Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan

On using contextual correlation to detect multi-stage cyber attacks in smart grids[☆]

Ömer Sen [a,*], Dennis van der Velde [a], Katharina A. Wehrmeister [a], Immanuel Hacker [a], Martin Henze [b,c], Michael Andres [a]

[a] Digital Energy, Fraunhofer FIT, 52062 Aachen, Germany
[b] Security and Privacy in Industrial Cooperation, RWTH Aachen University, 52074 Aachen, Germany
[c] Cyber Analysis & Defense, Fraunhofer FKIE, 53343 Wachtberg, Germany

ARTICLE INFO

Article history:
Received 9 December 2021
Received in revised form 7 April 2022
Accepted 15 June 2022
Available online 23 June 2022

Keywords:
Intrusion detection
Contextual correlation
Situational awareness
Cyber–physical security
Smart grid

ABSTRACT

While the digitization of the distribution grids brings numerous benefits to grid operations, it also increases the risks imposed by serious cyber security threats such as coordinated, timed attacks. Addressing this new threat landscape requires an advanced security approach beyond established preventive IT security measures such as encryption, network segmentation, or access control. Here, detective capabilities and reactive countermeasures as part of incident response strategies promise to complement nicely the security-by-design approach by providing cyber security situational awareness. However, manually evaluating extensive cyber intelligence within a reasonable timeframe requires an unmanageable effort to process a large amount of cross-domain information. An automated procedure is needed to systematically process and correlate the various cyber intelligence to correctly assess the situation to reduce the manuel effort and support security operations. In this paper, we present an approach that leverages cyber intelligence from multiple sources to detect multi-stage cyber attacks that threaten the smart grid. We investigate the detection quality of the presented correlation approach and discuss the results to highlight the challenges in automated methods for contextual assessment and understanding of the cyber security situation.

1. Introduction

To accommodate the increasing penetration by distributed energy resources (DERs) [1], power grids are currently undergoing fundamental changes [2], especially digitization, ultimately evolving into smart grids (SGs) [3]. The emerging paradigm changes within this transition are primarily characterized by interconnecting different stakeholders using information and communication technology (ICT), enabling the remote controllable integration of volatile DERs as well as novel grid components such as heat pumps and electric vehicles [4], which SGs can provide a foundation for the realization [5]. However, the increasing convergence between information and operational technology in the energy sector [6] is also leading to the emergence of a new threat landscape [7] and thus new cyber security challenges [8].

For active and digitized grid operations, this new threat landscape poses new risks of incidents [9] that can result in critical disruptive consequences [10]. To accommodate with this new threat situation, mitigation and countermeasures must be implemented in a holistic security concept that includes not only preventive security-by-design approaches, but also detective and thus reactive measures. In particular, detective measures such as intrusion detection systems (IDSs), which are designed to detect early indicators of an attack [7], can be used to monitor the cyber security posture of the network [11] and provide the basis for determining appropriate response and remediation actions [12]. Since power grids are critical infrastructures whose availability of power supply must be guaranteed at all times, active and operational restrictive response measures to detected cyber incidents must rely on highly accurate detection mechanisms. Therefore, the risk of false-positives in the detection of, e.g., intrusions and attacks by unauthorized persons into the central monitoring and control system within the network perimeter must be avoided. Preventing such intrusions and attacks is associated with fundamental challenges. In addition, legitimate components may

[☆] This is a significantly extended paper based on the work *Towards an Approach to Contextual Detection of Multi-Stage Cyber Attacks in Smart Grids* by Sen et al. presented at the 2021 International Conference on Smart Energy Systems and Technologies (SEST).

[*] Corresponding author.

E-mail addresses: oemer.sen@fit.fraunhofer.de (Ö. Sen), dennis.van.der.velde@fit.fraunhofer.de (D. van der Velde), katharina.wehrmeister@fit.fraunhofer.de (K.A. Wehrmeister), immanuel.hacker@fit.fraunhofer.de (I. Hacker), henze@cs.rwth-aachen.de (M. Henze), michael.andres@fit.fraunhofer.de (M. Andres).

https://doi.org/10.1016/j.segan.2022.100821
2352-4677/© 2022 Elsevier Ltd. All rights reserved.

perform permitted operations within the network, but their payload semantics lead to network-damaging consequences, possibly from undetected compromise of the component itself. Consequently, it is not sufficient to secure and monitor the system from the traditional, selective, domain-specific perspective such as the communication or process level of the infrastructure that is involved. Rather, in order to gain complete situational awareness of the system, both the communication and the process data transmitted via the industrial control network (e.g., measured values and control commands) must be checked for plausibility and indications of possible attack traces [13] in order to detect advanced attacks in time.

Timely detecting advanced attacks requires the contextual correlation of indicators of an attack from different components [14], inspecting different domains of the network [15], in particular process critical data flows [16], and temporal developments that unfold over time [13]. The management of various security-related data from different sources is usually addressed with approaches based on Security Information and Event Management (SIEM) systems that perform real-time traffic analysis, early detection of attack-related events, and event correlation [17]. Combining the traditional approach of using ICT layer data with utilizing process network data can be used to expand situational awareness of cyber security events in terms of intrusion, propagation, and impact. In particular, due to the static network structure, deterministic traffic, and physically constrained process information in the process network, there are particular advantages in detecting valuable attack-related indicators based on implausible events and anomalies in the process environment [18]. However, the additional domains to consider when analyzing the cyber security situation also means additional effort to not only correlate cross-domain threat intelligence, but also manage it efficiently.

To remedy security issues in SGs, different streams of research address the challenges of automatically analyzing large amounts of cyber threat data. In addition to preventive security concepts enhancing the network cyber security [19] such as security-by-design principles including, for instance, encryption, access control and network segmentation [20], detective defense measures are essential [21] as a complementary part of any comprehensive security concept. This issue becomes especially evident when considering the large amount of legacy components in traditional power grids [22] with limited power resources [23], where security solutions could add a large overhead to their performance and jeopardize operations. Therefore, passive, non-intrusive countermeasures are additionally needed in power grid security designs. In particular, an approach of physical consistency checking at the substation level has been proposed to validate process data according to a set of constraints, thus noticing when an individual substation enters a "bad state" that represents, e.g., a physical instability [13]. Further research aimed to reduce the false positive rate of rule-based IDS solutions by correlating different IDS events to create attack scenarios and using machine learning to teach a system which attacks reported by an IDS is likely to be genuine [24]. Moreover, a trust visualization to help operators in viewing an ICT network and its nodes was developed [25], including security concerns at each node, thus providing operators with an overview of a smart grid and the geographical location of a possible attacker [26]. Using a situational awareness approach where sensors distributed in the SG relay relevant information to a command center (similar to a SIEM system), event correlation and integrity checking can be used to detect complex attacks [27].

Despite these efforts, contextual assessment and reconstruction of security incidents, especially in SGs, remains a challenging area of research. Most importantly, limiting knowledge acquisition to events from the same source and not considering alarms from other security systems or logs from other ICT network components excludes additional information from different perspectives [28], increases intrinsic bias supporting false-positives [29], and reduces capabilities to identify potential false-negatives [30]. A limited knowledge base can lead to limited perception and assessment of potentially wide-ranging cyber incidents. For threats that impact multiple domains simultaneously, such as SG's information and operational landscape, additional information from domain-specific knowledge of power grids, combined with cyber threat insights, can enrich detection. In particular, domain-specific knowledge in the form of validation of data point authenticity and integrity, data flow, communication paths and routes, legitimate operation of assets, and plausibility of data semantics can provide an additional perspective for a holistic and global view of the cyber–physical situation.

To address the challenges of detecting multilevel cyber attacks from a holistic perspective that includes not only the communication level but also the process level by leveraging domain-specific attributes of attack indicators, we propose in this paper a framework for context-based cross-domain correlation of various indicators to enable situational awareness of cyber security in SG. To this end, we present a SIEM-based system for the **d**etection **o**f **m**ulti-stage **c**oordinated **a**ttacks (DOMCA) to identify the appropriate attack evolution and strategy. More precisely, our contributions in this paper are:

1. We present and highlight current cyber security issues in SG and appropriate countermeasures related to detection mechanisms (Section 2).
2. To identify attack actions at a meta-level, we propose an event correlation mechanism based on cyber threat observations (Section 3.3).
3. We present and describe a detection method to identify the corresponding attack strategy based on the detected attack actions (Section 3.4).
4. We demonstrate and discuss the detection quality of our proposed approach alongside simulated cyber attacks on power grids (Sections 4 and 5).

A preliminary version of this paper appears in the proceedings of the 2021 International Conference on Smart Energy Systems and Technologies (SEST) [31]. We extend and improve on our previous work in the following ways: First, we added a dedicated discussion of related work in Section 2.3, where we now provide a more detailed and broader introduction into the research landscape of multi-stage attack correlation, specifically discuss the Dempster Shafer Theory (DST), and finally derive our problem analysis. Second, we added a formal description of the approach underlying DOMCA's event correlation in Section 3.3 and strategy correlation in Section 3.4. Third, we provide more details on the methodology underlying our evaluation of DOMCA, specifically w.r.t. attack modeling and implementation (Section 4.2) as well as evaluation criteria (Section 4.3). Fourth, we extended our presentation of the results of the evaluation of DOMCA's classification accuracy to further assess other conditions on detection quality, such as the level of information in the observation base (Section 4.4) and added a dedicated discussion of the implications of these evaluation results for the timely detection of multi-stage cyber attacks in SGs (Section 5). Finally, we generally provide more details on various aspects of the design, implementation, and evaluation of DOMCA throughout our extended paper.

The remainder of this paper is structured as follows. In Section 2, we lay the foundation for this work by discussing cyber security-related issues with power grids and arguing the need for detective countermeasures, before we study the research landscape in multi-stage attack correlation and describe our problem analysis. We then introduce our framework DOMCA for the detection of multi-stage coordinated attacks on smart grids in Section 3. To evaluate and investigate the performance of DOMCA, we present our investigation procedure, our attack modeling and implementation and evaluation metrics, as well as our results in Section 4. Finally, we discuss our results in Section 5 and conclude this work with Section 6.

2. Cyber security in smart grids

In this section, we present the state of cyber security issues in process networks while laying the foundation for detection and correlation mechanisms against security incidents.

2.1. Cyber security and power grids

The exchange of process data and the networking of several stakeholders involved is enabled by the integration of ICT into power grids [32], in which, for example, measured values from sensors or control commands to actuators are passed between Remote Terminal Units (RTUs) and Master Terminal Unit (MTU) within Supervisory Control and Data Acquisition (SCADA) systems [33]. Here, the SCADA system is used to monitor the state of the grid based on the acquired data and performs a grid safety check to potentially trigger alarms in the event of grid disturbances (e.g., critical load, voltage threshold violations, power quality disturbances) [5]. In addition, higher-level decision and optimization functions are utilized for optimal grid operation, such as optimal power flow calculations to determine appropriate control commands or other operational decisions [13]. The performance of such an operation is primarily to optimize the grid state, taking into account stability, resource utilization and flexibility constraints.

Traditional process networks, which have a clearer separation of IT and Operational Technology (OT) components, created a "barrier" to unauthorized third parties due to the isolated, proprietary nature of legacy components and technologies. The digitization of these process networks through the increasing integration of ICT is leading to the breaking down of the "barrier" [32], resulting not only in the interconnection of various network assets and actors [34], but also in the emergence of a new cyber threat landscape [35]. In particular, cyber attacks can exploit the new access points, traversable communication paths, and vulnerabilities [36] to disrupt or damage the power grid [37] by intercepting, manipulating, and spoofing communications between its SCADA components on a large scale [38].

For instance, Iran's nuclear power plant was attacked by a cyber attack called Stuxnet, which entered the internal control system via removable media and spread laterally to the corresponding field device of the centrifuges to disrupt the stable operating point [39]. In addition, a Trojan similar to Stuxnet called Havex compromised more than 1000 energy company assets in 84 countries between 2013 and 2014 [39]. One particular cyber incident in power grids occurred in 2015 and 2016 in Ukraine [40], where bulk power generation plants were the target of coordinated cyber attacks to disrupt the power supply, resulting in a temporary blackout for more than 200,000 customers [39].

2.2. Contextual detection of cyber incidents

For the detection of coordinated cyber attacks, dedicated solution for intrusion detection is needed, which IDS takes care of automating the process of intrusion detection through various approaches [41]. The different approaches can be distinguished primarily in that they detect either attack indicators based on normal operation (anomaly-based) [42] or attack signatures (misuse-based) [43] or their combined knowledge [24]. In doing so, traditional IDSs approaches focus on monitoring the ICT network [44] and/or its host components (e.g., login attempts, network scans, suspicious protocol traffic, or syslog) [13] and neglect the OT components and process semantics of the power grid [45].

Extending detection capabilities beyond local detection of traditional IDS requires contextual detection based on a SIEM system [46] that combines features such as security data collection and consolidation, long-term data storage, automation of analysis and reporting, and real-time monitoring and correlation of events from multiple data sources [47]. In this context, dealing with extensive data sources requires efficient data aggregation, collecting log and event data from various sources (e.g. IDS or firewalls) [48]. Also, normalization of data to a common format that can also be managed without losing information or corrupting the fields is required [49], particularly synchronization of timestamps [50], providing comparable and accessible characteristics of the data for processing and correlation [51]. Subsequently, higher-level inferences can be made based on these functionalities, such as identifying C2 infrastructures by identifying compromised hosts in the ICT network, where either the malicious activity of one host provides evidence that the host is compromised, or that a compromised host could perform malicious activity on other hosts that may also be compromised as a result of that action [24].

Multi-stage cyber attack detection requires correlation and inference approaches that not only classify possible compromised nodes, but also infer features and relationships between entities and events on the sequential attack process by using available information contextually. In particular, hierarchical unidirectional dependencies between attack steps, their transitions, involved components, and attack targets within the system can be modeled by attack graphs [52]. By modeling the graph such that the nodes with multiple successors or predecessors represent conditional attack stages of an overarching campaign, strategies with multiple different succession paths can be represented via the attack graphs [52]. In the context of offensive security approaches, this method requires detailed knowledge of the attacker's perspective, goals, tactics, techniques, and patterns, as well as the victim's network, including system and security specifications and vulnerabilities [29]. For example, a structural approach to modeling sequential processes of cyber attacks in a multi-stage manner can be achieved through the kill-chain modeling concept, which enables structured modeling of multi-stage attacks aimed at disrupting or destroying vital processes or devices [53]. There are several variants of the modeling approach, with the original approach providing seven steps, including gaining access to and information about the target system, developing and testing new capabilities on the compromised targets, exploiting vulnerabilities and moving laterally in the network, building C2 infrastructure, and performing actions (e.g., disrupting network operations) [54].

Specifically, following the kill-chain concept [53], a cyber attack would begin with reconnaissance, which consists of observing the target system from the outside and gathering publicly available knowledge about it to uncover vulnerabilities and other useful information. Once enough information has been gathered, the attack moves to its next phase, which is to enable successful intrusion of the target system, for example, by weaponizing an inconspicuous file to make it usable for initial access to the system, such as a malicious macro in a document file. Afterwards, to gain initial access to the target system, the attacker proceeds to deliver the weaponized file to the target via, for example, a phishing email, exploit identified vulnerabilities, and gain a foothold by installing malware or modifying existing system functions for C2 and granting additional access rights via privilege escalation. Based on the established C2 channel to control the infiltrated victim hosts, lateral movements are made in the target system to decide, e.g., which devices with benefit to the attacker's overall goal could be compromised next. After gaining sufficient influence over the target system, the attacker pursues its goal by conducting a coordinated cyber attack through the established C2-overlay network to inflict damage on the system.

2.3. Related work

In order to explore reliable and accurate methods for detecting cyber attacks, several research efforts have been conducted to address the challenges of automatically analyzing large amounts of cyber information.

2.3.1. Contextual detection

The research field of multi-step attack detection methods have been explored by many works, such that different methods can be categorized, for which we refer to the corpus of a systematic survey performed by Navarro et al. [55]:

(i) Similarity-based: attack construction based on the degree of similarity determined from attack indicators (e.g., progressive construction, scenario clustering, or anomaly detection).

(ii) Causal correlation: assuming a causal relationship between attack actions, reconstructing a multi-stage attack sequence (e.g., preconditions and consequences, statistical inference, or model matching).

(iii) Structural: incorporating the underlying infrastructure, e.g., the ICT network, possible attack paths are predicted by projecting attack indicators onto the network model.

(iv) Case-based: based on known and well-documented attack cases, the received indicators are mapped to these cases.

(v) Mixed: combination of the above methods without focusing on a unique method (e.g., time-window aggregation, offline only correlation, online/real-time correlation, and/or ontology-based correlation).

In the area of causal correlation and statistical inference, common approaches to multi-stage attack detection are often based on Hidden Markov, Pertrinet, or Bayesian network models. One of these approaches, specifically the Bayesian network model, is used in the work of Kavousi and Akbari [56], which presents an offline attack pattern detection component and an online alarm correlation component. After preprocessing the alarms, which reduces the information redundancy, the causal relationships between alarm types from the historical aggregated and preprocessed alarms are correlated with structural constraints to accelerate the Bayesian learning process. The Bayesian network learning algorithm infers the causal relationships between alarm types using some conditional independence tests to extract the attack transition patterns.

Another approach bases on statistical inference and mining [57], which includes a correlation scheme based on a combination of statistical and stream mining techniques to aggregate alerts based on the similarity of their alert types, and an episode mining algorithm to determine the possible combinations of alerts. The method works in real time by extracting critical episodes from sequences of alerts that could be part of multi-stage attack scenarios. A causal correlation matrix is used to encode the correlation strength between alert types in attack scenarios.

In the direction of model- and case-based detection, approaches such as the one proposed by Liang [58] use scenario generation of security-related situations based on historical data. In particular, this work pursues the development of a security data collection platform that integrates data from different resources and in different formats for security scenario modeling. Scenario modeling is based on data mining of historical data to extract security-related information such as faults or warnings, which are then clustered into groups with similar characteristics such as name, reason, impact, and response methods.

Also, in the area of constructing multi-stage attack scenarios, the approach proposed by Bajtoš et al. [59] uses a method based on similarity-based correlation. Their method consists of aggregating event logs from the dataset into aggregated alerts, computing similarity using source and destination IP addresses and source and destination network ports, and pattern searching using a correlation matrix to create a directed graph. In this process, alerts are correlated only if they occur in quick succession in a given time window.

Incorporating correlation processes and semantics, an ontology-based method [60] can be used to fuse different types of information within the same data model. Based on a procedure that includes fusion of detected events, their validation, scenario reconstruction, and pattern mining, attack patterns are extracted from normalized datasets using an attribute-based induction algorithm. At its core, the correlation approach is based on a machine learning paradigm, such as learning from examples, which extracts generalized data from the original data, and hypergraph theory, which visualizes the frequent elements to identify unknown attack patterns.

Similarly, Ahmadinejad et al. [61] examined the hybrid model method used to correlate known and unknown attack scenarios. Specifically, the model consists of an attack graph-based method for correlating alerts triggered for known attacks and a similarity-based method for correlating alerts triggered for unknown attacks that could not be correlated with the previous component. The alerts are correlated based on an attack graph, specifically a queue graph, if the corresponding exploits in the attack graph have a causal relationship.

In terms of prioritizing intrusion alerts and detecting attacks using post-correlation analysis, ACSAnIA [62] is an approach based on a comprehensive intrusion alert analysis system. This approach relies on a metric for prioritizing alerts based on anomalous behavior, which is used for clustering correlated alerts within a data structure to represent robust attack patterns. Based on a local outlier factor algorithm for grouping alarms into a meta-alarm that are clustered into groups, the discovery of similar attack patterns by using frequent pattern mining with a graph mining algorithm is performed.

Aiming to address the challenge of novel attack techniques that leave no traces in the victim's file system and thus provide no artifacts for analysis by conventional attack analysis mechanisms, Chamotra and Barbhuiya [63] propose a network of interactive honeypots as a tool for large-scale detection and collection of multi-stage attacks. Based on interactive honeypots that provide operating systems and services to potential attackers, a high value of attack data is collected from real attack situations. Using this data, the authors have developed a method to characterize the multi-stage attacks using semantically meaningful attack graphs based on abstracted events to model attacks that exploit zero-day vulnerabilities.

Using proven taxonomies and categorizing multi-stage attacks via the MITRE ATT&CK matrix [64], the approach of Takey et al. [65] uses machine learning for early detection of multi-stage attacks in an online process. The authors use a runtime engine that reads process events from the Windows operating system to determine if the executable is malicious, and extract the features from the static binary and pass them to machine learning for malicious executable detection. The machine learning model takes the features as input and labels the executable as malicious or benign, detecting the case that an executable is malicious by predicting the phases the malware executes during the attack according to the MTIRE ATT&CK matrix.

Other research in the direction of inductive/deductive correlation such as the work of Moya et al. [66] proposes an approach based on indexing potential hostile behavior as part of a coordinated attack on the monitoring and control data flow in the process network and storing the generated indexes in a knowledge base. The correlation of attack indicators is based on an induction and deduction algorithm, where attack consequences are first computed by the induction algorithm based on the corrupted measurements and then further processed by the deduction algorithm to determine the attack indicators that are causal related to the consequences. Based on the processed attack indicators, the knowledge base scans and reasons the indicator in the database to calculate suitable defense strategies. The proposed correlation method focuses on monitor-control data, while multi-stage cyber attacks in their phases of intrusion, lateral movement, and C2 are not considered and thus are not part of the attack evolution detection.

Taking a different perspective, the work of Ten et al. [67] follows a model-based correlation using attack trees for impact analysis, leveraging probabilistic measures such as vulnerability index to account for attempted intrusion and policy enforcement. The modeled attack tree includes advisory objectives and cyber security conditions that represent the likelihood of successful compromise given technical countermeasures and password policy enforcement. Attributes of the assessment in the proposed approach are primarily port auditing, which computes the risk level of compromise based on the port policy, and, by analogy, password strength assessment, which is determined by the total combination of character types and their length. Considering a broader view of cyber attacks and potential propagation behavior, the correlation method in this paper emphasizes port auditing and password policy enforcement, while polymorphic attack vectors are not normalized to established attack catalogs such as the MITRE ATT&CK matrix.

Based on a decentralized architecture, Appiah-Kubi et al. [68] propose a four-phase method for detecting coordinated attacks and proposing countermeasures using a collaborative correlation approach. The first two phases are characterized by the continuous monitoring of the network state by the placed network-based IDS and the sharing of the observation in a multi-agent communication layer via a link-drop-max-consensus protocol in the last two phases (sharing the maximum value among agents without repeating already shared information). Within the distributed architecture, each agent is designed to predict the confidence level of the recommendation's target node based on attack patterns (weighted average of metrics that correlate protocol patterns), the criticality of the node's load (criticality indices according to a lookup table), and software correlation (measures the extent to which the reporting node's affected device is related to that of the receiving node). However, when introducing a collaboration-based correlation method, the proposed approach is limited to detecting a finite set of specific attack vectors and neglects the multi-stage attack correlation of complex attack sequences.

Finally, the work of Aparicio-Navarro et al. [69] addresses the challenge of uncertainty in the correctness of alerts and other security-related information and has developed a detection method based on DST [70]. Based on the combined use of different metrics from multiple layers of the protocol stack to perform detection, DST is used to fuse the evidence provided by the metrics [71]. In particular, a fuzzy cognitive map (prediction and decision making process) is integrated to incorporate the contextual information into the recognition process by providing the appropriate adjustments to the DST belief values assigned before the data fusion process.

2.3.2. Dempster shafer theory

Dempster Shafer Theory (DST) is seen as a generalization of traditional Bayesian probability theory [70], making it possible to assign a probability to sets of statements rather than individuals. This allows the combination of evidence from multiple sources without a priori knowledge, i.e., a priori probability distributions, about system states [72]. While traditional probability theory enforces that the unknown probability is uniformly distributed and assigns a value in the interval [0, 1] to each possible combination, DST uses lower (belief function) and upper (plausibility function) bounds to support the hypothesis, which allows quantification of the unknowns by determining the confidence level that a given sequence of evidence is correctly interpreted [24]. The definition of a belief and plausibility function depends on the frame of discernment and the mass distribution function. Subsequently, the frame of discernment, denoted θ, is defined as the set of disjoint hypotheses of interest, denoted x. In this regard, the mass distribution function is defined as $m : 2^{\theta} \rightarrow [0, 1]$, where m distributes the belief over the power set of θ and has the following properties:

(i) $\forall x \subseteq \theta : m\left(x\right) \geq 0$
(ii) $m\left(\varnothing\right) = 0$
(iii) $\sum_{x \subseteq \theta} m(x) = 1$

Based on these definitions, we obtain the belief of a hypothesis x as the sum of the masses that are a subset of x, i.e., $Bel\left(x\right) = \sum_{y \subseteq x} m(y)$, where $x \subseteq \theta$, thus $Bel\left(\varnothing\right) = 0, Bel\left(\theta\right) = 1$. The belief function $Bel\left(\theta\right)$ can be seen as a measure of confidence that a hypothesis is true given a set of evidence. Accordingly, the plausibility function $Pl\left(\theta\right)$ represents an upper limit of our confidence in a hypothesis, i.e., $Pl\left(x\right) = \sum_{y \cap x \neq \varnothing} m(y)$, where $x \subseteq \theta$, thus $Pl\left(\varnothing\right) = 0, Pl\left(\theta\right) = 1$. The actual probability $P_{dst}(x)$ is contained in the interval $[Bel\left(x\right), Pl(x)]$, thus the distance $\gamma = |Pl\left(x\right) - Bel(x)|$ describes the uncertainty regarding the hypothesis x. If $\gamma = 0$, then the probability $P_{dst}(x)$ determined using the DST corresponds to traditional probability $P_{bayesian}(x)$.

One of the challenges of alert correlation is the missing basis of data for the probability quantification of an attack. Because of that, assigning a priori probabilities to attack indicators is infeasible. Therefore, without being able to quantify the unknown, the alert correlation would be inaccurate, because the impact of events, i.e., the probability that an event indicates an attack, could not be quantified. While other theories can handle epistemic uncertainty such as Subjective Logic [73], Fuzzy Logic [74], and Possibility theory [75], DST has the advantages of a high degree of theoretical development and its relation to traditional probability theory. What further sets DST apart from other approaches, is its ability to combine evidence from multiple sources.

To determine the joint mass function of two independent mass distribution functions m_1, m_2 on the same frame of discernment, the basic DST combination rule can be used, which is defined as

$$m_{1,2}\left(C\right) = \begin{cases} \frac{1}{1-K} \cdot \sum_{A \cap B = C} m_1\left(A\right) \cdot m_2\left(B\right), \text{when } C \neq \varnothing \\ 0, \text{when } C = \varnothing \end{cases}$$

where

$$K = \sum_{A \cap B = \varnothing} m_1\left(A\right) \cdot m_2\left(B\right)$$

K is also called conflict. It is the sum of the masses of all conflicting evidence. The basic DST combination rule assumes independent evidence and may perform poorly when severe conflict is present.

Combining evidence from multiple sources depends on different types of evidence, considering consonant evidence, consistent evidence, arbitrary evidence, and disjunctive evidence [72]. Consonant evidence means that over time obtained information progressively narrows or refines the scope of the evidence set, while consistent evidence means that at least one common element to all subsets exists. Arbitrary evidence means there is no element common to all subsets, while disjunctive evidence means that any two subsets have no elements in common with any other subset. Each of these possible configurations of evidence from multiple sources has different implications for the degree of conflict associated with the situation, where, for example, in the case of disjoint evidence, all sources provide conflicting evidence. Here, K represents the base probability mass associated with the conflict, which is determined by summing the products of the base probability assignments of all sets whose intersection is zero. In addition, the operation of combining evidence from multiple sources also depends on the situation of how the sources are considered in terms of reliability, using, for example, a conjunctive combination operation for considering all evidence as reliable, a disjunctive combination operation for limited reliable evidence, or a compromise between the previous two combination rules. Since C is defined as an appropriate measure of the intersection between evidence A and B, it can be considered as the degree of agreement between the evidence to be combined.

The combination rule can also be viewed as a aggregation method for data fusion to rationally combine and simplify information from independent and diverse sources, ignoring any conflicting evidence through normalization. Any two mass functions associated with the evidences A and B over the same frame of discrimination with at least one common focal element can be combined into a new mass function of C using the DST combination rule. The assumption of independent evidence in IDS, i.e., independent IDS sensors and independent IDS sensor alerts, generally does not hold.

Consider two alerts A_1 and A_2 issued by the same sensor. A_1 gets triggered by an intercepting attack such as Man-in-the-Middle (MITM) attack listening in on network communication and establishing connections. A_2 gets triggered, because the MITM successfully attacked ongoing communication and now tries to actively take part in the network. Not only are A_1 and A_2 issued by the same sensor, but A_2 is also caused by the same activities that caused A_1. Therefore, the two alerts are not independent. Another example that illustrates the issue with conflict in the basic DST combination rule is given by Zadeh [76]. Suppose two inspectors A and B independently evaluate the same situation. Inspector A thinks that it is likely a case of x_A, with a probability of 0.99, or a case of x_C, with a probability of 0.01. Inspector B believes it to be a case of x_B, with a probability of 0.99, or a case of x_C, with a probability of 0.01. Here case x_A and x_B are in conflict with each other. Applying the basic DST combination rule would lead to a probability of 1.0 for case x_C because of the conflict between case x_A and x_B, leading to the rejection of both cases. For these reasons, the application of pure DST has been criticized and a plethora of modified combination rules have been proposed and analyzed.

2.3.3. Problem analysis

As described in Section 2.3.1, a broad range of different approaches exists to detect multi-stage cyber attacks in SG. Addressing the challenge imposed by knowledge acquisition limitations requires consideration of methods for fusing heterogeneous data from disparate sources. Not considering potential knowledge from other domains and process limits the extent to which a potential incident may be understood and assessed. Thus, when detecting complex attacks with data from multiple sources, additional information such as process and topology information from power grids enriches detection by providing a more coherent and global view of the current situation to interpret and infer complex suspicious activities at the system and component level. The challenge here is the transferability of different information to create a knowledge base that can be used not only to link cyber-threat traces to infrastructure but also to identify the attack campaign itself. Many of the related work relies on manually coded knowledge of multi-stage attacks and thus includes human error in the development of signatures. Moreover, related work proposing approaches to detect coordinated cyber attacks is typically limited to selected phases and attack vectors such as MITM, Denial-of-Service (DoS), and replay, and neglect the dynamic and polymorphic behavior of propagating cyber attacks in the phases proposed by MITRE ATT&CK Matrix. It should also be noted that many research papers only work with limited attack datasets such as the DARPA 2000 dataset [77], in which the models presented correspond to the same attacks and do not include many different examples of multi-stage attack models. Furthermore, overly strict context-specific approaches with limited generalizability necessitate a comprehensive and rich knowledge base that covers multiple instances of the same cyber incident with minor differences. This includes taking into account the polymorphic characteristics of attackers, who often do not execute their steps sequentially with the same techniques but perform their steps interchangeably with different techniques. Thus, detecting newly deployed adversary techniques in the context of a multi-stage cyber attack also requires understanding the overall end-to-end activities performed during the attack and, consequently, the targeted strategy. Identifying the appropriate attack strategy requires situational awareness not only in IT but also OT in the operational environment. The understanding and concept of situational awareness [15] is very similar to the concept of contextual awareness targeted by this work. In this work, we aim to discuss the design and subsequent implementation of DOMCA, which performs the correlation of distributed security information to reconstruct a potential cyber incident.

3. SIEM-based detection of multi-stage coordinated attacks

In this section, we present the architecture and overview of our SIEM-based attack detection system DOMCA. A difficult aspect of correlating cyber threat information and process data is accounting for false positives, particularly dealing with the reliability and certainty of evidence within traditional probabilistic correlation approaches [24]. Traditional probabilistic approaches do not consider the issue of certainty in assigning simple probability values to statements or conclusions based on unreliable data [24]. To account for the certainty of the data, and thus to model the confidence level of the conclusions on an appropriate quantification basis, a method is needed that extends the capabilities of traditional Bayesian probabilistic methods. In particular, quantifying probabilities in data that lacks a comprehensive database from real-world cyber incidents due to security or privacy concerns presents additional challenges in dealing with data without considering certainty. Therefore, assigning a priori probabilities for attack indicators in the form of confidence values is not feasible without considering certainty [24]. To address this issue, theories that deal with epistemic uncertainty can be used, such as DST (cf. Section 2.3.2) [72]. In the following, we present the architecture of DOMCA to detect the corresponding attack evolution and strategy based on domain-specific attribution and contextual correlation of cyber incident indicators using DST.

3.1. Framework overview

The core concept of our framework pursues the goal of reconstructing the propagation behavior and intended strategy of the cyber attack based on the observation and structural modeling of the attack (cf. Fig. 1). DOMCA digests indicators captured by the

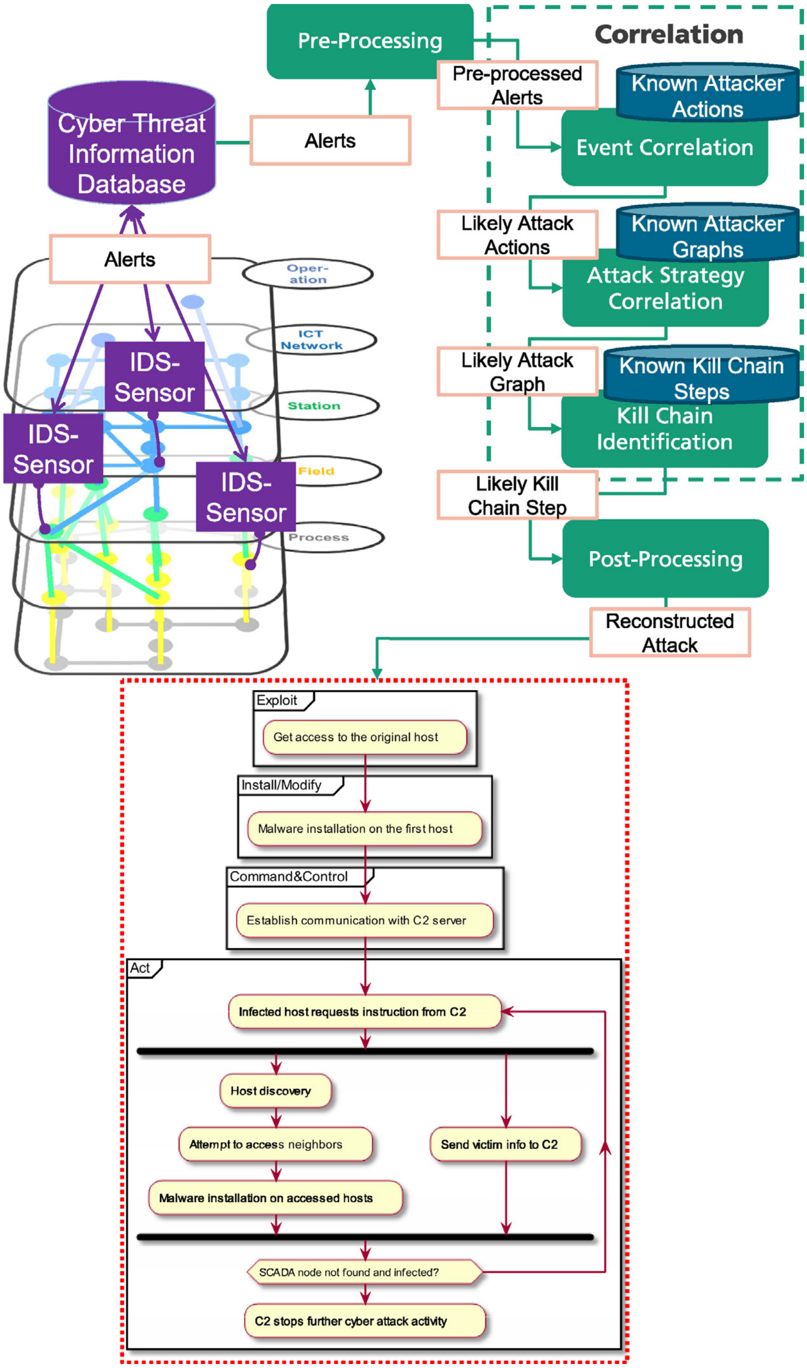

Fig. 1. Structural overview of the presented kill-chain-based correlation and detection system for contextual detection of multi-stage cyber incidents.

pre-processing component from distributed sensors that contain domain-specific information about the process, communication, and operational semantics of the attack actions represented in the attack behavior. This involves normalization processing of the attack indicators within pre-processing component (cf. Section 3.2). In this work, for simplicity, it is assumed that different outputs from multiple source of monitoring logs and attack indicators are generated within the distributed IDS sensors. Furthermore, a central architectural framework at e.g., the operations center level or dedicated cyber security centers such as security operation centers) is envisioned to increase situational awareness at a more global level. The issue of "single point of failure" and increased attack surface due to the central architecture can be addressed by a persistent and secure communication layer, e.g., a distributed ledger communication layer connecting the sensors and the central correlation framework. In particular, a hybrid architecture [78] can be developed based on distributed and centralized design principles [79] combined with other security mechanisms, such as the "moving target" approach [80], which addresses the issue of the "single point of failure" and the lack of global context of the situation. Since the architectural design concept is outside the scope of this paper, this concept will not be discussed in detail in this paper. After the pre-processing component, the *EC* determines possible attack actions based on the pre-processed attack indicators by using custom combination rules of DST in the context of a given set of known possible actions that an attacker could perform (cf. Section 3.3).

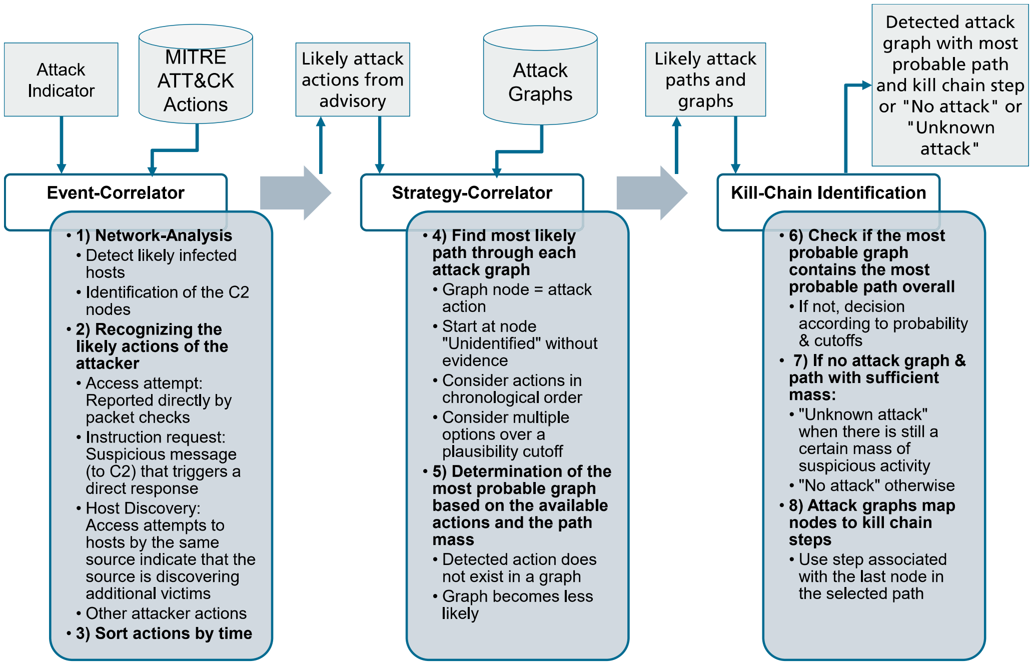
Attack Indicator
MITRE ATT&CK Actions
Likely attack actions from advisory
Attack Graphs
Likely attack paths and graphs
Detected attack graph with most probable path and kill chain step or "No attack" or "Unknown attack"
Event-Correlator
• 1) Network-Analysis
• Detect likely infected hosts
• Identification of the C2 nodes
• 2) Recognizing the likely actions of the attacker
• Access attempt: Reported directly by packet checks
• Instruction request: Suspicious message (to C2) that triggers a direct response
• Host Discovery: Access attempts to hosts by the same source indicate that the source is discovering additional victims
• Other attacker actions
• 3) Sort actions by time
Strategy-Correlator
• 4) Find most likely path through each attack graph
• Graph node = attack action
• Start at node "Unidentified" without evidence
• Consider actions in chronological order
• Consider multiple options over a plausibility cutoff
• 5) Determination of the most probable graph based on the available actions and the path mass
• Detected action does not exist in a graph
• Graph becomes less likely
Kill-Chain Identification
• 6) Check if the most probable graph contains the most probable path overall
• If not, decision according to probability & cutoffs
• 7) If no attack graph & path with sufficient mass:
• "Unknown attack" when there is still a certain mass of suspicious activity
• "No attack" otherwise
• 8) Attack graphs map nodes to kill chain steps
• Use step associated with the last node in the selected path

Fig. 2. Simplified overview of the core EC, SC and kill-chain identification components and their functionalities.

The process of identifying all potentially performed attack actions is then followed by *SC*, which performs the analysis of identifying possible paths in the context of known attack graphs based on DST combination rules. In the context of SC analysis, the consideration of the assigned mass values of the detected attack action and the edges of the attack graph is part of identifying feasible attack strategies based on the current observations (cf. Section 3.4). Following the correlation process, the *kill-chain identification* component is triggered to perform the analysis to determine the most likely attack path and the corresponding graph based on the SC results. In this step, the corresponding kill-chain step associated with the identified attack path structure is determined (cf. Section 3.5). The modularity of this approach enables transferability to other domains characterized by the architecture of industrial control systems (cf. Section 3.6). After the generated attack actions, graph and path, the *post-processing* component performs comprehensive visualization and higher-level processing of the result (cf. Section 3.2). A simplified overview of the core components and their functionalities is shown in Fig. 2. In the following sections, the design of the above process is presented and discussed in more detail.

3.2. Pre-processing

The goal of the pre-processing component is to digest various security-related information from different sources and convert it into a format that is normalized and processable by the other components of DOMCA.

A significant part of the security-related information in the context of this work is data from distributed specification-based IDS within the process network, which provide alerts as well as deviations from the normal characteristics of the system. In particular, the deployed sensors are placed within selected network locations (e.g., SPAN port of switches or network taps), on which basis the traffic is analyzed using the domain-specific attribution according to Table 1. The sensors perform a series of checks where, in the event of a failed check, an alarm is triggered containing the security-related information about the failed check such as check criteria, event information, and packet specification. Specifically, the information relates to the payload of the packet causing the test failure, the payload type (e.g., command, measurement, monitoring, IT, and OT payload), the timestamp of the alert, protocol-related packet information, monitored ICT edge endpoints, and the ID of reporting sensor. These are then consolidated by looping through all packet events to find clusters of events sent at the same time along with the same ICT network edge, represented by a single alert containing the common information. Furthermore, based on the unique sensor identification, the pre-processing component can construct globally unique event IDs by combining the IDs of each packet with the corresponding sensor that reported the event.

Table 1
Domain-specific attribution of alarms within events.

Fields	Description
`IoC`	Participation in an attempt to access a host.
`ADR_FROM_CHECK`	Suspicious source of the message.
`ADR_TO_CHECK`	Suspicious destination of the message.
`CON_CHECK`	The connection over which the packet was sent is not allowed.
`DP_FROM_CHECK`	The packet contains data points that are unexpected for the source host.
`DP_TO_CHECK`	The packet contains data points that are unexpected for the receiver host.
`CYCLE_CHECK`	A message that normally arrives cyclically deviates from its schedule.

An important part of pre-processing is also the identification of consistent communication paths forming traversable messaging routes message multiple hosts (IT and OT components) involved in the cyber attack. This is mainly reconstructed by topological and chronological mapping of packet events based on the timestamps and topology of the monitored network. In this context, sensor coverage and placement play a critical role, potentially affecting detection quality. However, some of the missing information due to missing sensors can be compensated by deduction based on consistent connectivity in the network, the degree of sensor coverage, topological relationship between sensors, and marginal temporal differences of events between pairs. To illustrate this issue, we assume that the last destination node of path p and the start node of path q are connected by an edge

Table 2
Normalized output format of the pre-processing component.

Fields	Description
EVENT_ID	List of event IDs assigned to this message path.
SEND_TIME	Sending time of the original responsible host.
RECEIVE_TIME	Receiving time of the last destination host.
FROM_HOST	Host responsible for the message.
PASSED_HOSTS	Hosts that the message passed through on its way.
TO_HOST	Final destination of the message.

Table 3
List of abstracted cyber attack actions considered within DOMCA.

Action	Description
Network access attempt	Attempt to gain user and, if possible, root access to a host by accessing it over the simulated ICT network.
Offline access attempt	Attempt to gain user and, if possible, root access to a host by accessing it through offline means (e.g., compromised removable media).
Host discovery	Find other hosts on subnets adjacent to a compromised host with at least user access.
Malware installation	Install malware on a compromised host with root access.
Instruction request	Malware-infected host asks a C2 host for instructions.
Send host info to C2	Sending information about an infected host to a connected C2 host.
Send host info to host	Sending information about an infected host to another infected host.
Victim communication	Other communication between infected hosts.

within the ICT network. As p was received before q was sent, and the timestamp difference is within marginal thresholds, it is not necessary to equip the edge between the end of p and the start of q with a sensor. Then, the identified pairs are consolidated by combining data from other stored pairs, forming new paths by concatenating their host paths and failed security checks. Finally, the component clusters the pre-processed events based on the constructed paths and returns them in the format of Table 2.

3.3. Event correlator

In this section, we introduce the EC, which aims to draw conclusions about the likely attacker actions in chronological order based on the pre-processed data. Based on the results of EC, the SC can use the result to identify the attack strategy. In Table 3 we provide an overview of exemplary abstracted cyber attack actions used to form the action set. We have categorized the actions according to the kill-chain concept [53], where the attack actions reflect the steps of reconnaissance, weaponization, delivery, exploitation & installation, and C2 communication. Another part of the normalization process is the mapping of alerts and security-related logs to the established and defined attack actions of the MITRE ATT&CK matrix [64], especially for industrial control systems [81].

The initial analysis of EC begins by identifying suspicious behaviors of potentially infected hosts based on inferences from frequently occurring suspicious events that do not indicate compromise with certainty, such as. (e.g., scanning other nodes' ports, sending packets over unauthorized connections). It is important to note that, in this work, suspicious detected packets are part of the alert or attack indicator portfolio of lower-level IDS sensors, some of which are also assumed to have the capabilities of well-established network-based IDS such as Snort [82]. As part of this process, the EC also pursues the goal of determining the position of a potential C2 coordinator. To perform this identification process, the C2 coordinator is assumed to be characterized as the host from which significant suspicious messages emanate, allowing the structure of the infected hosts' communications to be considered as part of the detection. After potential C2 coordinators are identified, the EC determines individual attacker actions, such as an attempted host access (e.g., to RTU), which may be characterized by an exploratory activity, such as a network scan that identifies vulnerable services on the host, followed by an exploit activity, such as a suspicious login or attempted privilege escalation. Host access attempts are identified based on the listing of the detected access attempt indicators (e.g., network scan, attempted or suspicious login, privilege escalation). The collection of indicators associated with the access attempt is then sorted chronologically, grouped by network host target, and combined into pairs of source and destination hosts in accordance with the detected Indicator of Compromise (IoC). At the end of this process, mass functions are assigned to the potential access attempts. In particular, the mass function is used here as the weight of interest of the observation in the context of the access attempt action, taking into account for the evaluation the type, the time of occurrence and the context with other related access attempts.

Furthermore, we distinguish between local or remote attempts based on the length of the source–destination pair list and the identity of the endpoints. Based on the identified access attempts, network-based access attempts are identified by determining which infected host first attempted to infiltrate another host based on a list of source–destination pairs. This includes detecting a possible malware installation on likely infected hosts, such as a compromised RTUs, by inferring possible actions from the temporal correlation of access attempts and the occurrence of suspicious messages sent by the targeted host. It is possible that the infected host does not immediately request commands from the C2 host after malware installation, but lays low to explore the environment or wait for incoming commands (e.g., RTUs waits for a control action to manipulate data). Detection of these cases can be based on observation of messages that violate acceptable connection paths (e.g., servers outside the process network), combined with observation of C2 hosts sending messages to suspicious hosts after malware installation.

Another interesting attacker action is the communication-dependent behavior of suspicious or compromised hosts (e.g., to gather information about the infiltrated network). For this detection, all pre-processed data associated with suspicious communications from potentially infected hosts is taken into account. For example, an interesting indicator is communication activity that is not part of access attempts or direct C2 communication, but bilateral communication between compromised components, especially if the communication paths are atypical (e.g., horizontal communication between RTUs). The mass functions assigned in the pre-processed data are used to derive the reliability of the detected attack. To account for the predominance of high confidence values in the mass function combining different pieces of evidence, a maximal mass distribution value is defined that contains the highest possible confidence that the EC can assign to an action after the initial assignment. A critical requirement for correlation is to also account for the possibility that false positives result in single alerts being held responsible for multiple actions that indicate non-legitimate associated actions. To deal with this conflict, each time an action is created, the EC checks whether any of the alerts involved in its creation have already been involved in another action to reduce the mass of the new action accordingly.

The combination of evidence using DST has to handle dependencies and conflicts between evidence. Thus, the need for a modification of the standard DST combination function arises. Therefore, Zhang's combination rule [72] is used to combine mass distributions of different statements as additive evidence. Zhang's center combination rule can handle mutually dependent evidence by first calculating the intersection of A and B, i.e., given evidences A and B with mass functions m_1 and m_2:

$$m_{1,2}\left(C\right)=k\cdot\sum_{A\cap B=C}\frac{|C|}{|A|\cdot|B|}\cdot m_1\left(A\right)\cdot m_2\left(B\right)$$

where k is a normalization factor that normalizes the sum to 1. For handling conflicting evidence, the logarithmic robust combination rule (RCR-L) [83] is used. Let evidence A and B have a conflict K, then functions α and β on K are given by

$$\alpha\left(K\right)=\frac{\log[(1+\lambda)^K\cdot\frac{(K+\lambda)^{1-K}}{\lambda}]}{\log[\frac{(1+\lambda)}{\lambda}]}$$

and

$$\beta\left(K\right)=\frac{\log[\frac{(1+\lambda)}{(K+\lambda)}]}{\log[\frac{(1+\lambda)}{\lambda}]}$$

where λ is a parameter in [0, 1] that is higher for lower cardinalities of θ. Given the evidence A and B with mass functions m_1 and m_2, and functions $\alpha(K)$ and $\beta(K)$, the RCR-L combined mass function is defined as

$$m_{1,2}\left(C\right)=\alpha\left(K\right)\cdot m_{\cup}\left(C\right)+\beta\left(K\right)\cdot m_{\cap}(C)$$

where

$$m_{\cup}\left(C\right)=\sum_{A\cup B=C}m_1\left(A\right)\cdot m_2\left(B\right)$$

and

$$m_{\cap}\left(C\right)=\sum_{A\cap B=C}m_1\left(A\right)\cdot m_2\left(B\right)$$

Within this framework, the two different combination functions are utilized as follows: Zhang's center combination rule is used for handling mass functions of interdependent evidence and RCR-L for handling mass functions of conflicting evidence. The significance of the combination rules can be illustrated by Zadeh's example (cf. Section 2.3.2), where $K\approx 1$ results in $\alpha(K)\approx 1$ and $\beta(K)\approx 0$ that leads to a probability of 0.01 for the case x_C where the maximum mass is 0.81 assigned to the focal element (x_A, x_B). The positive reinforcement of belief in the singleton x_C induced by RCR-L compared to the conjunctive rule such as the basic DST combination rule is insignificant in this case. Additionally, if multiple actions rely on the same alert, their current mass will each be combined with an "impact mass" m_{negA} that represents the negative impact that this has on their legitimacy (e.g., indication of legitimate but compromised RTUs). This impact mass has the properties $m_{negA}(\{t\})=0$, $m_{negA}(\{f\})=\mu$ and $m_{negA}(\{t,f\})=1-\mu$, with μ defined as a threshold belief that is higher than uncertainty. In doing so, the value of μ must be chosen to balance high confidence in the statement with allowing further combinations and processing without quickly becoming too overconfident. In particular, the selection of μ takes into account that $\mu\geq 0.5$ represents the belief in the statement, which is higher than the uncertainty. Therefore, certain values for μ in the range of $z\leq\mu\leq 0.5-z$ should be chosen to balance the high confidence in the statement with the possibility of further combination and processing without quickly becoming too uncertain. For example, assuming a high alarm rate in the target system and a possible false positive rate, selecting μ with $z=0.2$ results in a range of [0.2, 0.3] that provides enough flexibility to combine expected alarms. Thus, calibration runs in the target system can be automated with representative operation of the alarm output to select μ accordingly. If more than one alert referred to within the construction of the action had already been used, m_{negA} is applied again to the combination result as many times as there are such alerts. The EC finally outputs a list of attacker actions with associated confidence values, timestamps, and affected hosts for a preconfigured time horizon.

Table 4
Link types between nodes in attack graph.

Link type	Description
Same origin	The host that is involved in action *A* is also responsible for *B*.
Same target	Actions *A* and *B* both targeted the same host.
Extension	The host targeted by *A* continues to execute action *B*.
No overlap	No hosts involved in *A* was involved in *B* and vice versa.
Any overlap	The edge is valid independently of the hosts involved in either action.

3.4. Strategy correlator

After identifying the likely attack actions that have occurred by the EC, incorporating the potential list of infected hosts and the C2 host, the SC begins identifying potential attack strategies based on this that match the observations. The SC performs this analysis using predefined attack graphs that represent different attack strategies and include potential attack evolution processes as paths (cf. Fig. 3). The goal of our approach is to also shift the perspective of detection from the victim to the attacker's view, relaxing the need for detailed knowledge about the target and making our framework more flexible and modular by adding more attack graphs to the known set.

Conceptually, the attack graphs designed in this work have structural similarities with exploitation-based dependency graphs presented in [52]. The difference in graph structure is that the focus of our attack graphs is on general attack actions rather than vulnerabilities and their exploration. Thus, our attack graph design takes into account the attack development process by considering different types of attack steps depending on the situation and the action domain. In particular, in addition to meta-information about unique identifiers, the nodes in our attack graph also contain semantic links to the overall strategy, the attack action represented, and the corresponding kill chain phase of the step. The structural relationship between nodes and their predecessors and successors thus represents an attack decision in accordance with an overarching strategy, followed by the potential impact achieved on the target system or attacker state. Consequently, the edges in our attack graph represent the action transition considering the connections and participation of host.

For a directed edge from a node with action A to a node with action B, the possible kinds of host connection are defined in Table 4.

Tracing these connections makes it possible to understand the context of an attacker's actions without having to know in advance which hosts are present or connected in the victim's ICT network. There must also be a neutral starting point for paths through the attack graph, represented by an initial node that is added to all attack graphs and contains action and a kill-chain phase, both of which are called "unidentified" and serve as predecessors for all other nodes within the graph. In particular, each edge contains a mass distribution, which depends on the probability of the connected actions succeeding each other within the represented attack strategy. The mass distribution is based on a initial mass m_s with the properties $m_s(\{t\})=s$, $m_s(\{f\})=0$ and $m_s(\{t,f\})=1-s$, where $0\leq s\leq 1$ is a configurable weight to factor in preferable paths in the graph. In particular, the weighting factor s must be determined for the specific environment. To this end, configurable trial runs are used for calibration. Larger values

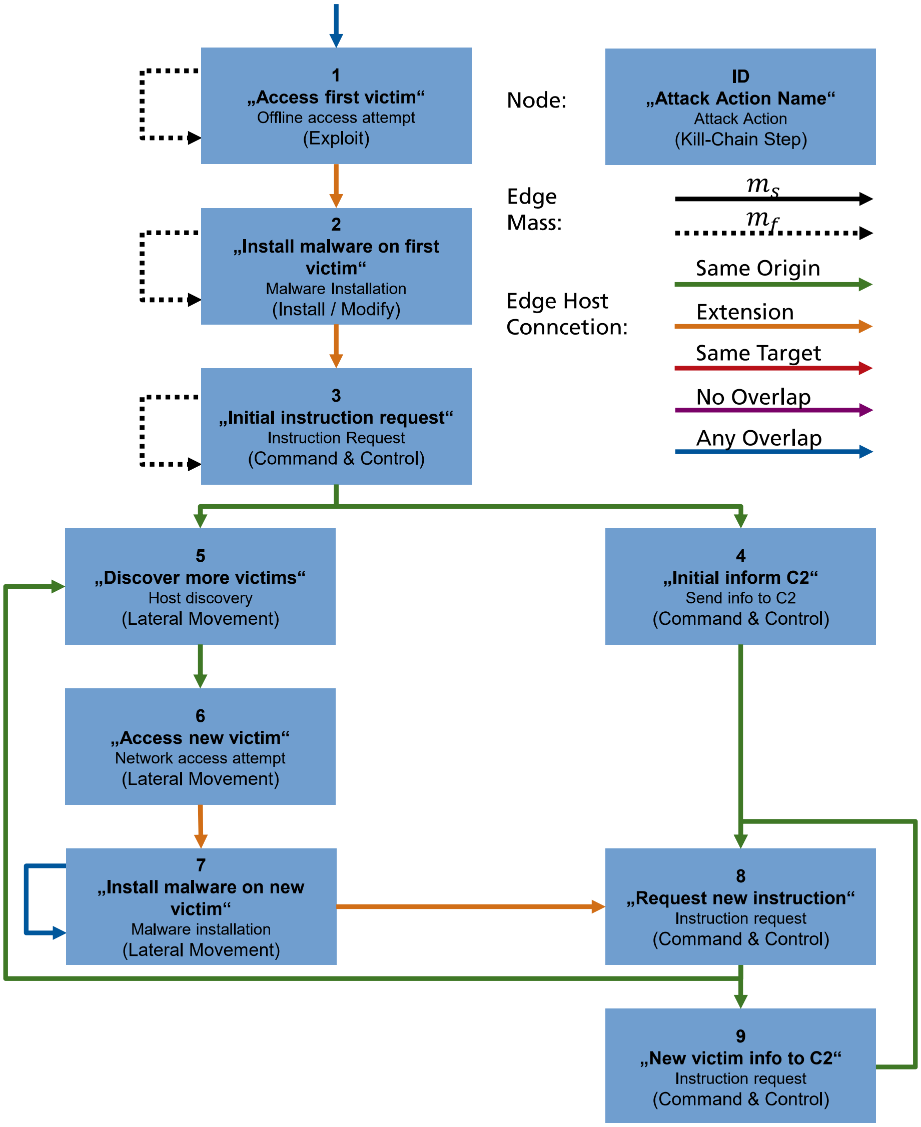

Fig. 3. Exemplary attack graph based on attacker actions for the Havex attack. Edges represent mass assignments and inference links between actions.

result in the SC giving too much confidence to the paths, while lower values result in the belief value values of the paths being too low to be properly evaluated. Overall, the structural design of the attack graph enables the representation of sequential attack processes in terms of paths containing nodes and edges of the graph. In essence, this consequently forms the attacker's attack strategy, which is represented by the attack graph. Thus, focusing the design of the attack graphs on SGs allows for domain-specific consideration of the attack campaign. Therefore, an essential part of the SC is the stored attack graphs that form the knowledge base of the correlator, based on which the analysis is performed to identify strategies. First, the predefined attack graph is extended to dynamically account for the current situation by adding new edges that provide the ability to account for undetected actions or inconsistencies within the observations. Particularly, the mass distribution depends on the shortest path length existing between the nodes in question and the masses of the edges along this shortest path.

Based on the initial preparation, the reconstruction of the potential attack path is performed in form of paths within the attack graphs, starting with a chronological listing of the nodes traversed. In addition to the listing, the mass functions are assigned to the paths considering the mass functions of the traversed nodes and the involved edges. In this case, the mass functions of the attack action are set to relatively high uncertainty. Given the additive nature of Zhang's central combination rule, this is a necessary step to avoid premature concentration of high certainty with few observations. Therefore, the mass distributions of all detected actions must be adjusted to contain a large fraction of the uncertainty before they can be used for path construction. Specifically, for each action mass m_{action}, the adjusted action mass m_a is calculated with the following properties:

$$m_a(\{t\}) = m_{action}(\{t\}) * s$$

$$m_a(\{f\}) = m_{action}(\{f\}) * s$$

$$m_a(\{t, f\}) = 1 - s * |m_{action}(\{t\}) - m_{action}(\{f\})|$$

Thus, the confidence in each action decreases by the same factor, while the ratio of belief in truth to belief in untruth is preserved. For each known attack graph, the path construction is initialized with the empty path and all detected actions are considered in chronological order and possible paths are constructed iteratively. To prioritize and strategically decide which paths should be further explored, it is determined which paths should be discarded if extending a path would cause its negative belief value to exceed a certain threshold or old paths that have already been updated with node extension.

As part of the analysis, mass functions are assigned to the attack graphs to represent the overall confidence level in detecting the correct attack strategy. Here, the mass functions are part of a "agreement" between the correctly recognized attack entities attack action, path, and graph, with the intersection of attack action within the graph and consequently contained path affecting the mass function value. Based on the mass function distribution, various checks are performed to determine whether the attack path with the highest confidence value is included in the attack graph with the highest confidence value and whether both exceed their respective confidence thresholds. To find the optimal pair of path and graph, or if there is a disagreement, it is decided whether to prioritize the belief value of the graph or the path to still output a plausible result, otherwise a sufficiently credible pair could not be found. To perform this optimal result matching, the following procedure is used: All found potentially traversable paths are sorted by the computed belief values to find the path p_b with the highest belief value $Bel(\{t\})$, according to the mass assigned to it. The same procedure is performed for the known attack graphs, where the graph with the highest belief value is denoted by g_b. To be considered as a result of interest, the discovered attack paths and the corresponding attack graphs need to fulfill certain confidence levels. Therefore, thresholds for plausibility and belief are introduced. To be considered plausible, the mass distribution m_p of a given attack path p must satisfy the following conditions: $Bel(\{t\}) \geq bel_p$ and $Pl(\{t\}) \geq pl_p$. Accordingly, the mass distribution m_g of a given graph g must satisfy the conditions: $Bel(\{t\}) \geq bel_g$ and $Pl(\{t\}) \geq pl_g$. For instance, the thresholds can be configured as follows to support optimal selection of attack paths and graph pairs: $bel_g \geq 0.5$, $bel_p = \frac{1}{2} \cdot bel_g$, $pl_g = pl_p$, (e.g., $bel_p = 0.3$, $pl_p = 0.9$, $bel_g = 0.6$, and $pl_g = 0.9$). After determining the most credible attack path p_b and graph g_b, it has to be checked whether p_b passes through g_b or not. If so, they are the most credible available pair of attack paths and associated graphs. If both g_b and p_b exceed the corresponding thresholds, they are considered the most credible feasible solution. If p_b exists and g_b fails only at its belief value boundary, the pair is still accepted provided $Bel(\{t\})$ is the same for both mass distributions. This is because these circumstances indicate a path that is credible enough and passes through the most credible graph. It also means that g_b does not exceed its required belief value only because of insufficient data, since there is no evidence to the contrary. Otherwise, no sufficiently credible pair of attack graph and path could be found. If the most credible path is through an attack graph $g_x \neq g_b$, a decision must be made whether to use p_b and the (less credible) graph g_x or g_b and its own most credible path p_x (which is less credible than p_b). In order for this decision to be made, it is first checked that both pairs (p_b, g_x) and (p_x, g_b) satisfy all the limits described earlier. If one pair fails, the other is taken as the optimal result. If both fail, no suitable pair could be found, and the process described begins without an optimal path and graph. If both pairs (p_b, g_x) and (p_x, g_b) pass all cutoffs, the following tests take place:

$$m_{g_b}(\{f\}) > m_{g_x}(\{f\})$$

$$m_{g_b}(\{t\}) - m_{g_x}(\{t\}) < g_{diff}$$

$$m_{p_b}(\{t\}) - m_{p_x}(\{t\}) > p_{diff}$$

where g_{diff} and p_{diff} mark the maximum reduction in belief using g_x instead of g_b and the minimum gain in belief using p_b instead of p_x, respectively (e.g., possible configuration are $g_{diff} = 0.05$ and $p_{diff} = 0.2$). These tests examine whether the pair (p_b, g_x) (with mass distributions m_{p_b} and m_{g_x}) is sufficiently more credible and plausible than (g_b, p_x) with mass distributions m_{g_b} and m_{p_x}. If at least one of these tests is passed, the pair (p_b, g_x) is accepted as the optimal feasible pair of attack graph and path. If not, (g_b, p_x) is used instead. At this stage, SC completes its analysis and outputs a collection of pairs containing reconstructed attack paths and associated graphs with the corresponding belief values for the next component to be considered.

3.5. Kill-chain identification

At this stage, the kill-chain identification component is responsible for determining the confidence level of the entire attack and, accordingly, based on the previous correlation results, identifying the optimal result pair that represents the attack campaign. Thus, the analysis is primarily based on finding an optimal pair of attack graphs and paths that are characterized as the most credible and plausible pair of outcomes according to the mass function values. In particular, the values of the mass functions are successively compared with predefined thresholds and limits that represent the minimum confidence level to be considered a result of interest. Consequently, the corresponding plausibility and belief values of the attack graph and the path pair constellation are used for the cutoff process in these comparison tasks.

Using the selected set of pairwise results, it is checked whether the most credible path is contained in the most credible graph that would form the optimal solution. Accordingly, the attack strategy determined in this process is defined by the attack graph contained in the optimal solution. Thus, the kill-chain phase of the attack is determined by the phase implications contained in the path of the optimal solution. The last kill chain phase of the attack node included in the path represents the current phase the attacker is in.

If no matching pair is found for the optimal solution, the output of the correlation process depicts either no attack or an attack whose strategy could not be identified. These two scenarios are distinguished based on the detected attack and the detected infected hosts, conditional on the cases when either no hosts were detected as infected or there was not a single access attempt to an ICT network host detected by the EC, the system concludes that no attack occurred, otherwise the result is an "unidentified attack". The final output of the correlation process consists of the detected attack actions and strategies, the last perceived kill-chain phase of the attacker, and the list of hosts detected as infected. Consequently, DOMCA provides results that identifies whether an attack occurred within a certain time horizon (defined by the available input and the assumed relevant correlation of the indicators), the development process of the attack (determined by the attack path), and the strategy followed (defined by the attack graph). In addition, the corresponding phase (detected kill chain phases) and the host involved in the attack (list of infected hosts by attack actions) are determined.

3.6. Domain-specific attribution

In the above sections, the DOMCA framework and methodology are presented and described, which provides transferability capabilities to other domains due to the modular correlation approach based on the predefined attack actions and graphs. In particular, the domain-specific mapping for SG is given by the attack indications that use the SG process network specification such as SCADA systems, and the attack graphs that include domain-specific attack targets. For example, lower level IDS sensors may provide DOMCA attack indications that represent anomalous or suspicious behavior that violates legitimate SG operational specifications and processes (e.g., incorrect addressing in industry protocol packets, unauthorized communication channels or routes, unauthorized operations or data points, anomalous data flow, violation of technical plant specifications, inconsistent measurements). These indications are obtained by incorporating domain-specific knowledge, such as known legitimate routes and the topological structure of the network, roles of components and allowed operations, communication structure, and channels, process domain, and plausibility of measurements and control measures. Combined with event and strategy correlation, these domain-specific attack indications are mapped to the appropriate attack levels in the attack graphs. Since the normalized attacks are mapped to the MITRE ATT&CK attack actions, DOMCA can be easily applied to other critical infrastructures, with the extension limited only to designing appropriate attack graphs that reflect the attack strategy for the specific target systems. The resulting domain-specific attack graphs, therefore, include objectives and actions that are specific to the target system and can only be achieved through observations or attack indications generated exclusively in that environment (e.g., causing a power imbalance by sending network-damaging setpoint commands to DER assets). In the context of multi-stage cyber attacks, these domain-specific attack indications, particularly alarms generated in the process and field environment, are often generated in the later stages of attack development, when it is often too late for potential countermeasures. Therefore, it is essential to take a holistic view of the entire attack evolution to provide early detection of the cyber attack campaign, starting with network penetration, lateral movement within the target system, the establishment of the corresponding C2 overlay network, and tracking of the set targets.

3.7. Post-processing

The result of DOMCA can be post-processed by an upstreaming component that provides visualization functions of the output attack campaign with higher-level processing functions to derive further information (e.g., prediction of next steps). When applied to security and situational awareness, streamlining and visualization can provide easier access to the current situational awareness by highlighting relevant information from the analysis. In addition, cyber incidents can be condensed into a format that is understandable to the user and presented with the appropriate level of confidence, for example, in security centers to respond to potential incidents. Thus, the result could play an important role in decision-making and deriving appropriate countermeasures to the currently identified incidents. Specifically, based on the results of the correlation process, a decision support system can be built to assist the response team during cyber incidents and emergencies by providing action instructions for appropriate remediation. In particular, by using prediction methods based on the current situation of the attack.

4. Results & evaluation

In this section, we present our results and evaluation of the investigation conducted, which follows in detail the procedure described in Section 4.1, with evaluation and discussion of the results in Section 4.2. The investigation is based on a simulation environment of SG that simulates multi-stage cyber attacks based on real precedence cases (cf. Section 4.2). In particular, we analyze the accuracy of the detected incidents and the corresponding strategies and evaluate the influence under different parameters that change the behavior of the ICT network or the attacker (cf. Section 4.4).

4.1. Procedure for the investigation

Due to the lack of attack data for multi-stage cyber attacks, which are described in more detail in Section 4.5, we chose to simulate attacks based on previous work in conjunction with the capabilities of simulating multi-stage cyber attacks for our study. Following a graph-based modeling approach for smart grid architectures [18], we simulate multi-stage cyber attacks with different strategies, some of which are inspired by real cyber incidents such as Havex, and Stuxnet. Moreover, in addition to the control scenario in which no attack takes place, we also simulate randomized attackers who randomly perform some cyber attacks without any specific strategy or pattern. Also, the parameters used to modify the network simulation and attacker behavior can be varied to provide more complete insight into the performance of DOMCA under different circumstances. Within the simulation, we perform various parameter variations for the network setting, such as the distribution of the existing vulnerabilities, in which the host configuration with respect to the vulnerability that explicitly specifies the chance of success of a remote compromise attempt by the attacker is varied. In addition, the lower-level sensors are also part of the scenario design, where the coverage is also varied. This is expected to have a significant impact on the detection quality of DOMCA, especially on the level of observation available for correlation. In particular, we vary the position of the sensor near the C2 host. Also, the duration of the entire simulation can be configured, measured in discrete integer steps, with longer simulation runs leading to more evolving attack scenarios and more diverse data generation. We performed a total of 207 simulation runs as part of the investigation, including 51 runs with no attack, 52 with an attacker based on Havex, 50 with an attacker based on Stuxnet, and 54 runs with a randomly acting attacker.

4.2. Attack modeling and implementation

Based on the MITRE ATT&CK matrix [64], especially for industrial control systems [81], and historic cyber security incidents in critical infrastructures [84], such as Stuxnet, Havex [85] and Industroyer [86], we model the multi-stage cyber attack scenarios. For each known software involved in cyber attacks, from the knowledge base formed by the aforementioned sources, the used techniques, their specific instances and the tactics they belong to can be extracted. To harmonize this knowledge base with the kill-chain concept, which in its essence attempts to structure coordinated attacks and divide them into chronological sections to depict common patterns, we assign the techniques from ATT&CK to the stages of the kill-chain. It combines the structure of the kill-chain with the specific attack contents provided by the MITRE ATT&CK matrix, enabling a more complete view of the attack software. Fig. 4 illustrates an example of modeling the attacker using the Havex example. Hereby, the adversary tactics, techniques, and procedures of the MITRE ATT&CK matrix are mapped to the attacker's kill-chain phases to form our multi-stage attack model.

After structure and content of known attacks have been combined to form an integrated representation, these attack strategies are abstracted further and implemented to the simulation environment. For the investigation, we implemented three attack strategies: One based on the Havex software, one based on the Stuxnet software, and one performing random malicious actions without following any known pattern. The execution of cyber attack steps is therefore integrated into the scheduling process of the simulation environment. Each time a simulation step is executed, the cyber attacker proceeds with its sequence of actions. With the cyber attacker running synchronously with the environment simulation, it is possible to communicate between

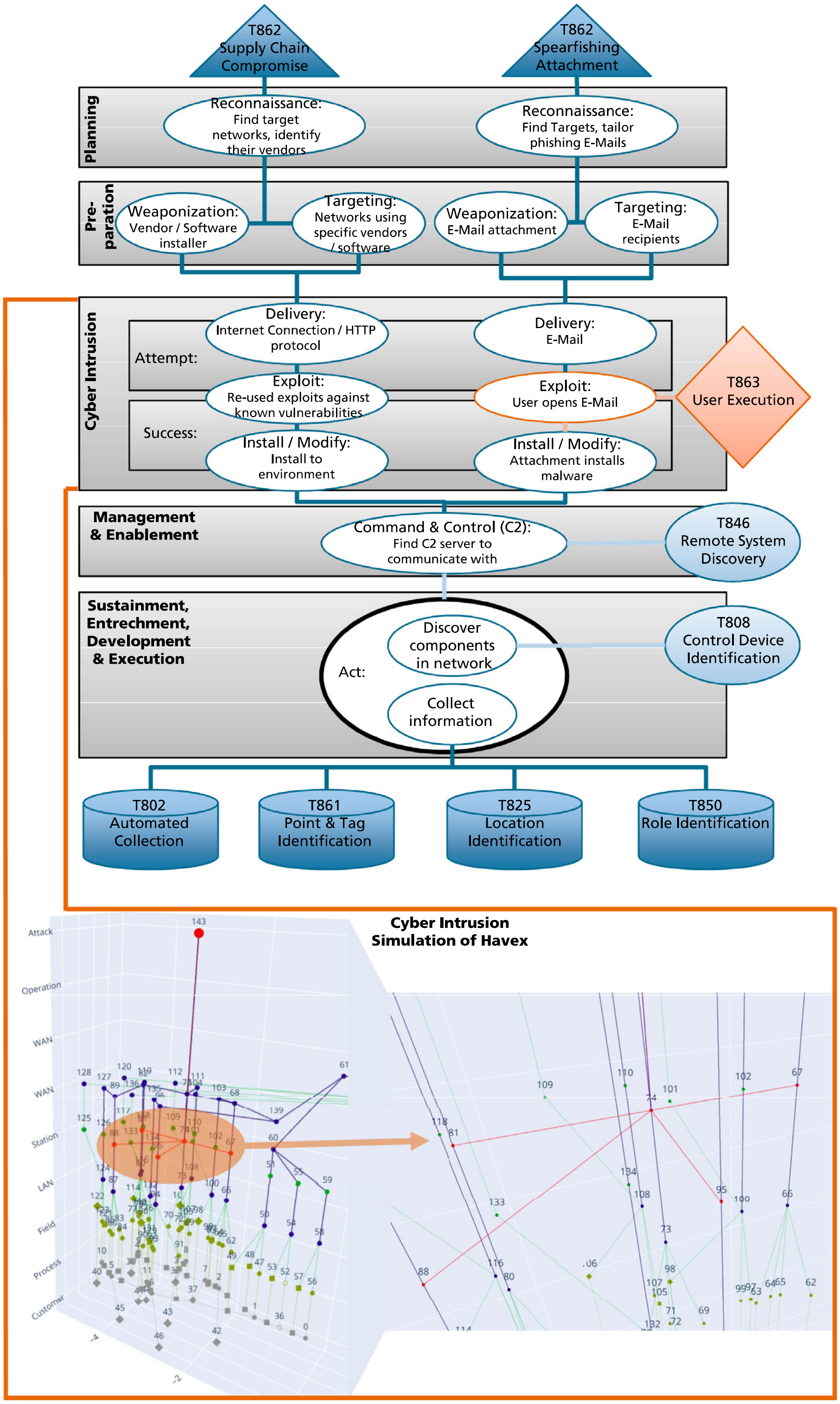

Fig. 4. Illustration of the attacker modeling approach using the Havex example. The upper right corner shows the propagation of the modeled attack in the simulation environment.

infected hosts and send packets within the general communication scheme of the simulation. The goal if the attacker is to use client–server communication to send messages and commands from a C2 server to infected hosts within the target's ICT network. Since not all nodes are connected to all other nodes of the communication network, messages between a host and the C2 server are relayed through other infected intermediate hosts.

The C2 server in our implementation has a complete control over the infected hosts and is able to decide what the next action of each one should be. The attacker's C2 server is responsible for gathering information about the environment and distributing instructions to the malware running on victim hosts. In the simulation, we initialize the C2 server that is able to communicate via the target's ICT network of the simulated system.

4.3. Evaluation criteria

The evaluation of the proposed DOMCA's performance is based on criteria that provide insight into the system's capabilities by measuring its ability to correctly identify the infected hosts within the ICT network or the attacker's executed actions. Besides, the criteria determine whether the system can correctly identify a known attack strategy and the phase of the kill-chain that is currently being executed within it. Thus, there are four main criteria for evaluating the performance of our attack detection approach:

(1) detection of compromised hosts in the network
(2) confidence level of detected attacker actions
(3) confidence level of detected attacker strategy
(4) identification of the current phase in the kill-chain (attacker's progress)

The first two criteria are intermediate results, which are used to evaluate DOMCA's detection capabilities. They examine the system's ability to correctly identify the infected hosts within the ICT network, and the attacker's executed actions respectively. The second and third criteria are used for determining whether DOMCA is capable of correctly identifying a known attack strategy and the phase of the kill-chain currently being executed within it.

Performing network analysis, DOMCA detects which hosts within the simulated ICT network are likely compromised by an attacker. The resulting list is compared to the attacker's own list of actually infected hosts, and the overlap stored in percent.

DOMCA also determines the actions taken by an attacker in chronological order. Each action is assigned a mass, storing plausibility and belief in it. The list of actions with a plausibility larger than the predefined threshold value is compared to the attacker's own list of executed actions, and the overlap stored in percent.

One of the relevant goals of DOMCA is to correctly identify an attack following a known strategy. DOMCA has a pool of known attack graphs that it can identify. Two of these attack graphs (the Havex-based and the Stuxnet-based attacker) are executed attacks that occur during the simulation. Two more attack graphs (communication-dependent and removable media attackers) are available to DOMCA to detect, but are not implemented and therefore will not occur within the simulation. Additionally, the randomized attacker is implemented within the environment and does not follow any known strategy. In this case, the system should detect that there is an attacker present, and that it does not conform to any predefined attack graph. The output "unidentified attack" is considered correct when a randomized attacker was present. Also, there exists the chance that the randomized attacker will replicate a known attack strategy. If DOMCA then attributes the attack to that strategy, its detection is considered incorrect. Finally, it is possible that there is no cyber attacker present in the ICT network at all. Therefore, an output of "no attack" is considered correct in this case.

Lastly, in our investigation, we also evaluate the identification of the kill-chain phase of an attacker. kill-chain phase identification is therefore evaluated as correctly, if the mapped kill-chain phases correspond to the actual executed phases of the attacker within the simulation.

4.4. Evaluation of the accuracy of classification

The results of our investigation are presented in Fig. 5, which depicts the detection quality of DOMCA in different attack scenarios that differ in terms of detected attack, strategy, and kill chain step. To investigate the influence of sensors, we also considered the effect of sensor placement on detection quality within the plot in terms of the average influence rate. Key observation in the research is that no false alarms were detected during the simulated attack scenarios, indicating in particular that, with high accuracy, no alerts are issued when no attack is present. The detection rate of the existing attack scenarios was also remarkably high, with a detection rate of 87.86%.

Furthermore, in Fig. 6 we plotted the detection rate in dependence of the simulation duration of each simulated scenario. The detection rate over all scenarios plotted on the y-axis represents the aggregated detection rates over each simulation sorted by the duration of the simulation runs. Thus, the plot depicts the evolution of the detection rate across the scenarios with increasing duration of the simulations to determine the impact of duration on the detection rate. As the figure shows, there is a trend towards an increasing detection rate across all scenarios, both for the correctly detected strategy and for the correct kill chain phases, based on an extended simulation time. Since a longer duration of attack scenarios also means potentially more observations that can be received via the deployed sensors, DOMCA has more information to detect the attack. Moreover, Fig. 7 indicates, that we can observe the same effect for each scenario depending on the sensor placement and independent of the chosen attack strategy. In addition to observing the same effect of increasing duration on detection rate, we can also observe the influence of sensor placement independent of duration. It is clear that the placement of the sensors relative to the C2 server also has an impact on the detection rate. The experiments show that a sensor placed near the C2 server improves the detection rate of DOMCA. In addition, in Fig. 8, the step size at which the attack model executes its actions is varied to also examine the effects of step size on the detection rate. For better comparison, the detection rate of the experiment is normalized to compare the variation in detection rate for each step size varied for an attack-induced and a non-attack-induced scenario. The results indicate that no significant effect of step size on the detection rate is observed in the attack-induced and non-attack-induced scenarios, as well as in the aggregated case. This result shows that DOMCA does not depend on the anomaly rate of attack signatures compared to normal traffic. Moreover, it implies that DOMCA can reliably distinguish between attacks and non-attacks even in imbalanced datasets.

Including data from all 207 simulation runs, the distribution of the actual attack strategies with respect to the attack pattern determined by the system can be obtained from Fig. 5). For example, when DOMCA issued the "No Attack" result, it was correct in 72.86% of all cases evaluated. When DOMCA yielded the result "Havex attack", this was correct in 50.86% of the cases. Moreover, our results show that correct distinction between normal and attack-induced situation is reliably detected, even if an attack is mostly identified as "unidentified" attack situation when no matching attack graph could be found. This means that the system detected that an attacker was present, but could not assign a known strategy with sufficient confidence.

In a "randomized" attack, where random attack actions are performed, the attack is still detected in the vast majority of cases. However, it is often incorrectly assigned to a known attack graph. Still, the most common conclusion the system draws is an "unidentified" attack is correct. When varying sensor placement, we observed that the detection accuracy of each known attack graph was up to 36 percentage points higher when such a sensor was placed near a C2 host but showed little impact on the correct detection of an unknown/random attack. With a sensor, detection becomes more accurate because more data is available and the C2 node in our scenarios plays a central role in many of the attacker's activities and thus has significantly more influence on detection quality due to sensor placement. The results regarding the correct determination of the last kill-chain phase of an attacker show a

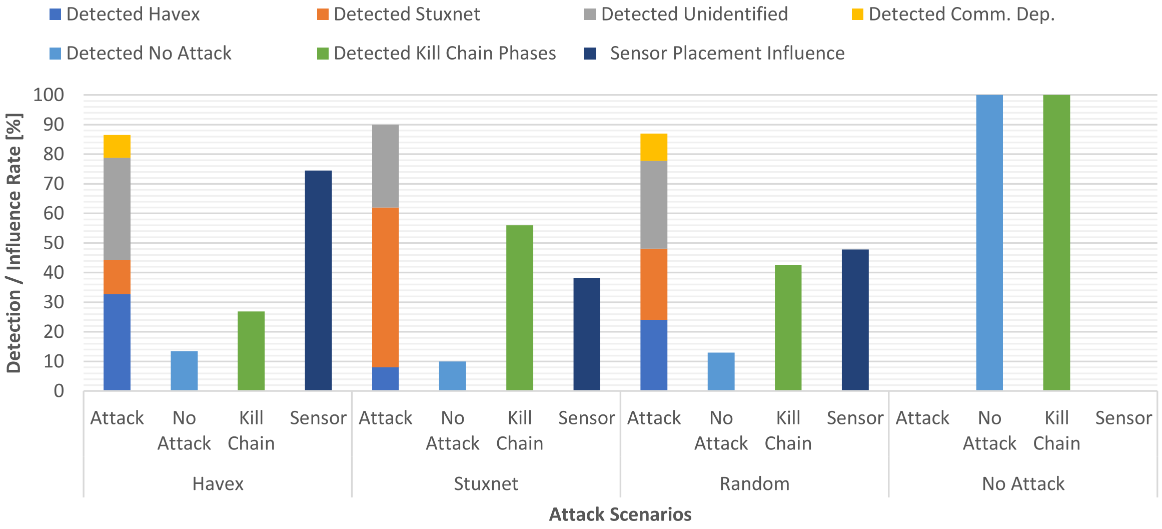

Fig. 5. The classification accuracy assessment chart shows the attack scenarios performed on the *x*-axis, including the "no attack" event, and on the *y*-axis the distribution of the detection rate of attack strategies, kill-chain phases, and the influence of sensor placement on detection quality.

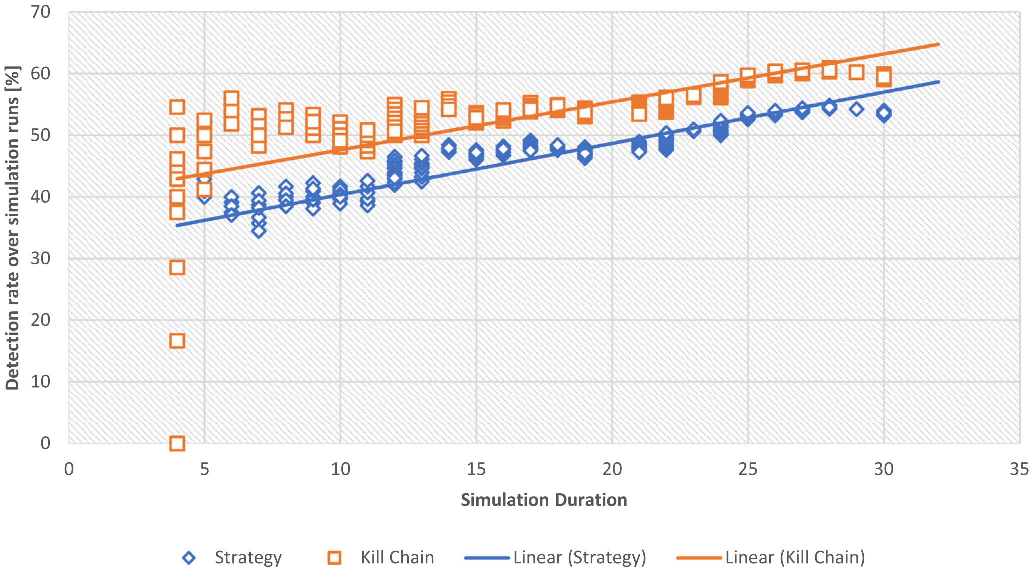

Fig. 6. The classification accuracy assessment chart shows the simulation duration of each simulated scenarios on the *x*-axis, and on the *y*-axis the distribution of the detection rate of attack strategies, and kill-chain phases over all simulated scenarios.

detection accuracy of 56.38% for different kill-chain phases for all attack strategies.

The reliability of kill-chain identification is highly dependent on the correct identification of the attack strategy. When a known attack graph was correctly identified, the kill-chain phase was also correctly determined in over 97% of the cases. If the attack strategy is not correctly detected, the corresponding kill-chain detection is less reliable and is correctly detected at a rate of 14%. We also observed that the detection rate of the different attack strategies depends on the simulation duration, with the false-negative rate decreasing as the duration of the simulated scenario increases. This is because DOMCA has more data that allows a more reliable identification of an attacker and the corresponding attack strategy.

4.5. Qualitative comparison with other systems

In this section, we perform a comparison of DOMCA with other similar systems to highlight the different features of the detection process and quality. However, the comparison is made on a qualitative basis for several reasons. A key component of our performance analysis in Section 4.4 is the simulated attack data described in Section 4.2, which represents a multi-stage cyber attack in SG. Due to the lack and deficiency of data on multi-stage cyber attacks, in which not only attack vectors but logically bound sequences of multiple attack vectors are mapped to form kill-chain related attack sequences, a basis for scientific comparison of different detection methods is not yet available to our knowledge. To address this problem, we have developed a simulation environment capable of simulating multi-stage cyber attacks that meet the requirement of forming a complex chain of attack actions with well-defined targets following an overarching strategy. Apart from the critical situation of missing benchmark datasets for multi-stage cyber attacks, the procedure for simulating or synthetically generating the missing datasets is also different in the different approaches and does not follow a standardized process. This also presents a challenge for comparing different approaches using synthetic datasets that meet

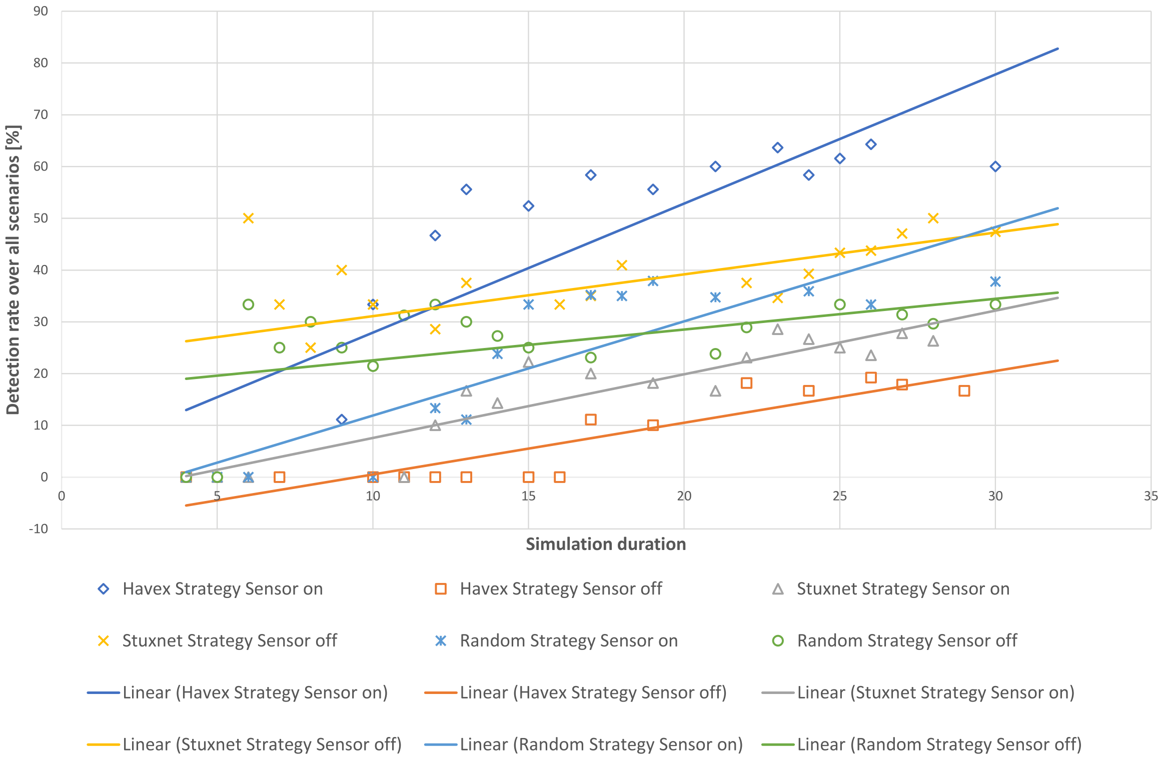

Fig. 7. Granular representation of the trend curve of each simulated scenario as a function of the detection rate across the simulated scenarios. In addition, the detection rate curve is differentiated by the type of sensor placement: "Sensor on" represents sensor placement near the C2 server and "Sensor off" represents the opposite case.

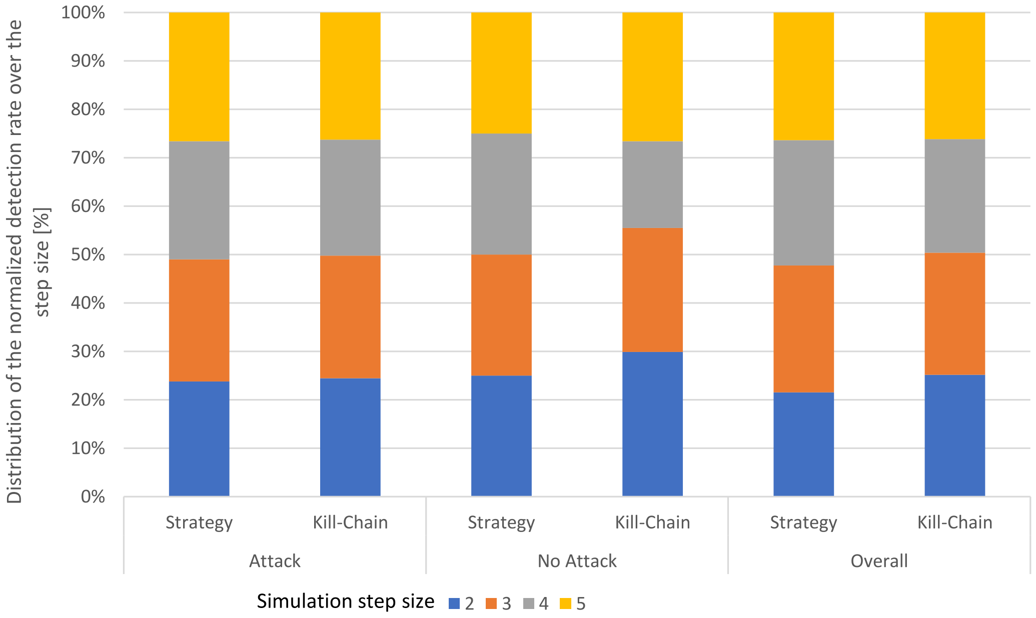

Fig. 8. Illustration of the distribution of the detection rate at different step sizes of the simulation runs. The *x*-axis represents attack-induced and no attack scenarios, which are additionally aggregated to an overall case.

a wide range of criteria for different approaches. With the focus on SG, the target system as the basis for attack simulation may also diverge, e.g., our simulation environment is tailored to SG in the European region, which differs from U.S. grids already by the communication protocols used (e.g., IEC 60870 vs. DNP3). Moreover, the results or expected outcomes differ between the different approaches, with our approach DOMCA aiming not only to detect attacks, but also to determine the corresponding attack actions, strategy, and development process based on the kill chain concept. Consequently, for a systematic and standardized comparison of different approaches, we see the current challenge of providing an existing benchmark dataset that includes the required attack data for different detection approaches in a standardized form that meets the requirements of multi-stage characteristics and common cyber–physical system environment. Nevertheless, at least on the basis of the following characteristics, we try to make a comparison with other similar systems and present the result in the Table 5:

Table 5
Qualitative comparison between DOMCA and other systems.

Reference	Method	Dataset	Input types	A_a	A_p	A_s
[87]	Attribute matching	Case study	IDS data	–	X	–
[88]	Attribute correlation	DARPA 2000	IDS alerts	X	–	X
[89]	Scenario clustering	Private	IDS alerts	–	X	X
[90]	Statistical analysis	Private	Logging data	X	–	–
[91]	Pre-/Post-condition analysis	DARPA 2000	IDS alerts	–	X	–
[92]	Model matching analysis	DARPA 2000	IDS alerts	–	X	–
[93]	Statistical inference analysis	DARPA 2000	IDS alerts	–	X	–
[94]	Structural-based analysis	Case study	IDS alerts	–	–	X
[95]	Mixed	DARPA 2000	IDS data	–	X	X
DOMCA	Mixed	Simulated	IDS, IT & OT data	X	X	X

- Method: the method used for the correlation approach
- Dataset: the dataset used for the evaluation that is presented in the reference
- Input Types: type of expected input data for the correlation approach
- Attack action (A_a): capabilities of the proposed approach to detect involved attack actions in the attack scenario
- Attack path (A_p): capabilities of the proposed approach to detect the attack development process in the attack scenario
- Attack strategy (A_s): capabilities of the proposed approach to detect the attack strategy in the attack scenario

In the qualitative comparison, it is noticeable that DOMCA differs from the other approaches primarily in the degree of output. While other systems also aim to detect multi-stage cyber attacks, the output of their correlation includes the sequential attack process or attack graph, but not combined the strategy involved in the attack process and development. In addition, the approaches often do not distinguish the individual phases within the attack process according to established structural concepts such as the kill chain. This is also observable for the attack actions involved in the attack campaign, which do not refer to established definitions such as those of MITRE ATT&CK. The types of input in the correlation approach also differ with regards to the different source types such as IDS or Logging data. Many approaches rely on traditional IDS alerts processed in the correlation approach to detect the attack within the communication layer. Depending on the method of the approaches, different data sources become more interesting than others, and the level of information in the results can potentially suffer from heterogeneous data. In addition to the divergent data basis for evaluation, it is clear that a missing standardized understanding of the results with regards to established concepts and definitions makes the comparison with DOMCA difficult.

5. Discussion

In Section 4, we analyzed and evaluated our proposed approach DOMCA against different attack scenarios in which DOMCA could reliably detect the presence of the attack, but also the corresponding strategy as well as the kill-chain phases. Since our approach uses dynamic security-related information such as logs and alarms from IDS sensors, we also observed a strong dependency of the detection rate on sensor placement in our experiments. It has been observed that placing sensors near the C2 server increases the detection rate of DOMCA by providing additional information about the C2 communication that characterizes most of our attack scenarios. Moreover, our approach is designed to enable situational awareness through contextual correlation of available information such as alarms from IDS. Thus, it is within expectations that as the information base decreases, the detection rate also decreases. Subsequently, sensor placement as well as the duration of attack scenarios contribute significantly to the accuracy of detection of attack strategies and kill-chain phases. In particular, the presence of a sensor monitoring communication with an C2 host contributes significantly to the accuracy of detecting the strategy used by an attacker in our experiments. However, our approach does not represent a learning-based system where a predefined dataset is required to train the model for detection using, e.g., anomaly-based or signature-based detection. Rather, our approach uses heterogeneous information via stochastic inference and model-based correlation for its detection mechanism. The impact of decreasing information on detection quality is limited by increasing uncertainty within security-related information such as alerts. Since we handle the uncertainty with the DST, the detection rate does not decrease as much due to insufficient information because the observation is robustly handled by the adjusted combination rules. However, less information still leads to a higher range of uncertainty within our evidence, potentially leading to lower belief values and thus lower confidence in the detection of the strategy. This can lead to an attack being detected in the early stages but not correctly mapped to a known attack graph. Here, we observed that the effect of a more universal kill-chain phase, independent of a particular strategy, benefited DOMCA in the initial early detection. However, misrecognized strategies result in a lower rate of detected kill-chain phases because the context of the attack is missing from the attack graph. Furthermore, the qualitative comparison between DOMCA and the other systems in Section 4.5 also illustrates the result quality of DOMCA. Compared to the other approaches, DOMCA is able to identify the involved attack actions, attack evolution and strategy, which are also linked to well-established concepts and definitions such as the kill chain concept and the MITRE ATT&CK matrix. The ability to incorporate more diverse data, such as IT and OT-related data, also allows richer output information to be obtained compared to systems that rely solely on traditional communications data. Overall, the results show that DOMCA can be used for reliable detection of multi-stage cyber attacks in SGs. Depending on the observable network area and prior knowledge of attack actions and strategies, reliable detection of the attacker's strategy and current position in the kill-chain are also possible and provide an advanced basis for attack prediction and defense.

6. Conclusion

In light of emerging challenges in mitigating cyber attacks in the IT and OT landscape, reactive mitigation and countermeasures require accurate and situationally descriptive attack campaign detection capabilities. In particular, this requires an understanding of the attack development process not only in terms of communication-dependent processes. This form of situational awareness requires further insight into the evolution of the attack, the quality of the information gathered, the impact of the attack on critical assets, the behavior of the attacker during an incident, and possible future developments. Subsequently, an advanced information base can be provided for the preparation, selection, and execution of appropriate mitigation, countermeasure, and recovery plans.

To this end, in this paper, we introduce the DOMCA correlation approach, which is designed to identify the cyber security posture within SG in the context of multi-stage cyber attacks. In this paper, we provide insights into the design of DOMCA, specifically the core event and strategy correlator component, followed by the kill-chain identification component. In addition, we also investigate the detection quality of DOMCA under different cyber threat scenarios. This allows us to investigate the validity of the results associated with the simulation data used and to identify challenges related to the partitioning of an attacker's operations through the kill-chain.

During our research, we were able to show that DOMCA can reliably detect multi-stage cyber attacks that follow specific strategies in terms of their actions, evolution, and phases. In alignment with our expectation, we were also able to confirm the dependence on sensor placement and its influence on the detection quality of DOMCA, especially the kill-chain phase and the identification of the attack strategy. Due to the modular nature of attack actions and graphs, DOMCA can be extended beyond the demonstrated use case in SGs by incorporating domain-specific attack actions and strategies. In future work, DOMCA can be tested with a wider range of known attack actions and graphs, as well as with a new attack model to structure cyber attacks from the target's perspective and to distribute an attacker's operations more evenly across the target's ICT network. Overall, our research indicates that DOMCA can reliably reconstruct complex attack campaigns, with reliable detection of the evolution and strategy, provided that a sufficient information base is available. Thus, the result of DOMCA provides an advanced foundation for further research towards decision support system and automated response to cyber attacks.

CRediT authorship contribution statement

Ömer Sen: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Dennis van der Velde:** Writing – review & editing, Project administration. **Katharina A. Wehrmeister:** Data curation, Formal analysis, Investigation, Methodology, Resources, Validation, Writing – review & editing. **Immanuel Hacker:** Writing – review & editing, Visualization. **Martin Henze:** Investigation, Methodology, Supervision, Validation, Writing – review & editing. **Michael Andres:** Supervision, Writing – review & editing, Funding acquisition, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has partly been funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under project funding reference 0350028. The responsibility for the content of this publication lies with the authors.

References

[1] M.L. Tuballa, M.L. Abundo, A review of the development of smart grid technologies, Renew. Sustain. Energy Rev. (2016).

[2] M. Uslar, Energy informatics: Definition, state-of-the-art and new horizons, ComForEn (2015).

[3] S. Barsali, et al., Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources, 2014.

[4] M. Kuzlu, M. Pipattanasomporn, Assessment of communication technologies and network requirements for different smart grid applications, in: IEEE PES ISGT, IEEE, 2013.

[5] B.M. Buchholz, Z. Styczynski, Smart Grids: Fundamentals and Technologies in Electric Power Systems of the Future, Springer, 2020.

[6] M. Serror, S. Hack, M. Henze, M. Schuba, K. Wehrle, Challenges and opportunities in securing the industrial internet of things, IEEE Trans. Ind. Inform. (2021).

[7] T. Krause, R. Ernst, B. Klaer, I. Hacker, M. Henze, Cybersecurity in power grids: Challenges and opportunities, 2021, arXiv:2105.00013 [cs.CR].

[8] D. van der Velde, M. Henze, P. Kathmann, E. Wassermann, M. Andres, D. Bracht, R. Ernst, G. Hallak, B. Klaer, P. Linnartz, et al., Methods for actors in the electric power system to prevent, detect and react to ICT attacks and failures, in: IEEE ENERGYCon, 2020.

[9] M. Henze, L. Bader, J. Filter, O. Lamberts, S. Ofner, D. van der Velde, Poster: Cybersecurity research and training for power distribution grids – A blueprint, in: ACM CCS, 2020.

[10] K. Kimani, V. Oduol, K. Langat, Cyber security challenges for IoT-based smart grid networks, IJCIP (2019).

[11] J. Mendel, et al., Smart grid cyber security challenges: Overview and classification, E-Mentor (2017).

[12] K. Wolsing, E. Wagner, M. Henze, Poster: Facilitating protocol-independent industrial intrusion detection systems, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2020.

[13] J.J. Chromik, Process-aware SCADA traffic monitoring: A local approach, 2019.

[14] O. Sen, D. van der Velde, S.N. Peters, M. Henze, An approach of replicating multi-staged cyber-attacks and countermeasures in a smart grid co-simulation environment, in: CIRED 2021 Conference, 2021.

[15] Y.-b. Liu, J.-y. Liu, G. Taylor, T.-j. Liu, J. Gou, X. Zhang, Situational awareness architecture for smart grids developed in accordance with dispatcher's thought process: a review, Front. Inf. Technol. Electron. Eng. (2016).

[16] K. Wolsing, E. Wagner, A. Saillard, M. Henze, IPAL: Breaking up silos of protocol-dependent and domain-specific industrial intrusion detection systems, 2021, arXiv:2111.03438.

[17] R. Uetz, C. Hemminghaus, L. Hackländer, P. Schlipper, M. Henze, Reproducible and adaptable log data generation for sound cybersecurity experiments, in: Proceedings of the 37th Annual Computer Security Applications Conference (ACSAC), 2021, http://dx.doi.org/10.1145/3485832.3488020.

[18] B. Klaer, O. Sen, D. van der Velde, I. Hacker, M. Andres, M. Henze, Graph-based model of smart grid architectures, in: SEST, 2020.

[19] M.G. Todeschini, G. Dondossola, Securing IEC 60870-5-104 communications following IEC 62351 standard: lab tests and results, in: AEIT, IEEE, 2020.

[20] WG15, I.E.C. TC57, IEC 62351 security standards for the power system information infrastructure, 2016.

[21] A. Tanveer, R. Sinha, M.M. Kuo, Secure links: Secure-by-design communications in IEC 61499 industrial control applications, IEEE Trans. Ind. Inf. (2020).

[22] J.H. Castellanos, D. Antonioli, N.O. Tippenhauer, M. Ochoa, Legacy-compliant data authentication for industrial control system traffic, in: International Conference on Applied Cryptography and Network Security, Springer, 2017.

[23] M.G. Todeschini, G. Dondossola, R. Terruggia, Impact evaluation of IEC 62351 cybersecurity on IEC 61850 communications performance, 2019.

[24] L.M. Zomlot, Handling uncertainty in intrusion analysis (Ph.D. thesis), Kansas State University, 2014.

[25] W.J. Matuszak, L. DiPippo, Y.L. Sun, Cybersave: situational awareness visualization for cyber security of smart grid systems, in: VizSec, 2013.

[26] A. Sanjab, W. Saad, I. Guvenc, A. Sarwat, S. Biswas, Smart grid security: Threats, challenges, and solutions, 2016, arXiv preprint arXiv:1606.06992.

[27] A. Mavridou, M. Papa, A situational awareness architecture for the smart grid, in: ICGS3/E-Democracy, Springer, 2011.

[28] A.D. Kent, Cyber security data sources for dynamic network research, in: Dynamic Networks and Cyber-Security, World Scientific, 2016.

[29] S. Roschke, F. Cheng, C. Meinel, High-quality attack graph-based IDS correlation, Logic J. IGPL (2013).

[30] R. Zuech, T.M. Khoshgoftaar, R. Wald, Intrusion detection and big heterogeneous data: a survey, J. Big Data (2015).

[31] O. Sen, D. van der Velde, K.A. Wehrmeister, I. Hacker, M. Henze, M. Andres, Towards an approach to contextual detection of multi-stage cyber attacks in smart grids, in: Proceedings of the 2021 International Conference on Smart Energy Systems and Technologies (SEST), 2021.

[32] P. Radoglou-Grammatikis, P. Sarigiannidis, I. Giannoulakis, E. Kafetzakis, E. Panaousis, Attacking IEC-60870-5-104 SCADA systems, in: IEEE SERVICES, IEEE, 2019.

[33] P. Matoušek, Description and analysis of IEC 104 Protocol, Tech. Rep, Faculty of Information Technology, Brno University O Technology, 2017.

[34] L. Kotut, L.A. Wahsheh, Survey of cyber security challenges and solutions in smart grids, in: CYBERSEC, IEEE, 2016.

[35] S. Nazir, S. Patel, D. Patel, Assessing and augmenting SCADA cyber security: A survey of techniques, Comput. Secur. (2017).

[36] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, E.G. Im, Z. Yao, B. Pranggono, H. Wang, Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid SCADA systems, 2012.

[37] W. Wang, Z. Lu, Cyber security in the smart grid: Survey and challenges, Comput. Netw. (2013).

[38] P. Eder-Neuhauser, T. Zseby, J. Fabini, G. Vormayr, Cyber attack models for smart grid environments, Sustain. Energy Grids Netw. (2017).

[39] N. Kshetri, J. Voas, Hacking power grids: A current problem, Computer (2017).

[40] Defense Use Case, Analysis of the cyber attack on the Ukrainian power grid, E-ISAC (2016).

[41] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity (2019).

[42] G. Fernandes, J.J. Rodrigues, L.F. Carvalho, J.F. Al-Muhtadi, M.L. Proença, A comprehensive survey on network anomaly detection, Telecommun. Syst. (2019).

[43] X. Li, X. Liang, R. Lu, X. Shen, X. Lin, H. Zhu, Securing smart grid: cyber attacks, countermeasures, and challenges, IEEE Commun. Mag. (2012).

[44] Z. El Mrabet, N. Kaabouch, H. El Ghazi, H. El Ghazi, Cyber-security in smart grid: Survey and challenges, Comput. Electr. Eng. (2018).

[45] J.J. Chromik, C. Pilch, P. Brackmann, C. Duhme, F. Everinghoff, A. Giberlein, T. Teodorowicz, J. Wieland, B.R. Haverkort, A. Remke, Context-aware local intrusion detection in SCADA systems: a testbed and two showcases, in: SmartGridComm, 2017.

[46] R. Leszczyna, M.R. Wróbel, Evaluation of open source siem for situation awareness platform in the smart grid environment, in: IEEE WFCS, 2015.

[47] M. Vielberth, Security information and event management (SIEM), 2021.

[48] B.D. Bryant, H. Saiedian, A novel kill-chain framework for remote security log analysis with SIEM software, Comput. Secur. (2017).

[49] K.-O. Detken, T. Rix, C. Kleiner, B. Hellmann, L. Renners, SIEM approach for a higher level of IT security in enterprise networks, in: IEEE IDAACS, 2015.

[50] P. Dairinram, D. Wongsawang, P. Pengsart, SIEM with LSA technique for threat identification, in: IEEE ICON, IEEE, 2013.

[51] P. Radoglou-Grammatikis, P. Sarigiannidis, E. Iturbe, E. Rios, S. Martinez, A. Sarigiannidis, G. Eftathopoulos, Y. Spyridis, A. Sesis, N. Vakakis, et al., SPEAR SIEM: A security information and event management system for the smart grid, Comput. Netw. (2021).

[52] M. Angelini, S. Bonomi, E. Borzi, A.D. Pozzo, S. Lenti, G. Santucci, An attack graph-based on-line multi-step attack detector, in: ICDCN '18, 2018.

[53] M.J. Assante, R.M. Lee, The industrial control system cyber kill chain, SANS Institute InfoSec Reading Room, 2015.

[54] R. Kour, A. Thaduri, R. Karim, Railway defender kill chain to predict and detect cyber-attacks, J. Cyber. Security Mobility (2020).

[55] J. Navarro, A. Deruyver, P. Parrend, A systematic survey on multi-step attack detection, Comput. Secur. 76 (2018) 214–249.

[56] F. Kavousi, B. Akbari, Automatic learning of attack behavior patterns using Bayesian networks, in: 6th International Symposium on Telecommunications (IST), IEEE, 2012, pp. 999–1004.

[57] A.A. Ramaki, M. Amini, R.E. Atani, RTECA: Real time episode correlation algorithm for multi-step attack scenarios detection, Comput. Secur. 49 (2015) 206–219.

[58] L. Liang, Abnormal detection of electric security data based on scenario modeling, Procedia Comput. Sci. 139 (2018) 578–582.

[59] T. Bajtoš, P. Sokol, T. Mézešová, Multi-stage cyber-attacks detection in the industrial control systems, in: Recent Developments on Industrial Control Systems Resilience, Springer, 2020, pp. 151–173.

[60] Q. Wang, J. Jiang, Z. Shi, W. Wang, B. Lv, B. Qi, Q. Yin, A novel multi-source fusion model for known and unknown attack scenarios, in: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), IEEE, 2018, pp. 727–736.

[61] S.H. Ahmadinejad, S. Jalili, M. Abadi, A hybrid model for correlating alerts of known and unknown attack scenarios and updating attack graphs, Comput. Netw. 55 (9) (2011) 2221–2240.

[62] R. Shittu, A. Healing, R. Ghanea-Hercock, R. Bloomfield, M. Rajarajan, Intrusion alert prioritisation and attack detection using post-correlation analysis, Comput. Secur. 50 (2015) 1–15.

[63] S. Chamotra, F.A. Barbhuiya, Analysis and modelling of multi-stage attacks, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2020, pp. 1268–1275.

[64] B.E. Strom, A. Applebaum, D.P. Miller, K.C. Nickels, A.G. Pennington, C.B. Thomas, Mitre att&ck: Design and philosophy, Mitre Product Mp (2018) 18–0944.

[65] Y.S. Takey, S.G. Tatikayala, S.S. Samavedam, P.L. Eswari, M.U. Patil, Real time early multi stage attack detection, in: 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 1, IEEE, 2021, pp. 283–290.

[66] C. Moya, J. Hong, J. Wang, Application of correlation indices on intrusion detection systems: Protecting the power grid against coordinated attacks, 2018, arXiv preprint arXiv:1806.03544.

[67] C.-W. Ten, G. Manimaran, C.-C. Liu, Cybersecurity for critical infrastructures: Attack and defense modeling, IEEE Trans. Syst. Man Cybern. Part A 40 (4) (2010) 853–865.

[68] J. Appiah-Kubi, C.-C. Liu, Decentralized intrusion prevention (DIP) against co-ordinated cyberattacks on distribution automation systems, IEEE Open Access J. Power Energy 7 (2020) 389–402.

[69] F.J. Aparicio-Navarro, T.A. Chadza, K.G. Kyriakopoulos, I. Ghafir, S. Lambotharan, B. AsSadhan, Addressing multi-stage attacks using expert knowledge and contextual information, in: 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), IEEE, 2019, pp. 188–194.

[70] A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping, in: Classic Works of the Dempster-Shafer Theory of Belief Functions, Springer, 2008, pp. 57–72.

[71] F.J. Aparicio-Navarro, K.G. Kyriakopoulos, I. Ghafir, S. Lambotharan, J.A. Chambers, Multi-stage attack detection using contextual information, in: MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM), IEEE, 2018, pp. 1–9.

[72] K. Sentz, S. Ferson, et al., Combination of Evidence in Dempster-Shafer Theory, Sandia National Laboratories Albuquerque, 2002.

[73] A. Jøsang, Subjective Logic, Springer, 2016.

[74] L.A. Zadeh, Fuzzy logic, Computer 21 (4) (1988) 83–93.

[75] D. Dubois, Possibility theory and statistical reasoning, Comput. Statist. Data Anal. 51 (1) (2006) 47–69.

[76] L.A. Zadeh, Review of a mathematical theory of evidence, AI Mag. 5 (3) (1984) 81.

[77] J. McHugh, Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory, ACM Trans. Inf. Syst. Secur. 3 (4) (2000) 262–294.

[78] I. Nieto, J.A. Botía, A.F. Gómez-Skarmeta, Information and hybrid architecture model of the OCP contextual information management system, J. UCS 12 (3) (2006) 357–366.

[79] S.M. Othman, N.T. Alsohybe, F.M. Ba-Alwi, A.T. Zahary, Survey on intrusion detection system types, Int. J. Cyber-Secur. Digital Forensics 7 (4) (2018) 444–463.

[80] J.-H. Cho, D.P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T.J. Moore, D.S. Kim, H. Lim, F.F. Nelson, Toward proactive, adaptive defense: A survey on moving target defense, IEEE Commun. Surv. Tutor. 22 (1) (2020) 709–745.

[81] O. Alexander, M. Belisle, J. Steele, MITRE ATT&CK® for industrial control systems: Design and philosophy, The MITRE Corporation, Bedford, MA, USA, 2020.

[82] T. AbuHmed, A. Mohaisen, D. Nyang, A survey on deep packet inspection for intrusion detection systems, 2008, arXiv preprint arXiv:0803.0037.

[83] M.C. Florea, A.-L. Jousselme, E. Bossé, D. Grenier, Robust combination rules for evidence theory, Inf. Fusion 10 (2) (2009) 183–197.

[84] J. Rrushi, H. Farhangi, C. Howey, K. Carmichael, J. Dabell, A quantitative evaluation of the target selection of havex ics malware plugin, in: Industrial Control System Security (ICSS) Workshop, 2015.

[85] D. Kushner, The real story of stuxnet, Ieee Spectrum 50 (3) (2013) 48–53.

[86] C. Osborne, Industroyer: An in-depth look at the culprit behind Ukraine's power grid blackout, ZDNet. Com. Retrieved March 28 (2018) 2020.

[87] G. Brogi, V.V.T. Tong, Terminaptor: Highlighting advanced persistent threats through information flow tracking, in: 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), IEEE, 2016, pp. 1–5.

[88] C.-H. Wang, Y.-C. Chiou, Alert correlation system with automatic extraction of attack strategies by using dynamic feature weights, Int. J. Comput. Commun. Eng. 5 (1) (2016) 1.

[89] C.T. Kawakani, S.B. Junior, R.S. Miani, M. Cukier, B.B. Zarpelão, Intrusion alert correlation to support security management, in: Anais Do XII Simpósio Brasileiro de Sistemas de Informação, SBC, 2016, pp. 313–320.

[90] F. Skopik, I. Friedberg, R. Fiedler, Dealing with advanced persistent threats in smart grid ICT networks, in: ISGT 2014, IEEE, 2014, pp. 1–5.

[91] F.M. Alserhani, Alert correlation and aggregation techniques for reduction of security alerts and detection of multistage attack, Int. J. Adv. Stud. Comput. Sci. Eng. 5 (2) (2016) 1.

[92] P. Holgado, V.A. Villagrá, L. Vazquez, Real-time multistep attack prediction based on hidden markov models, IEEE Trans. Dependable Secure Comput. 17 (1) (2017) 134–147.

[93] Z. Yongtang, L. Xianlu, L. Haibo, A multi-step attack-correlation method with privacy protection, J. Commun. Inform. Netw. 1 (4) (2016) 133–142.

[94] S. Luo, J. Wu, J. Li, L. Guo, A multi-stage attack mitigation mechanism for software-defined home networks, IEEE Trans. Consum. Electron. 62 (2) (2016) 200–207.

[95] R.O. Shittu, Mining intrusion detection alert logs to minimise false positives & gain attack insight (Ph.D. thesis), City University London, 2016.