection," in *Proc. Saudi Int. Electron., Commun. Photon. Conf.*, Apr. 2013, pp. 1–6.

[71] M. Alnas, A. M. Hanashi, and E. M. Laias, "Detection of Botnet multi-stage attack by using alert correlation model," *Int. J. Eng. Sci.*, vol. 2, no. 10, pp. 24–34, 2013.

[72] S. Saad and I. Traore, "Semantic aware attack scenarios reconstruction," *J. Inf. Secur. Appl.*, vol. 18, no. 1, pp. 53–67, Jul. 2013.

[73] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Comput. Secur.*, vol. 48, pp. 35–57, Feb. 2015.

[74] Y. Liu, M. Zhang, D. Li, K. Jee, Z. Li, Z. Wu, J. Rhee, and P. Mittal, "Towards a timely causality analysis for enterprise security," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.

[75] M. Barzegar and M. Shajari, "Attack scenario reconstruction using intrusion semantics," *Expert Syst. Appl.*, vol. 108, pp. 119–133, Oct. 2018.

[76] M. Khosravi and B. T. Ladani, "Alerts correlation and causal analysis for APT based cyber attack detection," *IEEE Access*, vol. 8, pp. 162642–162656, 2020.

[77] E. Mahdavi, A. Fanian, and F. Amini, "A real-time alert correlation method based on code-books for intrusion detection systems," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101661.

[78] H. Hu, J. Liu, Y. Zhang, Y. Liu, X. Xu, and J. Tan, "Attack scenario reconstruction approach using attack graph and alert data mining," *J. Inf. Secur. Appl.*, vol. 54, Oct. 2020, Art. no. 102522.

[79] S. J. Templeton and K. Levitt, "A requires/provides model for computer attacks," in *Proc. Workshop New Secur. Paradigms (NSPW)*, 2000, pp. 31–38.

[80] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metric," in *Data and Applications Security XXII*. Berlin, Germany: Springer, 2008, pp. 283–296.

[81] R. J. Trudeau, *Introduction to Graph Theory*. Chelmsford, MA, USA: Courier Corporation, 2013.

[82] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100219.

[83] S. Roschke, F. Cheng, and C. Meinel, "A new alert correlation algorithm based on attack graph," in *Computational Intelligence in Security for Information Systems*, vol. 6694. Berlin, Germany: Springer, 2011, pp. 58–67.

[84] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, pp. 344–348, Jun. 1962.

[85] S. Warshall, "A theorem on Boolean matrices," *J. ACM*, vol. 9, no. 1, pp. 11–12, Jan. 1962.

[86] X. Jinghu, L. Aiping, Z. Hui, and Y. Hong, "A multi-step attack pattern discovery method based on graph mining," in *Proc. 2nd Int. Conf. Comput. Sci. Netw. Technol.*, Dec. 2012, pp. 376–380.

[87] R. Shittu, A. Healing, R. Ghanea-Hercock, R. Bloomfield, and M. Rajarajan, "Intrusion alert prioritisation and attack detection using post-correlation analysis," *Comput. Secur.*, vol. 50, pp. 1–15, May 2015.

[88] J. Navarro-Lara, A. Deruyver, and P. Parrend, "Morwilog: An ACO-based system for outlining multi-step attacks," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–8.

[89] M. N. Hossain, S. M. Milajerdi, J. Wang, B. Eshete, R. Gjomemo, R. Sekar, S. Stoller, and V. Venkatakrishnan, "SLEUTH: Real-time attack scenario reconstruction from COTS audit data," in *Proc. 26th USENIX Secur. Symp. (USENIX Secur.)*, 2017, pp. 487–504.

[90] M. Albanese and S. Jajodia, "A graphical model to assess the impact of multi-step attacks," *J. Defense Model. Simul., Appl., Methodol., Technol.*, vol. 15, no. 1, pp. 79–93, Jan. 2018.

[91] A. Shameli-Sendi, M. Dagenais, and L. Wang, "Realtime intrusion risk assessment model based on attack and service dependency graphs," *Comput. Commun.*, vol. 116, pp. 253–272, Jan. 2018.

[92] M. Angelini, S. Bonomi, E. Borzi, A. D. Pozzo, S. Lenti, and G. Santucci, "An attack graph-based on-line multi-step attack detector," in *Proc. 19th Int. Conf. Distrib. Comput. Netw.*, Jan. 2018, pp. 1–10.

[93] P. Jaccard, "The distribution of flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, Feb. 1912.
[94] C. T. Kawakani, S. B. Junior, R. S. Miani, M. Cukier, and B. B. Zarpelão, "Intrusion alert correlation to support security management," in *Proc. Anais do 12th Simpósio Brasileiro de Sistemas de Informação*, 2016, pp. 313–320.
[95] S.-H. Chien and C.-S. Ho, "A novel threat prediction framework for network security," in *Advances in Information Technology and Industry Applications*. Berlin, Germany: Springer, 2012, pp. 1–9.
[96] Z. Zali, M. R. Hashemi, and H. Saidi, "Real-time attack scenario detection via intrusion detection alert correlation," in *Proc. 9th Int. ISC Conf. Inf. Secur. Cryptol.*, Sep. 2012, pp. 95–102.
[97] I. Kotenko and A. Chechulin, "A cyber attack modeling and impact assessment framework," in *Proc. 5th Int. Conf. Cyber Conflict (CYCON)*, 2013, pp. 1–24.
[98] M. Marvasti, A. Poghosyan, A. Harutyunyan, and N. Grigoryan, "An anomaly event correlation engine: Identifying root causes, bottlenecks, and black swans in IT environments," *VMware Tech. J.*, vol. 2, pp. 35–45, Jun. 2013.
[99] E. Godefroy, E. Totel, M. Hurfin, and F. Majorczyk, "Automatic generation of correlation rules to detect complex attack scenarios," in *Proc. 10th Int. Conf. Inf. Assurance Secur.*, Nov. 2014, pp. 23–28.
[100] S. Parkinson, M. Vallati, A. Crampton, and S. Sohrabi, "GraphBAD: A general technique for anomaly detection in security information and event management," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 16, p. e4433, Aug. 2018.
[101] R. Melo and D. Macedo, "A cloud immune security model based on alert correlation and software defined network," in *Proc. IEEE 28th Int. Conf. Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE)*. IEEE, Jun. 2019. pp. 52–57.
[102] H. Zhang, X. Jin, Y. Li, Z. Jiang, Y. Liang, Z. Jin, and Q. Wen, "A multistep attack detection model based on alerts of smart grid monitoring system," *IEEE Access*, vol. 8, pp. 1031–1047, 2020.
[103] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," *IEEE Trans. Dependable Secur. Comput.*, vol. 9, no. 1, pp. 61–74, Jan. 2012.
[104] R. Anbarestani, B. Akbari, and F. Fathi, "An iterative alert correlation method for extracting network intrusion scenarios," in *Proc. 20th Iranian Conf. Electr. Eng. (ICEE)*, May 2012, pp. 684–689.
[105] H. Ren, N. Stakhanova, and A. A. Ghorbani, "An online adaptive approach to alert correlation," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*. Berlin, Germany: Springer, 2010, pp. 153–172.
[106] C.-H. Wang and J.-M. Yang, "Adaptive feature-weighted alert correlation system applicable in cloud environment," in *Proc. 8th Asia Joint Conf. Inf. Secur.*, Jul. 2013, pp. 41–47.
[107] F. Kavousi and B. Akbari, "Automatic learning of attack behavior patterns using Bayesian networks," in *Proc. 6th Int. Symp. Telecommun. (IST)*, Nov. 2012, pp. 999–1004.
[108] F. Kavousi and B. Akbari, "A Bayesian network-based approach for learning attack strategies from intrusion alerts," *Secur. Commun. Netw.*, vol. 7, no. 5, pp. 833–853, 2014.
[109] A. A. Ramaki, M. Khosravi-Farmad, and A. G. Bafghi, "Real time alert correlation and prediction using Bayesian networks," in *Proc. 12th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Sep. 2015, pp. 98–103.
[110] M. Pivarníková, P. Sokol, and T. Bajtoš, "Early-stage detection of cyber attacks," *Information*, vol. 11, no. 12, p. 560, Nov. 2020.
[111] M. Marchetti, M. Colajanni, and F. Manganiello, "Identification of correlated network intrusion alerts," in *Proc. 3rd Int. Workshop Cyberspace Saf. Secur. (CSS)*, Sep. 2011, pp. 15–20.
[112] R. Katipally, L. Yang, and A. Liu, "Attacker behavior analysis in multistage attack detection system," in *Proc. 7th Annu. Workshop Cyber Secur. Inf. Intell. Res. (CSIIRW)*, 2011, pp. 1–4.
[113] F. Xuewei, W. Dongxia, H. Minhuan, and S. Xiaoxia, "An approach of discovering causal knowledge for alert correlating based on data mining," in *Proc. IEEE 12th Int. Conf. Dependable, Autonomic Secure Comput.*, Aug. 2014, pp. 57–62.
[114] Y. Zhang, S. Zhao, and J. Zhang, "RTMA: Real time mining algorithm for multi-step attack scenarios reconstruction," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun., IEEE 17th Int. Conf. Smart City, IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Aug. 2019, pp. 2103–2110.
[115] H. Hu, Y. Liu, H. Zhang, and Y. Zhang, "Security metric methods for network multistep attacks using AMC and big data correlation analysis," *Secur. Commun. Netw.*, vol. 2018, pp. 1–14, Jun. 2018.
[116] N. Luktarhan, X. Jia, L. Hu, and N. Xie, "Multi-stage attack detection algorithm based on hidden Markov model," in *Proc. Int. Conf. Web Inf. Syst. Mining*, Oct. 2012, pp. 275–282.
[117] H. A. Kholidy, A. Erradi, and S. Abdelwahed, "Attack prediction models for cloud intrusion detection systems," in *Proc. 2nd Int. Conf. Artif. Intell., Modeling Simulation*, Nov. 2014, pp. 270–275.
[118] H. A. Kholidy, A. M. Yousof, A. Erradi, S. Abdelwahed, and H. A. Ali, "A finite context intrusion prediction model for cloud systems with a probabilistic suffix tree," in *Proc. Eur. Model. Symp.*, Oct. 2014, pp. 526–531.
[119] H. A. Kholidy, A. Erradi, S. Abdelwahed, and A. Azab, "A finite state hidden Markov model for predicting multistage attacks in cloud systems," in *Proc. IEEE 12th Int. Conf. Dependable, Autonomic Secure Comput.*, Aug. 2014, pp. 14–19.
[120] P. Holgado, V. A. Villagra, and L. Vazquez, "Real-time multistep attack prediction based on hidden Markov models," *IEEE Trans. Depend. Sec. Comput.*, vol. 17, no. 1, pp. 134–147, Jan. 2020.
[121] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 260–269, Apr. 1967.
[122] Y. Zhang, D. Zhao, and J. Liu, "The application of Baum-Welch algorithm in multistep attack," *Sci. World J.*, vol. 2014, May 2014, Art. no. 374260.
[123] L. R. Welch, "Hidden Markov models and the baum-welch algorithm," *IEEE Inf. Theory Soc. Newslett.*, vol. 53, no. 4, pp. 10–13, Dec. 2003.
[124] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Nov. 2012.
[125] W. Zegeye, R. Dean, and F. Moazzami, "Multi-layer hidden Markov model based intrusion detection system," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 1, pp. 265–286, Dec. 2018.
[126] W. K. Zegeye, "Multi-stage attack detection using layered hidden Markov model intrusion detection system," in *Proc. Int. Found. Telemetering*, 2019, pp. 1–10.
[127] T. Shawly, A. Elghariani, J. Kobes, and A. Ghafoor, "Architectures for detecting interleaved multi-stage network attacks using hidden Markov models," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2316–2330, Oct. 2021.
[128] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, pp. 1–38, Feb. 1977.
[129] D. Hsu, S. M. Kakade, and T. Zhang, "A spectral algorithm for learning hidden Markov models," *J. Comput. Syst. Sci.*, vol. 78, no. 5, pp. 1460–1480, 2012.
[130] A. A. Sá, A. O. Andrade, A. B. Soares, and S. J. Nasuto, "Estimation of hidden Markov models parameters using differential evolution," in *Proc. Conv. Commun., Interact. Social Intell.*, vol. 1, 2008, pp. 51–56.
[131] A. Anandkumar, D. Hsu, and S. M. Kakade, "A method of moments for mixture models and hidden Markov models," in *Proc. Conf. Learn. Theory*, 2012, pp. 33.1–33.34.
[132] T. Chadza, K. G. Kyriakopoulos, and S. Lambotharan, "Analysis of hidden Markov model learning algorithms for the detection and prediction of multi-stage network attacks," *Future Gener. Comput. Syst.*, vol. 108, pp. 636–649, Jul. 2020.
[133] S. Shin, S. Lee, H. Kim, and S. Kim, "Advanced probabilistic approach for network intrusion forecasting and detection," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 315–322, 2013.
[134] A. Saaudi, Y. Tong, and C. Farkas, "Probabilistic graphical model on detecting insiders: Modeling with SGD-HMM," in *Proc. 5th Int. Conf. Inf. Syst. Secur. Privacy*, 2019, pp. 461–470.
[135] H. Brahmi and S. Ben Yahia, "Discovering multi-stage attacks using closed multi-dimensional sequential pattern mining," in *Proc. Int. Conf. Database Expert Syst. Appl.*, Aug. 2013, pp. 450–457.
[136] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc. 17th Int. Conf. Data Eng.*, 2001, pp. 215–224.
[137] Y. Lv, S. Xiang, J. Geng, Y. Li, and C. Xia, "An alert correlation algorithm based on the sequence pattern mining," in *Proc. IEEE Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Dec. 2015, pp. 1146–1151.

[138] M. Xian and Y. Zhang, "A privacy-preserving multi-step attack correlation algorithm," in *Proc. IEEE Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, Oct. 2016, pp. 1389–1393.

[139] K. E. Emam and F. K. Dankar, "Protecting privacy using k-anonymity," *J. Amer. Med. Informat. Assoc.*, vol. 15, no. 5, pp. 627–637, 2008.

[140] M. Marchetti, M. Colajanni, and F. Manganiello, "Framework and models for multistep attack detection," *Int. J. Secur. Appl.*, vol. 5, no. 4, pp. 73–90, 2011.

[141] F. F. Daneshgar and M. Abbaspour, "Extracting fuzzy attack patterns using an online fuzzy adaptive alert correlation framework," *Secur. Commun. Netw.*, vol. 9, no. 14, pp. 2245–2260, Sep. 2016.

[142] A. Saaudi, Z. Al-Ibadi, Y. Tong, and C. Farkas, "Insider threats detection using CNN-LSTM model," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2018, pp. 94–99.

[143] C. Cipriano, A. Zand, A. Houmansadr, C. Kruegel, and G. Vigna, "Nexat: A history-based approach to predict attacker actions," in *Proc. 27th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 2011, pp. 383–392.

[144] S. Benferhat, A. Boudjelida, and K. Tabia, "Revising the outputs of a decision tree with expert knowledge: Application to intrusion detection and alert correlation," in *Proc. IEEE 24th Int. Conf. Tools Artif. Intell.*, vol. 1, Nov. 2012, pp. 452–459.

[145] M. Soleimani and A. A. Ghorbani, "Multi-layer episode filtering for the multi-step attack detection," *Comput. Commun.*, vol. 35, no. 11, pp. 1368–1379, Jun. 2012.

[146] A. Pecchia, D. Cotroneo, R. Ganesan, and S. Sarkar, "Filtering security alerts for the analysis of a production SaaS cloud," in *Proc. IEEE/ACM 7th Int. Conf. Utility Cloud Comput.*, Dec. 2014, pp. 233–241.

[147] D. Cotroneo, A. Paudice, and A. Pecchia, "Automated root cause identification of security alerts: Evaluation in a SaaS cloud," *Future Gener. Comput. Syst.*, vol. 56, pp. 375–387, Mar. 2016.

[148] Y.-C. Chang and S.-D. Wang, "The concept of attack scenarios and its applications in Android malware detection," in *Proc. IEEE 18th Int. Conf. High Perform. Comput. Commun., IEEE 14th Int. Conf. Smart City, IEEE 2nd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2016, pp. 1485–1492.

[149] C. Feng, S. Wu, and N. Liu, "A user-centric machine learning framework for cyber security operations center," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2017, pp. 173–175.

[150] M. D. Mauro and C. D. Sarno, "Improving SIEM capabilities through an enhanced probe for encrypted skype traffic detection," *J. Inf. Secur. Appl.*, vol. 38, pp. 85–95, Feb. 2018.

[151] F. Manganiello, M. Marchetti, and M. Colajanni, "Multistep attack detection and alert correlation in intrusion detection systems," in *Proc. Commun. Comput. Inf. Sci.*, vol. 200, Aug. 2011, pp. 101–110.

[152] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1285–1298.

[153] Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini, "Tiresias: Predicting security events through deep learning," in *Proc. Conf. Comput. Commun. Secur.*, 2018, pp. 592–605.

[154] D. Zhao, J. Liu, J. Wang, W. Niu, E. Tong, T. Chen, and G. Li, "Bidirectional RNN-based few-shot training for detecting multi-stage attack," 2019, *arXiv:1905.03454*.

[155] P. Zhou, G. Zhou, D. Wu, and M. Fei, "Detecting multi-stage attacks using sequence-to-sequence model," *Comput. Secur.*, vol. 105, Jun. 2021, Art. no. 102203.

[156] F. J. Abdullayeva, "Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm," *Array*, vol. 10, Jul. 2021, Art. no. 100067.

[157] A. A. Ramaki, M. Amini, and R. E. Atani, "RTECA: Real time episode correlation algorithm for multi-step attack scenarios detection," *Comput. Secur.*, vol. 49, pp. 206–219, Mar. 2015.

[158] A. A. Ramaki and A. Rasoolzadegan, "Causal knowledge analysis for detecting and modeling multi-step attacks," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 6042–6065, Dec. 2016.

[159] M. Bateni, A. Baraani, and A. Ghorbani, "Using artificial immune system and fuzzy logic for alert correlation," *Int. J. Netw. Secur.*, vol. 15, no. 3, pp. 190–204, 2013.

[160] H. H. W. Hua, M. M. Siraj, and M. M. Din, "Integration of PSO and K-means clustering algorithm for structural-based alert correlation model," *Int. J. Innov. Comput.*, vol. 7, no. 2, pp. 34–39, 2017.

[161] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.

[162] X.-L. Tao, L. Shi, F. Zhao, S. Lu, and Y. Peng, "A hybrid alarm association method based on AP clustering and causality," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–10, Mar. 2021.

[163] D. Dueck and B. J. Frey, "Non-metric affinity propagation for unsupervised image categorization," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[164] L. Wang, Z. Qu, Y. Li, K. Hu, J. Sun, K. Xue, and M. Cui, "Method for extracting patterns of coordinated network attacks on electric power CPS based on temporal–topological correlation," *IEEE Access*, vol. 8, pp. 57260–57272, 2020.

[165] Y. Lin, Z. Chen, C. Cao, L.-A. Tang, K. Zhang, W. Cheng, and Z. Li, "Collaborative alert ranking for anomaly detection," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1987–1995.

[166] G. Chen, Y. Zhang, and C. Wang, "A wireless multi-step attack pattern recognition method for WLAN," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7068–7076, Nov. 2014.

[167] B. Jasiul, M. Szpyrka, and J. Sliwa, "Malware behavior modeling with colored Petri nets," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.*, Nov. 2014, pp. 667–679.

[168] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, and F. J. Aparicio-Navarro, "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Gener. Comput. Syst.*, vol. 89, pp. 349–359, Dec. 2018.

[169] V. Papataxiarhis and S. Hadjiefthymiades, "Event correlation and forecasting over multivariate streaming sensor data," 2018, *arXiv:1803.05636*.

[170] Y. Djemaiel, B. A. Fessi, and N. Boudriga, "Using temporal conceptual graphs and neural networks for big data-based attack scenarios reconstruction," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2019, pp. 991–998.

[171] B. K. Raju and G. Geethakumari, "Event correlation in cloud: A forensic perspective," *Computing*, vol. 98, no. 11, pp. 1203–1224, Nov. 2016.

[172] M. Wu and Y. Moon, "Alert correlation for cyber-manufacturing intrusion detection," *Proc. Manuf.*, vol. 34, pp. 820–831, Jan. 2019.

[173] A. Sapegin, D. Jaeger, F. Cheng, and C. Meinel, "Towards a system for complex analysis of security events in large-scale networks," *Comput. Secur.*, vol. 67, pp. 16–34, Jun. 2017.

[174] G. Yang, L. Cai, A. Yu, and D. Meng, "A general and expandable insider threat detection system using baseline anomaly detection and scenario-driven alarm filters," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun., 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 763–773.

[175] (Accessed: Apr. 19, 2021). *DARPA 1999 Dataset*. [Online]. Available: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset

[176] (Accessed: Apr. 19, 2021). *DARPA 2000 Dataset*. [Online]. Available: https://www.ll.mit.edu/r-d/datasets/(2000)-darpa-intrusion-detection-scenario-specific-datasets

[177] (Accessed: Apr. 19, 2021). *DARPACGP Dataset*. [Online]. Available: https://www.ll.mit.edu/r-d/datasets/cyber-grand-challenge-datasets

[178] (Accessed: Apr. 19, 2021). *DEFCON Dataset*. [Online]. Available: https://www.defcon.org/html/links/dc-torrent.html

[179] (Accessed: Oct. 6, 2021). *The 2008 UCSB International Capture the Flag (iCTF)*. [Online]. Available: https://ctftime.org/ctf/5/

[180] G. Vigna, "Teaching network security through live exercises," in *Proc. IFIP World Conf. Inf. Secur. Educ.* New York, NY, USA: Springer, 2003, pp. 3–18.

[181] (Accessed: Oct. 6, 2021). *Capture Files From Mid-Atlantic CCDC*. [Online]. Available: https://www.netresec.com/?page=MACCDC

[182] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, May 2012.

[183] I. Sharafaldin, A. H. Lashkari, and A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, Jan. 2018.

[184] B. Lindauer, "Insider threat test dataset," CERT, New Delhi, India, Tech. Rep., Sep. 2020.

[185] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014.

[186] P. Ning, Y. Cui, and D. S. Reeves, "Constructing attack scenarios through correlation of intrusion alerts," in *Proc. 9th ACM Conf. Comput. Commun. Secur. (CCS)*, 2002, pp. 245–254.

[187] (Accessed: Jun. 22, 2021). *KDD Cup 1999 Data*. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[188] *NSA Datasets*. Accessed: Oct. 6, 2021. [Online]. Available: https://www.westpoint.edu/centers-and-research/cyber-research-center/data-sets
[189] *Contagio Malware Dump*. Accessed: Oct. 10, 2021. [Online]. Available: https://www.impactcybertrust.org/dataset_view?idDataset=1273
[190] (Accessed: Jun. 22, 2021). *LBNL Dataset*. [Online]. Available: http://www.icir.org/enterprise-tracing/
[191] (Accessed: Jun. 22, 2021). *Capture Files From 4SICS Geek Lounge*. [Online]. Available: http://www.icir.org/enterprise-tracing/
[192] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.

IGOR KOTENKO (Senior Member, IEEE) received the graduate degree (Hons.) from the St. Petersburg Academy of Space Engineering and St. Petersburg Signal Academy, St. Petersburg, Russia, the Ph.D. degree, in 1990, and the National degree of Doctor of Engineering Science, in 1999. He is currently a Professor of computer science and the Head of the Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation. He is the author of more than 500 refereed publications. He has a high experience in the research on computer network security and participated in several projects on developing new security technologies. For example, he was a Project Leader in the research projects from the U.S. Air Force Research Department, via its the European Office of Aerospace Research and Development (EOARD) Branch, EU FP7, and FP6 Projects, HP, Intel, and F-Secure. His research results were tested and implemented in more than 50 Russian research and development projects. His main research interests include computer network security, network security analysis, intrusion detection, digital right management, machine learning, and data mining.

DIANA GAIFULINA received the B.S. degree from Orenburg State University, Orenburg, Russia, in 2017, and the M.S. degree from ITMO University, St. Petersburg, Russia, in 2019. She is currently a Junior Researcher at the Laboratory of Computer Security Problems, St. Petersburg Federal Research Center. Her main research interests include information security, machine learning, data mining, anomaly detection, and cyber-physical systems.

IGOR ZELICHENOK received the B.S. and M.S. degrees from the St. Petersburg State University of Telecommunications, St. Petersburg, Russia, in 2019 and 2021, respectively. He is currently a Junior Researcher at the Innovation Laboratory of Cybersecurity Research, St. Petersburg Federal Research Center. His research interests include information security, big data, machine learning, and multi-step attacks detection.