CyPhERS: A Cyber-Physical Event Reasoning System providing real-time situational awareness for attack and fault response

Nils Müller[a,*], Kaibin Bao[b], Jörg Matthes[b], Kai Heussen[a]

[a] *Wind and Energy Systems Department, Technical University of Denmark, Building 330, Risø campus, 4000 Roskilde, Denmark*
[b] *Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Building 445, Campus North, 76344 Eggenstein-Leopoldshafen, Germany*

Abstract

Cyber-physical systems (CPSs) constitute the backbone of critical infrastructures such as power grids or water distribution networks. Operating failures in these systems can cause serious risks for society. To avoid or minimize downtime, operators require real-time awareness about critical incidents. However, online event identification in CPSs is challenged by the complex interdependency of numerous physical and digital components, requiring to take cyber attacks and physical failures equally into account. The online event identification problem is further complicated through the lack of historical observations of critical but rare events, and the continuous evolution of cyber attack strategies. This work introduces and demonstrates CyPhERS, a **Cy**ber-**Ph**ysical **E**vent **R**easoning **S**ystem. CyPhERS provides real-time information pertaining the occurrence, location, physical impact, and root cause of potentially critical events in CPSs, without the need for historical event observations. Key novelty of CyPhERS is the capability to generate informative and interpretable event signatures of known and unknown types of both cyber attacks and physical failures. The concept is evaluated and benchmarked on a demonstration case that comprises a multitude of attack and fault events targeting various components of a CPS. The results demonstrate that the event signatures provide relevant and inferable information on both known and unknown event types.

Keywords: Cyber-physical systems, Situational awareness, Event identification, Machine learning, Cyber security

1. Introduction

The recent development of critical infrastructure such as power grids and water distribution networks is driven by digitalization and automation. The closed-loop integration of physical processes with computer systems and communication technologies renders them cyber-physical systems (CPSs). In such systems, critical incidents can arise from failures of a variety of interconnected physical and digital devices [1, 2]. The increasing trend of connecting critical infrastructure to the internet adds cyber attacks as another dimension of possible incident causes [3]. Cyber attacks against CPSs constitute a particular risk, as they can entail damage to physical equipment or even humans. Appropriate countermeasures for critical incidents are facilitated by real-time information about affected devices, root causes and physical impact. As incidents can be caused by failure or attack against a variety of physical and digital devices, integrated monitoring of the cyber and physical domain is advantageous [4]. The problem is further complicated by new attack types or unseen physical failures, where no prior knowledge is available for event identification. Consequently, a monitoring system for CPSs is required which provides real-time information about unknown and known types of both cyber attacks and physical failures.

Nowadays, monitoring of the cyber and physical domain is largely conducted in isolated silos, for example through intrusion or fault detection systems. Some recent works propose integration of both through supervised machine learning (ML) [5, 6]. While these approaches excel as they automate event identification, they come with two inherent drawbacks: 1) Substantial amounts of naturally scarce historical attack and fault samples are required. 2) Inability to provide information on unknown event types. These shortcomings motivate the following question: *How can operators of CPSs be provided with relevant information for real-time incident response, given the lack of historical critical event observations and the variety of known and unknown attack and fault types potentially affecting different physical or digital components of a CPS?* To address this question, this work proposes CyPhERS, a new **Cy**ber-**Ph**ysical **E**vent **R**easoning **S**ystem (see Fig. 1). CyPhERS comprises a two-stage process which infers event information such as occurrence, location, root cause, and physical impact from joint evaluation of network traffic and physical process data in real time. Stage 1 creates informative event signatures of unknown and known cyber attacks and faults by combining methods including cyber-physical data fusion, unsupervised multivariate time series anomaly detection, and anomaly type differentiation. In Stage 2, the event signatures are evaluated either by automated matching with a signature database of known events, or through manual interpretation by the operator.

*Corresponding author. E-mail address: nilmu@dtu.dk (N. Müller).

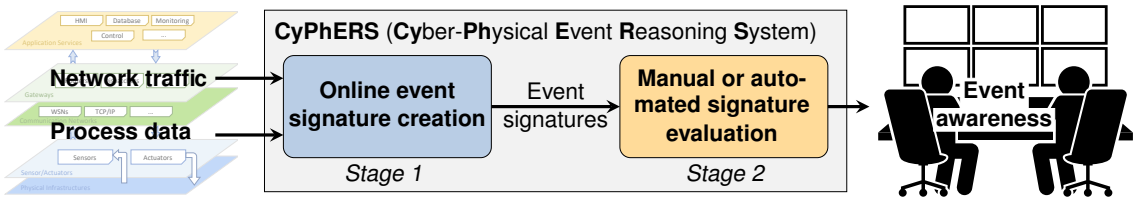


Figure 1: Schematic representation of CyPhERS.

1.1. Related works

Literature on attack or fault identification for CPSs is rich [1, 7–12]. Many works propose methods which are independent of historical event observations, and able to detect both known and unknown events. Table 1 summarizes these approaches and classifies them by the groups A-C. Conceptual differences exist in the considered data sources (process data, network traffic or both), and the information they provide (e.g., event occurrence, location or impact). A description of methods falling under group A-C is provided in Section 1.1.1. Thereafter, a comparison with CyPhERS follows in Section 1.1.2.

1.1.1. Existing event identification concepts for CPSs

Group A. Works summarized in group A are conceptually characterized by the joint evaluation of multiple CPS variables (e.g., multiple sensor readings), where considered variables are derived either exclusively from physical process [13–15] or network traffic data [16]. Unsupervised multivariate time series anomaly detection is applied to detect deviations from normal behavior induced by attacks or faults. Their output is a binary description of the system state (*normal* vs. *abnormal*). Differences among the works mainly exist in the applied ML models and detection rules. The proposed methods can detect unknown and known attack or fault types in real time, given that these entail anomalies in the monitored data. However, as they provide system-wide anomaly flags they only inform about occurrence of an event, while further information, e.g., affected devices or event type, is neglected. Moreover, their restriction to either network or process data limits the events they can detect. While exclusively monitoring network data is blind to physical faults and physical impacts of cyber attacks, limiting to process data misses pure cyber events and only detects cyber-physical attacks when their impact on the system already happened.

Group B. Methods falling under group B apply unsupervised multivariate time series anomaly detection either on process [17–23] or network data [24, 25], similar to group A. Central difference is the provision of feature-level rather than system-wide anomaly flags. By identifying the system variables with the highest individual anomaly scores, affected components are localized in the CPSs. Thus, these concepts provide additional information about detected events to operators. However, the scores only describe the intensity of the deviation from normal behavior, while further characteristics of an anomaly (i.e., polytypic anomaly flags), for example the direction of the deviation or occurrence of missing data, are not considered. Consequently, information on the physical impact or root causes is limited. Moreover, as for group A, monitoring is limited to either network or process data. Consequently, they cannot detect and distinguish anomalies induced by both cyber attacks and physical failures.

Group C. Methods in group C apply unsupervised multivariate time series anomaly detection to features of both process and network data [26–28]. Consequently, they are equally capable of detecting anomalies caused by cyber attacks and physical failures. Moreover, compared to the subset of methods from group A and B which only monitors process data, cyber-physical attacks can be detected earlier and potentially before impacting the process. However, only system-wide monotypic anomaly flags are provided. Therefore, these methods only inform operators about the occurrence of an event without further context information. In contrast to the concepts represented by group A and B, literature on cyber-physical unsupervised anomaly detection for CPSs is rare.

Table 1: Comparison of CyPhERS to existing event sample-independent attack or fault identification concepts.

Group	Concept description	Event coverage	Early detection	Localization	Cause & impact identification
A	Multivariate physical *or* network features, system-wide monotypic anomaly flags	o	o	–	–
B	Multivariate physical *or* network features, feature-level monotypic anomaly flags	o	o	o	–
C	Multivariate physical *and* network features, system-wide monotypic anomaly flags	+	+	–	–
CyPhERS	Multivariate physical *and* network features, feature-level polytypic anomaly flags	+	+	+	+

1.1.2. Comparison of existing concepts to CyPhERS

CyPhERS combines strategies of the described concepts (group A-C), such as fusion of process and network data, unsupervised multivariate time series anomaly detection, and provision of feature-level anomaly flags (see Table 1). It further leverages their associated advantages by considering polytypic anomaly flags. Together, this allows CyPhERS to generate highly informative and recognizable event signatures in Stage 1 (see Fig. 1). Moreover, CyPhERS comprises strategies for manual and automated evaluation of the event signatures (Stage 2).

1.2. Contribution and paper structure

The main contributions of this work are as follows:

- Introduction of CyPhERS, a cyber-physical event reasoning system which provides real-time information about unknown and known attack and fault types in CPSs, while being independent of historical event observations.
- Concept demonstration, evaluation and benchmarking on a CPS study case, considering a variety of attack and fault types affecting several system components.
- Discussion of possible modifications to further improve and extend CyPhERS.

The remainder of the paper is structured as follows: In Section 2, CyPhERS is conceptually introduced. The considered demonstration case is presented in Section 3. Section 4 explains methodological details of CyPhERS, and demonstrates its implementation on the given case. In Section 5, results of applying CyPhERS on the study case are presented. Finally, demonstration results are discussed in Section 6, followed by a conclusion and view on future work in Section 7.

2. Conceptual introduction of CyPhERS

This section introduces the two stages of CyPhERS at a conceptual level. Section 2.1 provides details on the online event signature creation (Stage 1). Thereafter, signature evaluation (Stage 2) is explained in Section 2.2. Finally, Section 2.3 provides a taxonomy of CPSs that CyPhERS can be applied to.

2.1. Online event signature creation (Stage 1)

Stage 1 of CyPhERS is schematically outlined in Fig. 2. To provide event signatures that contain information about occurrence, location, root cause and physical impact of unknown and known types of both attacks and faults in real time, CyPhERS combines several strategies, which are introduced in the following.

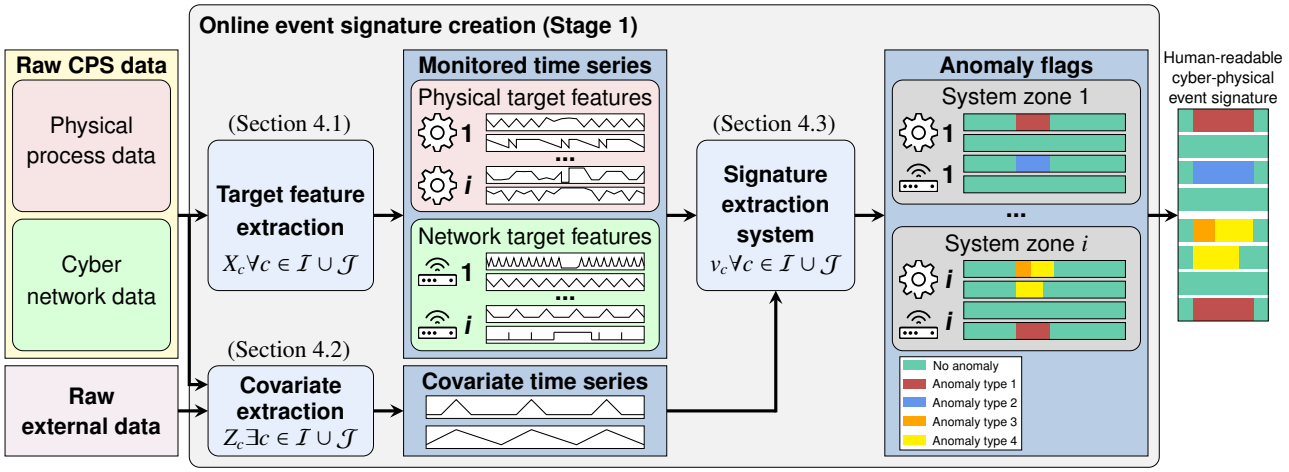


Figure 2: Schematic overview of CyPhERS' online event signature creation (Stage 1).

2.1.1. Multi-domain information

First element is the fusion and joint evaluation of physical process and cyber network data of a CPS (see Fig. 2), which is required for real-time detection and differentiation of attacks and failures. Moreover, cyber-physical monitoring allows to determine whether a cyber attack already entails physical impact or detect it early enough in network traffic to mitigate damage through countermeasures such as isolation of affected devices. Other data such as maintenance activities and schedules (human domain) can potentially be included to take human errors into consideration.

2.1.2. Feature-level monitoring

Second element is the individual monitoring of multiple system variables (see Fig. 2), aiming for a more detailed picture of critical events in contrast to monitoring the system state. Within CyPhERS, the set of monitored variables is multivariate in two dimensions:

- Monitoring variables of multiple physical and network components (*cross-device multivariate monitoring*).
- Monitoring several variables of a single component (*in-device multivariate monitoring*).

While cross-device multivariate monitoring aims at localizing affected devices, in-device multivariate monitoring is supposed to provide further details about a component's abnormal behavior. The monitored system variables are derived, for example, from sensor measurements or traffic of network devices. In the following, the resulting set of monitored system variables is referred to as *target features*, with $\mathcal{I}$ being the physical target feature and $\mathcal{J}$ the network target feature subset. The time series of an arbitrary target feature c is defined as $X_c = \{x_1^c, x_2^c, ..., x_N^c \mid x_i^c \in \mathbb{R}\ \forall i\}$. Details on the extraction of target features follow in Section 4.1.

2.1.3. Unsupervised time series anomaly detection considering covariates

CyPhERS applies unsupervised time series anomaly detection[1] to identify the occurrence of critical events. Time series models are applied to provide normal behavior references of individual target features, which are compared to actual observations for detecting abnormal system behavior. Central argument is the independence of historical event observations, and ability to detect both known and unknown event types, given that they entail anomalies. Monitoring target features in time series format allows to detect deviations from normal behavior which are only abnormal in a specific temporal context (local anomalies) [29]. Furthermore, covariates are considered for time series modeling. Covariates allow to provide models with further system internal or external information (see Fig. 2). As a result, situational anomalies can be detected which are only abnormal in the context of the provided covariates. A covariate time series of a target feature c is defined as $Z_c = \{z_1^c, z_2^c, ..., z_N^c \mid z_i^c \in \mathbb{R}\ \forall i\}$. Covariate extraction is detailed in Section 4.2.

2.1.4. Differentiation of anomaly types

The fourth element is the semantic differentiation of multiple anomaly types (see Fig. 2). In case that an anomaly is flagged for a target feature c, it is further classified based on information such as the direction of the deviation (e.g., abnormally *many* data packets received by network device X). Considering various anomaly types provides additional information for identification of event root causes and impact. The series of flags provided by the signature extraction system for a target feature c is given as $v_c = \{v_1^c, v_2^c, ..., v_N^c \mid v_i^c \in \{-2, -1, 0, 1, 2\}\ \forall i\}$. A detailed explanation of the signature extraction system, including anomaly types, follows in Section 4.3.

2.1.5. Event signature visualization

By covering multiple domains, system variables and anomaly types, Stage 1 of CyPhERS provides dense information about critical events in form of anomaly flag series of a set of target features. To ease extraction of these information, the flag series are re-organized by grouping them for each system zone of a CPS (see Fig. 2). A system zone comprises a group of physical components and network devices which are directly related, such as a set of physically connected process units and a programmable logic controller (PLC) monitoring and controlling them. Due to the logical relation within a system zone, anomaly flags of different target features can be quickly related. As a result, Stage 1 of CyPhERS provides information-rich and human-readable event signatures.

2.2. Signature evaluation (Stage 2)

The concept of CyPhERS' Stage 2 is schematically depicted in Fig. 3. In Stage 2, the event signatures of Stage 1 are evaluated, which can be realized through interpretation by human operators as well as by automated reasoning systems. The provided signatures are event specific and distinguishable. Thus, for known attacks or faults they can be predefined and stored in a database. Once Stage 1 indicates occurrence of an event, the associated signature can be compared to the database. In case of a signature match, the stored information about event type, affected component, root cause, and/or physical impact provide the event hypothesis. Signature matching can either be conducted by the operator through visual comparison or an automated evaluation system. One automation approach would be the transformation of a signature into a set of rules, e.g.,

```
flagging of (anomaly type 1 in target feature X) and (type 2 in target feature Y)
indicates (device A being targeted by attack type B causing physical impact C).
```

Signatures can also be defined for unknown event types based on partial knowledge. In this instance, they carry reduced information, e.g.,

```
flagging of (anomaly type 1 in target feature X)
indicates (device A failure [type unknown] causing physical impact B).
```

In case of unknown or undefined signatures, automated evaluation cannot infer event information. In these situations, operators may deduce information such as affected system components or physical impact based on process expertise. The minimum information CyPhERS provides in any event case is the occurrence of abnormal system behavior.

[1] Sometimes referred to as self-supervised anomaly detection.

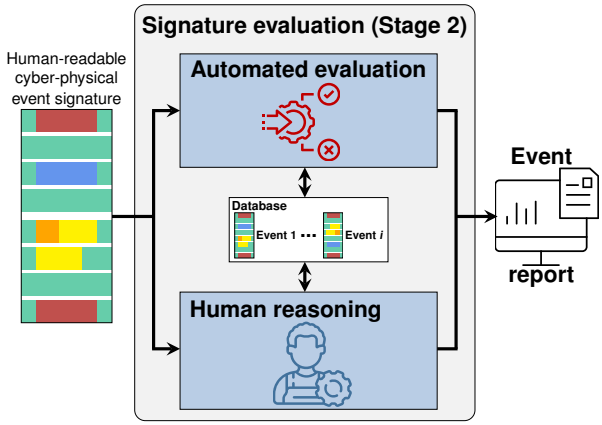


Figure 3: Schematic overview of CyPhERS' signature evaluation (Stage 2).

2.3. Application scope of CyPhERS

The CyPhERS concept is applicable to CPSs which fulfill the following requirements: 1) Real-time data availability of and 2) learnable normal behavior patterns in both process and network traffic data. CPS traffic typically exhibits periodical patterns as it arises from automated processes (e.g., polling) in static network architectures with a consistent number of devices [30]. Moreover, many technical systems exhibit learnable process patterns, including manufacturing processes [31], transportation systems [32], water distribution systems [33], and spacecraft [34]. A potentially complicating factor is process volatility and randomness, which, for example, can result from unpredictable weather or user influences. While CyPhERS is conceptually applicable to most CPS types, its implementation requires some case specific adaptations given the heterogeneity of processes and network architectures. These include selection or definition of target features, anomaly types, and known event signatures. Among instances of the same system type (e.g., health monitoring system of a specific provider), the implementation of CyPhERS is fully transferable.

3. Demonstration case description

This section describes the considered demonstration case, which is introduced by Faramondi *et al.* in [35]. The underlying CPS is detailed in Section 3.1. Thereafter, included attack and fault scenarios, and the associated dataset are described in Section 3.2. The demonstration case was selected as the only complete cyber-physical dataset which describes various types of both cyber attacks and physical faults affecting different system components. The physical process represents a typical CPS laboratory setup.

3.1. Cyber-physical structure of demonstration case

Fig. 4 provides a simplified overview of the investigated CPS. The system distributes water between several tanks, where one process cycle is defined by a filling/emptying process of all tanks. This procedure is continuously repeated, making it a cyclical process. The physical system comprises eight water tanks (T1-T8), a reservoir, and several sensors and actuators. Actuators include valves (V10-V22) and pumps (P1-P6), which realize the water distribution between the tanks. Note that for better readability not all sensors and actuators are depicted in Fig. 4. Pressure sensors in T1-T8 and flow sensors (FSs) (FS1-FS4) measure tank fill levels H and water flows F, respectively. The process is monitored and controlled by a typical supervisory control and data acquisition (SCADA) architecture consisting of the sensors and

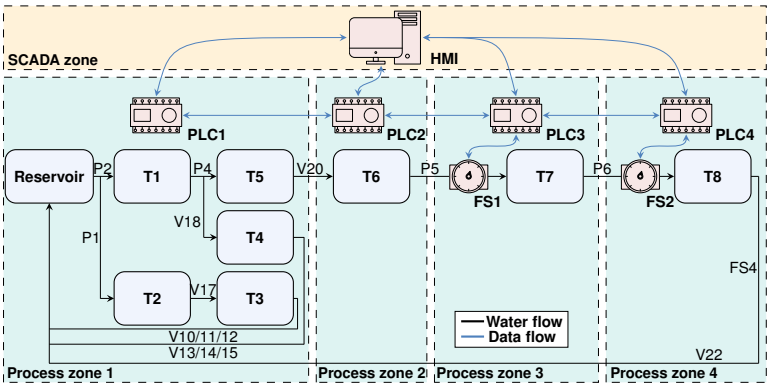


Figure 4: Simplified overview of the demonstration case CPS based on schematic representations in [35].

actuators (field instrumentation control layer), four PLCs (process control layer), and a SCADA workstation, including a human-machine interface (HMI) and data historian (supervisory control layer). The SCADA workstation, HMI and data historian together are referred to HMI in the following. The communication is conducted via MODBUS TCP/IP protocol. The process consists of four stages, each of which is controlled by one of the four PLCs. The PLCs send sensor values to the SCADA workstation, so that physical process data can be stored centrally on the historian. Moreover, values of tank fill levels H and water flows F are directly exchanged between the PLCs and FSs, which they require to control tank fill levels by (de-)activating pumps and valves. While most sensors and actuators are connected to the PLCs via wired links, FS1 and FS2 are MODBUS TCP/IP sensors with own IP addresses. Thus, the communication network in total consists of seven devices, which are marked red in Fig. 4. Note that an additional Kali Linux machine was used to launch cyber attacks, which is not depicted in Fig. 4.

3.2. Threat scenarios and dataset

The dataset comprises four partitions, each covering multiple process cycles. While the first partition describes a normal operation scenario (S0) the remaining three describe attack and fault scenarios (S1-S3). S1-S3 exhibit an increasing level of event type variety. S1 includes several physical component breakdowns and water leaks as well as man-in-the-middle (MITM) attacks. In MITM attacks a perpetrator positions himself between two victim devices to relay and potentially alter communication while the victims assume a direct communication [36]. In the present case, the attacker modifies H values send by victim one, which are required by victim two to control fill levels of tanks in the respective process zone. In S2, denial-of-service (DoS) attacks are additionally included. These cause a disconnection of the targeted device from the network by flooding it with requests [37]. In the investigated dataset, several DoS attack variants are used to disconnect specific PLCs or the SCADA workstation. Finally, S3 adds scanning attacks. Scanning is a reconnaissance method used by attackers to determine possible vulnerabilities by searching for services and service identifiers in a target network or host [38]. The given case considers several scanning attacks, which are used to gather information about various PLCs. Note that S1-S3 comprise of unique attack and fault events affecting various components and communications links in the system, so that each scenario represents an entirely new case. In total, eight MITM, five DoS, and seven scanning attacks as well as three water leaks and six sensor or pump breakdowns are included. The considered attack types are among the most relevant for CPSs [39–42]. Table 2 lists the raw physical and network features of the dataset. The physical data has a constant one-second resolution, resulting in 3429 (S0), 2421 (S1), 2105 (S2), and 1255 (S3) samples. The network data during normal operation (S0) on average contains 2265 packets per second, and in total comprises ~7.8 (S0), ~5.5 (S1), ~5.2 (S2), and ~5.9 (S3) $\times 10^6$ packets. For a more detailed explanation of the demonstration case the reader is referred to [35].

Table 2: Raw network and process features within the study case.

No.	Physical features	No.	Network features
1	Timestamp	1	Timestamp
2-9	Fill level H of T1-T8	2-3	IP address (src. & dst.)[a]
10-15	Activation state S of P1-P6	4-5	MAC address (src. & dst.)
16-19	Flow value F measured by FS1-FS4	6-7	Port (src. & dst.)
		8	Protocol
20-41	Activation state S of V1-V22	9	TCP flags
		10	Packet size
		11	MODBUS function code
		12	MODBUS response value
		13-14	No. of packets (src. & dst.)

[a]Src. and dst. refer to source and destination, respectively.

4. Methodology and implementation of CyPhERS

This section first provides details on the methodology and case-specific implementation of the online event signature creation (Stage 1). This includes extraction of target features (Section 4.1) and covariates (Section 4.2) as well as the signature extraction system (Section 4.3). Thereafter, methodology and case-specific implementation of the event signature evaluation (Stage 2) is detailed in Section 4.4.

4.1. Target feature extraction

Methodology. The landscape of CPSs is characterized by a pronounced heterogeneity, resulting from factors such as the diversity of physical components (e.g., tanks, engines or batteries) and communication protocols (e.g., MODBUS, UDP or DNP). Thus, a general set of target features cannot be defined. Nevertheless, guidelines for extraction of relevant features can be provided.

CyPhERS considers sensor measurements and actuator states as raw physical process data. Monitoring such data has two motivations, namely the identification of i) true physical events and ii) manipulation of process-relevant data. The former requires features which represent the behavior (e.g., state or output) of all physical components of a CPS to 1)

localize affected devices and 2) infer the impact on them. For the latter, readings exchanged among devices for automated process control need to be monitored. For processes exhibiting low randomness and noise levels, raw sensor readings can directly be used as physical target features. Otherwise, further processing is required to extract the available information. Strategies include resampling (e.g., moving average) or derivation of features which describe a component's behavior on a simplified level (e.g., on/off state).

In CyPhERS, network target features are extracted from OT network traffic[2] of a CPS. Traffic monitoring is considered to retain information for i) localizing affected digital devices, and ii) concluding on attack types. For the former, traffic of each network device is monitored separately. The latter requires extraction of several features for each device in order to provide sufficient information for distinguishing attack types. Detecting and differentiating attacks is challenged by the fact that some solely concern individual packets (e.g., sending a malicious control command), while others are only visible from the context of multiple packets (e.g., replaying valid data transmission). Therefore, CyPhERS considers both extraction of network features which i) are sensible to values of single packets (e.g., count of packets send from unknown source IP or MAC addresses), and ii) set multiple packets into context (e.g., average number of received packets within a time period).

Case-specific implementation. In the present demonstration case, noise in the raw physical features is comparatively low, which allows direct use as target features with original per-second resolution. For the sake of clarity, not all available sensor readings and actuators states are taken into account. Instead, fill levels H of the water tanks are considered, as they allow to monitor all four process steps, and describe the behavior of the most important components. The network traffic is separately monitored for PLC1-4, FS1-2 and the HMI. For this purpose, it is first filtered by the destination MAC addresses and then evaluated for each second through several features. An overview of the resulting set of physical $\mathcal{I}$ and network target features $\mathcal{J}$ can be found in Table 3.

Table 3: Overview of target features extracted in the study case.

No.	Physical target features $\mathcal{I}$	No.	Network target features[a] $\mathcal{J}$
1-8	Fill level H of T1-T8	1-7	Average packet size s_{packet}
		8-14	Packet count n_{packet}
		15-21	New src. IP/MAC count $n_{\text{IP/MAC}}$
		22-28	New TCP flags count n_{TCP}
		29-35	Mean of encoded TCP flags[b] μ_{TCP}
		36-42	Different src. ports count n_{ports}

[a]For PLC1-4, FS1-2 and HMI, each considered as destination device.
[b]Each flag type is encoded as a specific integer.

4.2. Covariate extraction

Methodology. As for target features, extraction of covariates is dependent on the respective CPS. In any case, however, they should represent process-relevant context such as environmental conditions or human interactions. Examples are irradiation for solar plants or user intervention in case of self-driving cars. Such information allows models to learn whether an event is normal or abnormal given the current context. For example, irradiation facilitates differentiation of normal weather- or malicious attacker-induced drops of solar feed-in power. CPSs often exhibit repeating processes. In such cases, covariates which inform models about the current position in a cycle (e.g., process stage or time of the day) provide valuable context information. CyPhERS considers sine and cosine transformation [43] of such cyclical covariates, which is schematically represented in Fig. 5. Let $Z_c = \{z_1^c, z_2^c, ..., z_N^c \mid z_i^c \in \mathbb{R}\ \forall i\}$ be a cyclical covariate time series of length N for a target feature c. Within one cycle, values of Z_c are linearly increasing on the range $z_i^c \in [\min(Z_c), \max(Z_c)]$. Due to the jump discontinuity between two cycles, a linear representation cannot properly describe the continuity of cyclical processes. To eliminate the discontinuity, values of Z_c are transformed according to

$$z_{\text{sin},i}^c = \sin\left(\frac{2\pi z_i^c}{\max(Z_c)}\right), \text{ and } z_{\text{cos},i}^c = \cos\left(\frac{2\pi z_i^c}{\max(Z_c)}\right), \tag{1}$$

[2]Other potential data sources include system logs and key performance indicators of digital devices (e.g., memory usage).

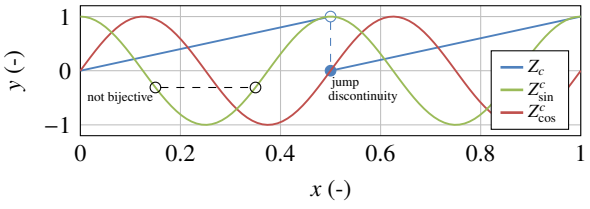


Figure 5: Illustration of sine and cosine transformation of cyclical covariates.

$\forall i \in [1, N]$, resulting in the two new covariate time series $Z^c_{\sin}$ and $Z^c_{\cos}$. The use of both sine and cosine transformation is required as they individually are not bijective, which would lead to ambiguity in the transformed covariate (see Fig. 5).

Case-specific implementation. In the investigated study case, the normal progress of a process cycle is considered as covariate time series $P_c = \{p^c_1, p^c_2, ..., p^c_N \mid p^c_i \in \mathbb{R}\ \forall i\}$ of length N, $\forall c \in \mathcal{I}$. Using S0 (normal operation scenario) the usual duration d_c of a process cycle is determined. Based on the duration, values of P_c are defined on the range $p^c_i \in [0, d_c]$. The additional covariate time series $P^c_{\sin}$ and $P^c_{\cos}$ are extracted $\forall c \in \mathcal{I}$ by applying sine and cosine transformation on the values of P_c according to (1) with $\max(P_c) = d_c$.

4.3. Signature extraction system

The signature extraction system (see Fig. 2) follows the idea of applying individual anomaly detection and classification pipelines to each target feature. Compared to the joint processing in one large model, several advantages exist: 1) The complexity of time series models can be adjusted to individual target features. As features in CyPhERS originate from very different sources (process data and network traffic), they exhibit strong variations in characteristics such as observation rates and noise levels. 2) Independent definition of abnormal behavior for each system component. By selecting covariates for individual target features, it is possible to define which context models should consider when deciding whether a component is behaving abnormally. 3) Promotes a distributed implementation of CyPhERS on edge devices. As attackers can manipulate data to hide induced physical impact from centralized monitoring [9], this facilitates detection of hidden process manipulations.

The anomaly detection and classification pipelines are explained in Section 4.3.1. After that, Section 4.3.2 addresses the forecasting models which are applied within the pipelines. Finally, the procedure for automated implementation of the signature extraction system is detailed in Section 4.3.3.

4.3.1. Anomaly detection and classification pipelines

Methodology. The anomaly detection and classification pipeline of a target feature c is schematically depicted in Fig. 6. A pipeline consists of two fundamental and consecutive steps, namely a time-series forecasting model and an anomaly detector. Given $X_c = \{x^c_1, x^c_2, ..., x^c_N \mid x^c_i \in \mathbb{R}\ \forall i\}$ and $Z_{c,1}, ..., Z_{c,n} = \{\{z^c_{1,1}, z^c_{1,2}, ..., z^c_{1,N}\}, ..., \{z^c_{n,1}, z^c_{n,2}, ..., z^c_{n,N}\} | z^c_{j,i} \in \mathbb{R} \forall (j,i)\}$ of a target feature c, the forecasting model predicts the expected value $\hat{x}^c_t$ at time t based on lag values $x^c_{t-w}, ..., x^c_{t-1}$ and covariates $z^c_{1,t}, ..., z^c_{n,t}$ according to

$$\hat{x}^c_t = \Phi\left([x^c_{t-w}, ..., x^c_{t-1}], [z^c_{1,t}, ..., z^c_{n,t}]\right), \tag{2}$$

where w is the length of the history window and n the number of covariates. Depending on the target feature, $x^c_{t-w}, ..., x^c_{t-1}$ and $z^c_{1,t}, ..., z^c_{n,t}$ are only partially used as model input, which is specified in Section 4.3.2.

Next, the expected value $\hat{x}^c_t$ and ground truth x^c_t are forwarded to the anomaly detector. In CyPhERS, anomalies are flagged based on multiple consecutive observations instead of only the most recent one, aiming at reducing noise-induced false positives (FPs). For that purpose, the detector first calculates the average of the distances of the last l observations to their respective expected values according to

$$\varepsilon^c_t = \frac{\sum_{j=0}^{l-1} |x^c_{t-j} - \hat{x}^c_{t-j}|}{l}. \tag{3}$$

Based on ε^c_t and further characteristics of the current target feature observations, the detector then differentiated several anomaly types. While the definition of meaningful anomaly types is facilitated by taking process specificities of a CPS into account, some widely applicable ones exist. Table 4 defines some of them. The listed types can be transferred into the detector's anomaly flag decision function according to

$$v^c_t = \begin{cases} 2 & \text{if } \overbrace{(\varepsilon^c_t > \tau_c)}^{\text{Detection}},\ \overbrace{(x^c_t > \hat{x}^c_t)}^{\text{Direction}} \text{ and } [\overbrace{\exists (x^c_i = x^c_{i-1} \text{ or NaN}) \in X^c_J}^{\text{Disruption}}] \\ 1 & \text{if } (\varepsilon^c_t > \tau_c),\ (x^c_t > \hat{x}^c_t) \text{ and } [\nexists (x^c_i = x^c_{i-1} \text{ or NaN}) \in X^c_J] \\ -1 & \text{if } (\varepsilon^c_t > \tau_c),\ (x^c_t < \hat{x}^c_t) \text{ and } [\nexists (x^c_i = x^c_{i-1} \text{ or NaN}) \in X^c_J] \\ -2 & \text{if } (\varepsilon^c_t > \tau_c),\ (x^c_t < \hat{x}^c_t) \text{ and } [\exists (x^c_i = x^c_{i-1} \text{ or NaN}) \in X^c_J] \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where τ_c is a target feature-specific threshold, and $X^c_J = x^c_{t-l+1}, ..., x^c_t$ the ground truth values within the distance averaging window. The automated adaption of τ_c to individual target features is described in Section 4.3.3. The differentiation

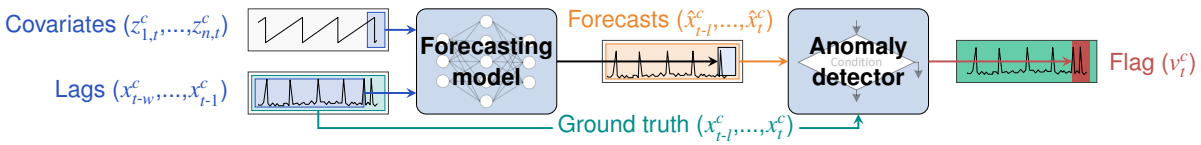
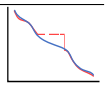
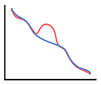
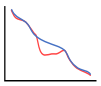
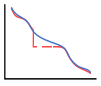
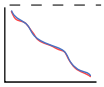


Figure 6: Schematic overview of the anomaly detection and classification pipeline of a target feature c.

Table 4: Description of some general anomaly types.

Flag v	Anomaly type	Description	Schematic
2	Positive disrupted	Target feature positively differentiates from normal behavior, exhibiting static or NaN values.	
1	Positive undisrupted	Target feature positively differentiates from normal behavior, not exhibiting static or NaN values.	
−1	Negative undisrupted	Target feature negatively differentiates from normal behavior, not exhibiting static or NaN values.	
−2	Negative disrupted	Target feature negatively differentiates from normal behavior, exhibiting static or NaN values.	
0	No anomaly	No abnormal behavior.	

between static and non-static behavior is neglected for target features exhibiting static values during normal operation, which in this event reduces (4) to

$$v_t^{c*} = \begin{cases} 2 & \text{if } \overbrace{(\varepsilon_t^c > \tau_c)}^{\text{Detection}},\ \overbrace{(x_t^c > \hat{x}_t^c)}^{\text{Direction}} \text{ and } \overbrace{[\exists (x_i^c = \text{NaN}) \in X_J^c]}^{\text{Disruption}} \\ 1 & \text{if } (\varepsilon_t^c > \tau_c),\ (x_t^c > \hat{x}_t^c) \text{ and } [\nexists (x_i^c = \text{NaN}) \in X_J^c] \\ -1 & \text{if } (\varepsilon_t^c > \tau_c),\ (x_t^c < \hat{x}_t^c) \text{ and } [\nexists (x_i^c = \text{NaN}) \in X_J^c] \\ -2 & \text{if } (\varepsilon_t^c > \tau_c),\ (x_t^c < \hat{x}_t^c) \text{ and } [\exists (x_i^c = \text{NaN}) \in X_J^c] \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Case-specific implementation. For the investigated study case, a distance averaging window $l = 10$ is selected for calculation of ε_t^c according to (3), $\forall c \in \mathcal{I}$ and $\mathcal{J}$. Considered anomaly types correspond to the ones listed in Table 4. For H_{T7} and H_{T8} as well as $n_{\text{IP/MAC}}$ and n_{TCP} of all network devices, the detector's decision function reduces to (5) as they contain static values during S0.

4.3.2. Forecasting models

Methodology. As some CPSs come with computational constraints, keeping model complexity at a required minimum is favorable. The forecasting models used within the anomaly detection and classification pipelines (see Fig. 6) are interchangeable, which allows adapting their complexity to the specific characteristics of a target feature. Four target feature property classes are differentiated in CyPhERS. These are listed in Table 5 together with feature examples and recommended forecasting model types. Class A comprises target features exhibiting unchanged values during normal operation. In these cases, a trivial constant-value forecast is sufficient, which simplifies (2) to $\hat{x}_t^c = a_c$, where a_c corresponds to the constant value of the respective target feature during normal operation. In class B, continuous target features with covariate availability are summarized, which typically includes physical sensor measurements. Due to the additional information covariates provide, less complex forecasting models can be considered. The use of simple regression models (e.g., linear regression) for modeling class B features according to (2) is recommended. Target features may also exhibit discrete values (Class C) as in the case of actuator states. In this event, the use of ensemble models such as a random forest (RF) regressor [44] is proposed for modeling features according to (2). The rationale behind using ensemble models for class C features is their internal process of discretizing continuous variables, which facilitates prediction of sudden steps. Moreover, they

Table 5: Target feature property classes and proposed forecasting models.

Class	Condition[a]	Model type	Feature example
A	Target features with solely constant values	Constant value	Occurrence of new IP address
B	Continuous target features with covariates	Simple regression (e.g., linear or RF)	Sensor measurements
C	Target features with discrete levels	Ensemble regression (e.g., RF)	States of actuators
D	Continuous target features without covariates	Deep-learning (e.g., LSTM)	Number of transmitted network packets

[a]Constant, continuous and discrete behavior relates to normal operation.

are known for robustness, few parameters to tune and good performance compared to many other standard methods on a variety of prediction problems [45, 46]. For a detailed theoretical description of ensemble models, the reader is referred to [47]. Finally, Class D comprises continuous target features without availability of covariates. Since covariates are neglected, (2) reduces to

$$\hat{x}_t^c = \Phi([x_{t-w}^c, ..., x_{t-1}^c]). \tag{6}$$

Due to lack of additional information through covariates, more advanced models are required, which are capable of exploiting short and long-term temporal dependencies within a target feature. Thus, deep-learning-based forecasting models are suggested for class D features. Prominent representatives are long short-term memory (LSTM) networks [48]. LSTMs constitute a special architecture of neural networks capable of capturing complex long-term temporal dependencies in sequential data, which makes them well suited for time series forecasting. Many works have demonstrated their superior performance in various areas [49–51]. For a detailed theoretical description of LSTM networks, the reader is referred to [52].

Case-specific implementation. In the investigated demonstration case, $n_{\mathrm{IP/MAC}}$ and n_{TCP} constitute constant target features (class A), which hence are modeled with trivial constant value forecasts. All considered physical target features of the study case (H_{T1}-H_{T8}) fall into class B. Thus, RF is selected as the regression model. As P_c, P_{sin}^c and P_{cos}^c are available as supporting covariate time series, a comparatively short history window of $w = 10$ is considered. Table 6 lists the tuned hyperparameters and their respective search spaces. The underlying model selection procedure is detailed in Section 4.3.3. The RF models are implemented in Python using the forecasting-library *Darts* [53].

For forecasting the network target features, no supporting covariates are available. Thus, the set of non-constant network target features $\mathcal{J}_{\mathrm{nc}}$ falls into property class D. Consequently, concerned features are modeled using LSTM networks. To allow a LSTM to capture long-term dependencies, the history window is extended to $w = 300$, covering an entire process cycle. In Table 6, the tuned hyperparameters and their respective search spaces are given. Implementation of the LSTM models is realized in Python using the forecasting-library *Darts* [53].

Table 6: Hyperparameters and search spaces for RF and LSTM.

No.	Hyperparameter[a]	Search space
RF (physical target features $\mathcal{I}$)		
1	Number of trees	1, 50, 100, 250, 500, 1000
2	Nr. of features for best split determination	1, 3, 5, 7, 9, 11
LSTM (non-constant network target features $\mathcal{J}_{\mathrm{nc}}$)		
1	Number of LSTM layers	1, 2, 3
2	Batch size	32, 64
3	Number of epochs	100, 200, 500
4	Dropout rate	0, 0.2
5	Number of nodes in LSTM layers	20, 50, 100

[a]For other hyperparameters, default values from [53] are used.

4.3.3. Automated model and detector tuning procedure

Methodology. Implementing the signature extraction system requires performing the same automated tuning procedure for the detection and classification pipelines of all target features, which is schematically represented in Fig. 7. First step is the selection of forecasting models, which is concerned with the determination of appropriate target feature-specific hyperparameters. The hyperparameters are selected based on a grid search using time series cross-validation on a training/validation set $X_{\mathrm{train/val}}^c$ of normal operation data. For each fold $X_{\mathrm{fold}}^c = \{x_1^c, x_2^c, ..., x_{N_{\mathrm{fold}}}^c \mid x_i^c \in \mathbb{R}\ \forall i\}$ of the cross-validation, data is scaled to $\bar{x}_i^c \in [0, 1]$ before training by

$$\bar{x}_i^c = \frac{x_i^c - \min\left(X_{\mathrm{train}}^c\right)}{\max\left(X_{\mathrm{train}}^c\right) - \min\left(X_{\mathrm{train}}^c\right)}, \tag{7}$$

$\forall x_i^c \in X_{\mathrm{fold}}^c$, where X_{train}^c corresponds to the first 75 % of the respective fold. In case a covariate time series $Z_{\mathrm{train/val}}^c$ or multiple of them are used, they are scaled in the same way as target features in (7). To finalize model selection, the resulting forecasting models are retrained on the respective full training/validation set $X_{\mathrm{train/val}}^c$, again on values scaled according to (7), where now $X_{\mathrm{train}}^c = X_{\mathrm{train/val}}^c$.

After selecting appropriate forecasting models, the anomaly detectors (see Fig. 6) of all pipelines are fitted to the respective target feature. For that purpose, a target feature-specific threshold τ_c is determined for each pipeline as follows. First, the associated forecasting model is used to predict the expected values of a normal operation test set X_{test}^c based on a rolling one-step ahead forecast. The expected values are required to calculate all averaged distances of the test set $E_{\mathrm{test}}^c = \{\varepsilon_1^c, \varepsilon_2^c, ..., \varepsilon_{N_{\mathrm{test}}}^c \mid \varepsilon_i^c \in \mathbb{R}\ \forall i\}$. Based on E_{test}^c and a threshold factor f, the feature-specific threshold is determined according to

$$\tau_c = f \cdot \max\left(E_{\mathrm{test}}^c\right). \tag{8}$$

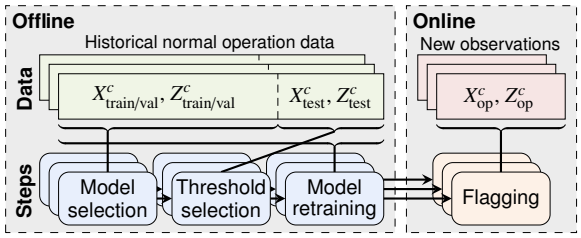


Figure 7: Automated tuning procedure of the anomaly detection and classification pipelines constituting the signature extraction system.

If a target feature exhibits a high noise level or if non-optimal hyperparameters are selected, $\max(E^c_{\text{test}})$ increases due to a weaker forecasting performance. Thus, according to (8), τ_c automatically adapts to the prediction performance of the respective forecasting model. Objective of the proposed adaptive threshold is the reduction of FPs.

Next, to fully exploit available historical data, the selected forecasting models are retrained on the respective totality of normal operation observations ($X^c_{\text{train/val}} + X^c_{\text{test}}$) and potentially covariates ($Z^c_{\text{train/val}} + Z^c_{\text{test}}$).

After tuning the forecasting models and anomaly detectors, the detection and classification pipelines are ready for operation and can be applied to create anomaly flags for newly incoming observations X^c_{op}, which again are scaled according to (7) with $X^c_{\text{train}} = X^c_{\text{train/val}}$. Finally, the resulting anomaly flags of individual pipelines are grouped for each system zone of a CPS to obtain event signatures as output of CyPhERS' Stage 1.

Case-specific implementation. In the considered study case, $X^c_{\text{train/val}}$ and $Z^c_{\text{train/val}}$ are taken from the first 75 % and X^c_{test} and Z^c_{test} from the remaining 25 % of S0, $\forall c \in \mathcal{I}$ and $\mathcal{J}$. X^c_{op} and Z^c_{op} are provided by the three attack and fault scenarios S1-S3, $\forall c \in \mathcal{I}$ and $\mathcal{J}$. Moreover, the threshold factor f in (8) is specified to $f = 1.5$. Consequently, an anomaly within a target feature c only is flagged if the average distance over the last 10 observations exceeds the biggest average distance during normal operation by at least 50 %. The target features are grouped according to the system zones depicted in Fig. 4.

4.4. Signature evaluation (Stage 2)

Methodology. Signature evaluation in CyPhERS is based on manually or automatically matching anomaly flags provided by Stage 1 with a database of known event signatures. The possible attack and fault types that can affect a CPS depend on the system's physical and digital components and architectures. Consequently, a set of event signatures which is valid across all systems cannot be defined. However, some general anomaly flag interpretation rules which hold for most CPSs have been identified, and are listed in Table 7. These general rules can be applied to create signature databases. They range from simple principles, such as indication of affected system components through appearance of anomaly flags in the associated target features (rule 1), to more complex anomaly flag patterns which point to specific attack or fault types (e.g., rule 12.2). Therefore, also signatures of different information detail can be defined, ranging from unclassified signatures,

Table 7: General anomaly flag interpretation rules for the definition of event signatures.

No.	Rule description
1	Appearance of anomaly flags in a target feature indicates that the underlying physical or network component is affected (localization).
2	Anomaly flags exclusively in physical target features points towards a physical failure.
3	Anomaly flags exclusively in network target features, including flags which indicate malicious activities[a], points towards a cyber attack.
4	Anomaly flags exclusively in network target features, without flags indicating malicious activities[a], points towards a network (device) failure.
5	Flags in both physical and network target features, including flags indicating malicious activities[a], points towards a cyber-physical attack.
6	Flags in ph. and netw. target features, w/o flags indicating mal. activities[a], indicate a network (device) failure entailing physical impact.
7	Anomaly flags exclusively in physical target features of one component indicates a local failure without impact on other components.
8	Physically plausible and coherent flags in physical target features of multiple components indicates a problem of their physical connection.
9	Anomaly flags exclusively in network target features of one device indicates a local device problem w/o impact on the rest of the system.
9.1	Rule 9 together with flags indicating malicious activity[a] point towards a reconnaissance attack (e.g., scanning).
10	Flags simultaneously and exclusively in network target features of two netw. devices indicates a problem of their bilateral communication.
10.1	Rule 10 together with flags indicating malicious activities[a] for both devices point towards a MITM attack.
10.2	Rule 10.1 with flags in physical target features indicate a MITM attack which manipulates process relevant data entailing physical impact.
11	Flags in physical target features indicating data disruption point towards disconnection of the network device which sends the data.
11.1	Rule 11 together with flags indicating malicious activities[a] indicates a DoS attack against the disconnected device.
11.2	Rule 11.1 with flags in ph. target features which indicate true physical impact point towards DoS attack interrupting process relevant data.
12	Flags only in network target features of a device X and the ones connected to it indicate disconnection of device X.
12.1	Rule 12 together with flags indicating malicious activities[a] point towards a DoS attack ag. device X.
12.2	Rule 12.1 with flags in ph. target features which indicate true physical impact point towards DoS attack interrupting process relevant data.

[a]Malicious activities are, for example, communication with unknown devices or untypical connection requests from known devices.

e.g. `Unknown event type affecting device X`, to specific event hypotheses, e.g. `DoS attack against device X entailing physical impact Y on component Z`.

Case-specific implementation. Based on the general flag interpretation rules in Table 7, a set of known event signatures is defined for the demonstration case. Fig. 8 visualizes them for selected victim devices, and on a reduced version of the system. A description of the signatures is provided in Table 8. For the sake of clarity, flags in network target features are grouped for each network device according to

$$\tilde{v}_t = \begin{cases} 1\ (anomaly) & \text{if } \exists(v_t^c = 2, 1, -1 \text{ or } -2) \in V_t^{\mathcal{J}} \\ 0\ (no\ anomaly) & \text{otherwise,} \end{cases} \tag{9}$$

where $V_t^{\mathcal{J}}$ is the set of anomaly flags of one device at time t. However, note that the grouped network target features of the victim devices during attacks must include flags indicating malicious activities (e.g., connection to unknown external device) to fulfill the attack signatures. The differentiation between anomaly types is not applicable for the grouped anomaly flags $\tilde{v}_t$ and thus neglected. For the demonstration of CyPhERS on the given study case in Section 5, manual recognition of the defined event signatures is considered.

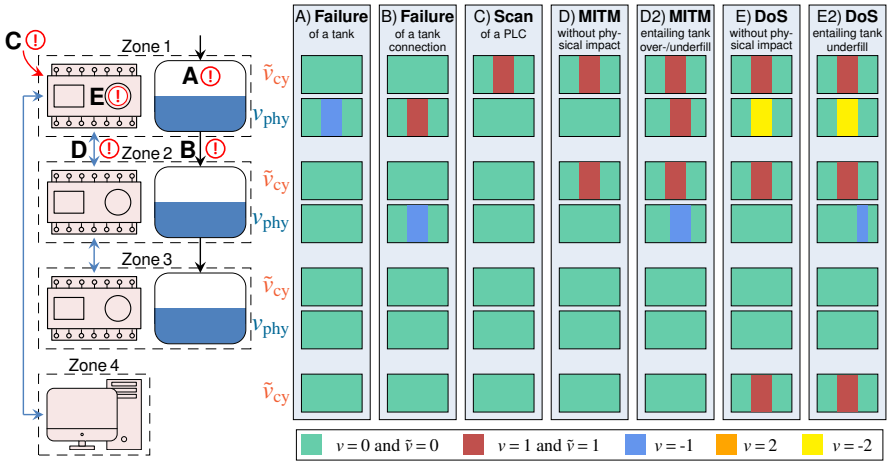


Figure 8: Event signatures for the study case. The signatures are depicted for selected victim devices and physical impacts. Grouped network features $\tilde{v}_{cy}$ of the victim devices include flags indicating malicious activities during attacks.

Table 8: Description of the event signatures depicted in Fig. 8, and applied anomaly flag interpretation rules.

Event signature	Description	Applied rules
A) **Failure** of a tank	Abnormal high/low level of individual tank, no network anomaly. Thus, leak, in- or outflow failure[a].	1,2,7
B) **Failure** of tank connection	Parallel over-/underfill of linked tanks, no netw. anomaly. Thus, pump/valve failure or leak between them[a].	1,2,8
C) **Scan** of a PLC	Individual PLC affected, communication to unknown device with unusual TCP flags, no phy. anomalies.	1,3,9.1
D) **MITM** w/o phy. impact	Simult. & excl. netw. anomalies in two conn. PLCs, both comm. with unknown device, no phy. anomalies.	1,3,10.1
D2) **MITM** w. over-/underfill	Signature D with over-/underfill of tanks controlled by the victim PLCs. Manip. fill levels distract pumps.	1,5,10.2
E) **DoS** w/o phy. impact	Network anomalies only for a device X (e.g., a PLC) and the ones connected to it, physical data communicated by device X disrupted, connection of device X to unknown device, no phy. plausible anomalies.	1,3,11.1,12.1
E2) **DoS** w. tank underfill	Signature E with underfill of a tank controlled by a PLC which receives data from the disc. victim device.	1,5,11.2,12.2

[a]Specific failure type can be concluded from anomaly flag directions and/or actuator type between tanks.

5. Demonstration

In this section, results of applying CyPhERS on the three attack and fault scenarios (S1-S3) of the demonstration case are presented. In preparation of that, Section 5.1 explains how the considered alternative approaches are represented for benchmarking, and Section 5.2 demonstrates attack signatures within ungrouped network target features. Thereafter, S1-S3 are successively evaluated in the Sections 5.3-5.5. In that context, examination of S2 includes comparison with the three benchmarks. The investigated dataset contains some wrong ground truth event lengths and labels as well as further unlabeled anomalies. Thus, focus of this section is on a qualitative demonstration of CyPhERS since a meaningful quantitative assessment is impractical under these circumstances.

5.1. Benchmark concepts

As part of the following demonstration of CyPhERS, a qualitative comparison to the existing event identification concepts introduced in Section 1.1 is conducted (group A-C). Group A is represented by considering only physical target features of CyPhERS, and grouping associated flags to provide system-wide monotypic anomaly flags v_{CPS}. CyPhERS' physical target features without grouping their flags are considered for representing monotypic anomaly flags on feature level (group B). For group C, anomaly flags in both physical and network target features together are grouped, providing cyber-physical system-wide monotypic anomaly flags.

5.2. Attack signatures in ungrouped network target features

The subsequent evaluation of S1-S3 considers grouped network target features for clarity (see Section 4.4). To demonstrate the appearance of the ungrouped flags that CyPhERS' Stage 1 provides during MITM, DoS and scanning attacks, they are depicted in Fig. 9 for selected network devices.

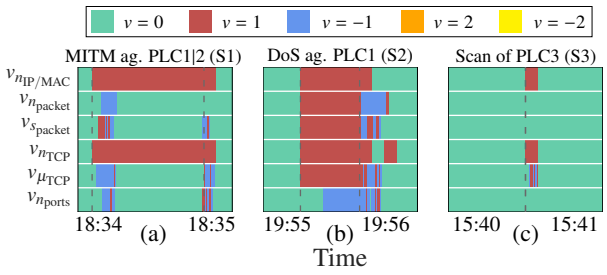


Figure 9: Flags in ungrouped network target features provided by CyPhERS' Stage 1 for a) PLC1 during MITM, b) PLC1 during DoS and c) PLC3 during scanning attack.

During all three attack types malicious activities are flagged, which is a requirement of the pre-defined attack signatures (see Fig. 8). These activities comprise communication with an unknown device ($v_{n_{\text{IP/MAC}}} = 1$) and use of unusual TCP flags ($v_{n_{\text{TCP}}} = 1$). Fig. 9 further indicates that the different attack types also express in distinctive signatures within the ungrouped network features. For example, DoS attacks result in pronounced anomaly flags on all features in contrast to the others. Fig. 10 showcases this on a comparison to MITM attacks. While the DoS attack entails a global anomaly, the MITM attack only results in a local one, primarily in the beginning and end of the attack[3]. While the differences within ungrouped network features are not taken into account in the following demonstration of CyPhERS, they potentially can be integrated to provide further information for distinction of different attack types.

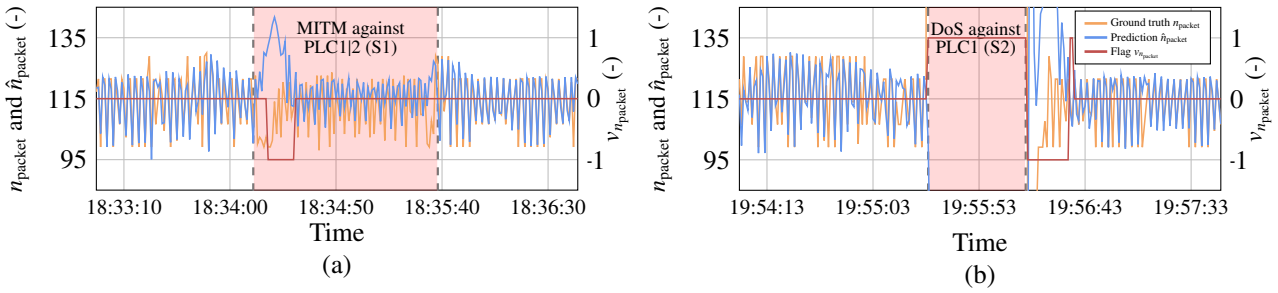


Figure 10: Anomalies in n_{packet} of PLC1 induced by a (a) MITM and (b) DoS attack.

5.3. Evaluation of attack and fault scenario S1

The event signatures provided by CyPhERS' Stage 1 during S1 are depicted in Fig. 11. The anomaly flags of tank fill levels ($v_{H_{\text{T1}}}$-$v_{H_{\text{T8}}}$) are located next to the grouped network feature flags $\tilde{v}$ of the PLC controlling the respective process zone (see Fig. 4). The scenario comprises five MITM attacks and three physical faults, affecting several network devices and physical components.

[3]Note that the detection of the MITM attack-induced local anomaly in Fig. 10 demonstrates the advantage of incorporating temporal information through time series models, as motivated in Section 2.1.3.

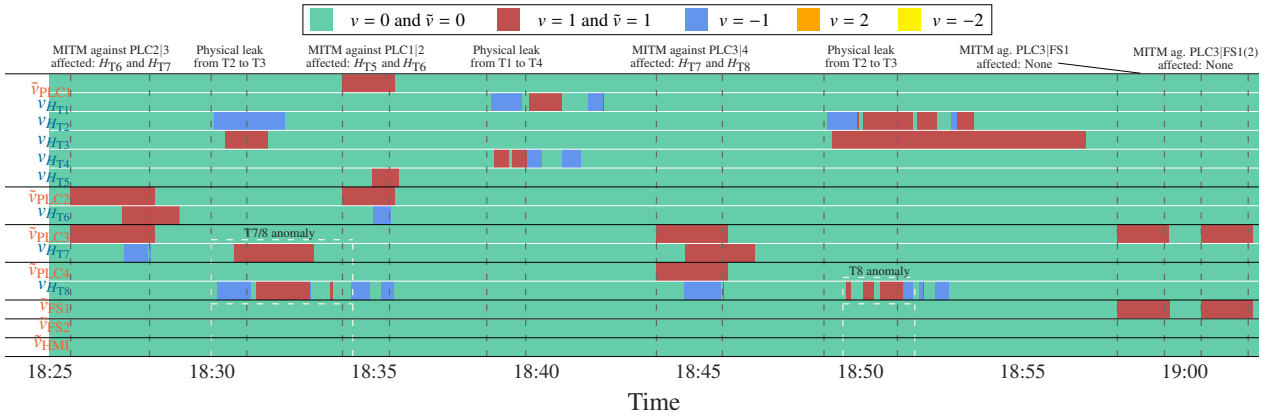


Figure 11: Event signatures of CyPhERS' Stage 1 during scenario S1. Additional anomalies not labeled by [35] are marked using beige boxes, wrong ground truth labels are given in parenthesis.

5.3.1. *MITM attacks*

From Fig. 11 it is visible that anomaly flags during MITM attacks either follow event signature D or D2 (see Fig. 8), which allows to conclude on the attack type, victim devices, attacker location and physical impact[4]. For example, during the third MITM attack (~18:45), signature D2 indicates a MITM attack against PLC3 and PLC4 from an external device entailing overfill of T7 ($v_{H_{T7}} = 1$) and underfill of T8 ($v_{H_{T8}} = -1$) due to failed pump activation as a result of manipulated fill levels exchanged between the victim PLCs. Note that in all cases MITM-induced anomalies are flagged in network features before the physical process is impacted, potentially allowing incident response mechanisms to take timely countermeasures.

The modification of the physical process during the first and third MITM attack is comparatively strong. As a result, the impact also extends over the next process cycle as well as consecutive tanks, which explains the two additional anomalies indicated in Fig. 11. As an example, Fig. 12 depicts the abnormal behavior of T7 in the next process cycle after the first MITM attack. The comparison with the subsequent process cycle clearly indicates that T7 remains filled for an unusually long period.

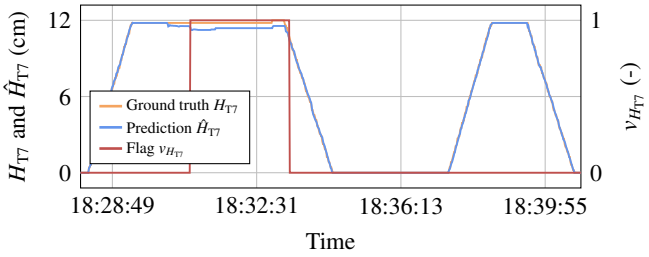


Figure 12: Unlabeled anomaly of T7 during S1 as a result of the MITM attack in the previous process cycle.

5.3.2. *Physical faults*

Anomaly flags during physical faults follow event signature B (see Fig. 8), allowing to localize affected components, and infer fault types and physical impact. For example, during the second fault (~18:40) the provided signature indicates a leak between T1 and T4, resulting in a simultaneous underfill of T1 ($v_{H_{T1}} = -1$) and overfill of T4 ($v_{H_{T4}} = 1$). During the first and third fault, parallel anomalies from the previous MITM attacks complicate recognition of signature B. However, as the flags for T7 and T8 can be explained by the preceding attacks, while a plausible connection to the abnormal behavior of T2 and T3 cannot be derived, these two events can be disentangled.

Fig. 11 indicates that CyPhERS' Stage 1 detects events at an early stage, and reliably differentiates anomaly types in the beginning of events. However, the flags overrun the events for two reasons: Firstly, the CPS does not immediately recover after an attack or fault, thus, anomalies naturally persist. Secondly, the anomaly detection and classification pipelines exhibit a *recovering phase* after detected events. As the detector evaluates the average distance of several consecutive observations, anomalies are flagged for some more steps even though the system behavior is already normal. Moreover, while passing an event, abnormal observations become the new model input, which manipulates the predictions entailing longer anomaly flags and unreliable flag types. A concept improvement tackling this shortcoming is discussed in Section 6.2.

[4]According to [35], the last MITM attack is supposed to affect PLC3 and FS2. However, the anomaly flags point towards FS1 instead of FS2. A look on Fig. 4 shows that PLC3 and FS2 in fact do not communicate, proving successful identification of a wrong label.

5.4. Evaluation of attack and fault scenario S2

In the second scenario, DoS attacks are performed in addition to MITM attacks and physical faults. The event signatures provided by CyPhERS' Stage 1 during S2 are depicted in Fig. 13 together with the ones of the three benchmarks.

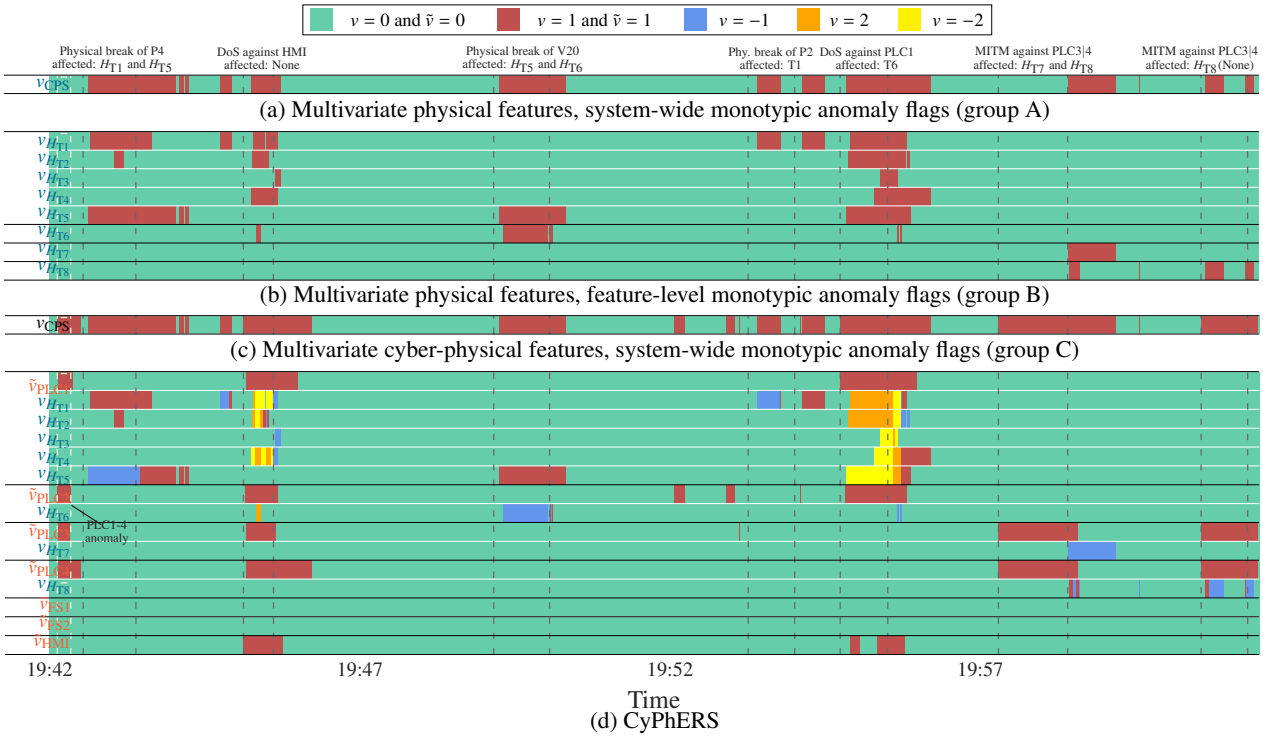


Figure 13: Event signatures of CyPhERS' Stage 1 and the three benchmarks during scenario S2. Additional anomalies not labeled by [35] are marked using beige boxes, wrong ground truth labels are given in parenthesis.

5.4.1. MITM attacks and physical faults

Anomaly flags provided by CyPhERS during faults either follow signature A or B. Thus, the affected tanks are localized, and pump or valve failures together with resulting physical impact inferred. The benchmarks providing system-wide flags (Fig. 13 (a) and (c)) indicate fault-induced anomalies, however, do not provide information on affected components, event types and physical impact. In contrast, flagging anomalies in individual physical features (Fig. 13 (b)) additionally allows for localizing affected tanks. Nevertheless, the lack of network target features and anomaly type differentiation renders identification of event types and physical impact infeasible.

Flags of CyPhERS during the two MITM attacks follow signature D2, allowing to conclude attack type, victims, attacker location, and physical impact. During the last attack, no physical impact should exist according to [35]. However, as can be seen from Fig. 14, T8 in fact exhibits abnormal behavior.

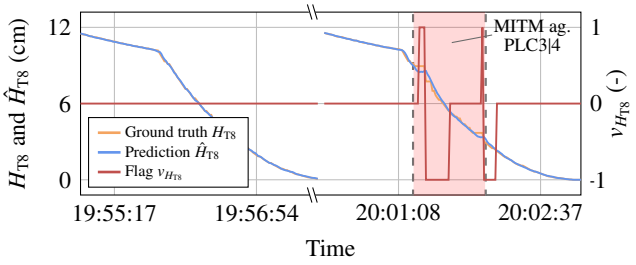


Figure 14: Unlabeled anomaly of H_{T8} in S2 during the MITM attack against PLC3 and PLC4.

The benchmarks of group A and B only detect the physical impact of MITM attacks (see Fig. 13 (a) and (b)). As a result, they exhibit high detection delays during the first one. Moreover, none of the three benchmarks can indicate that anomalies are caused by a cyber attack.

5.4.2. DoS attacks

The flags provided by CyPhERS' Stage 1 during the two DoS attacks correspond to signature E or E2, allowing to conclude on attack type, victim device, attacker location, and physical impacts. For example, in the second case, signature E2 indicates a DoS attack against PLC1 from an external device, resulting in a slight underfill of T6 ($v_{H_{T6}} = -1$) due to interrupted communication of H_{T5} values from PLC1 to PLC2.

The three benchmarks neither indicate disconnection of a network device nor localize it, since they either only detect DoS-induced anomalies in physical process data, or provide non-interpretable system-wide flags. Moreover, as they only output monotypic flags, it cannot be inferred that most anomalies result from data disruption instead of true physical events.

5.4.3. Unlabeled event

CyPhERS' Stage 1 detects an unlabeled event in the beginning of S2, which is depicted in Fig. 15 on the example of PLC2's μ_{TCP}. Although the event signature is not known, fulfillment of flag interpretation rules 1 and 4 (see Table 7) indicates a network failure only affecting PLC1-4 without physical impact. This example demonstrates how CyPhERS provides real-time information including occurrence, affected devices, physical impact, and differentiation between network failure and cyber attack also for unknown event types. The alternative detection strategies either entirely miss this event (see Fig. 13 (a) and (b)) or cannot provide any more information than its occurrence (see Fig. 13 (c)).

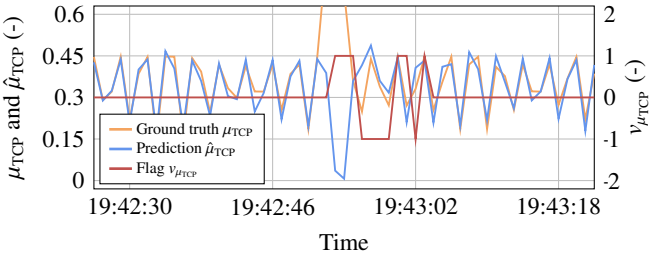


Figure 15: Anomaly in μ_{TCP} of PLC2 induced by an unlabeled event in S2.

5.5. Evaluation of attack and fault scenario S3

The third scenario adds scanning attacks to the previously evaluated attack and fault types. The event signatures provided by CyPhERS' Stage 1 during S3 are depicted in Fig. 16.

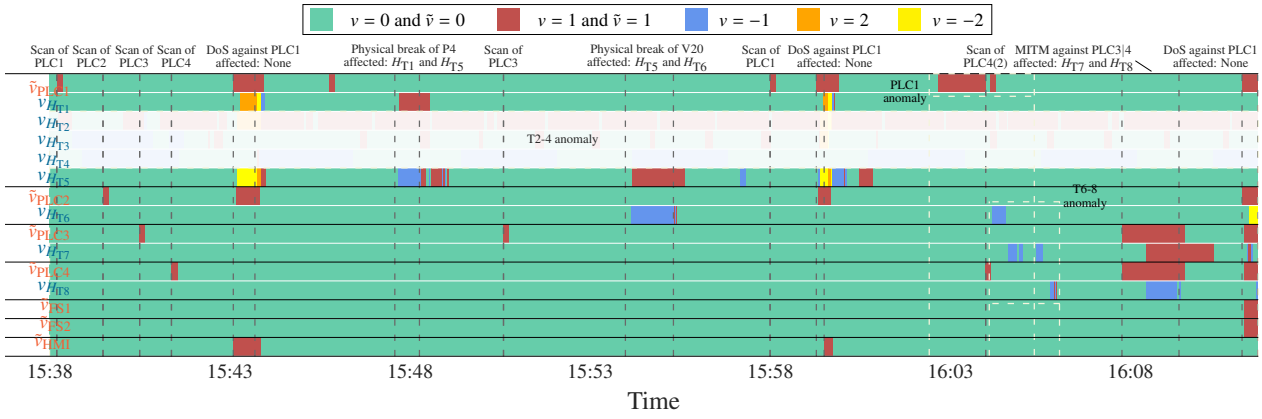


Figure 16: Event signatures of CyPhERS' Stage 1 during scenario S3. Additional anomalies not labeled by [35] are marked using beige boxes, wrong ground truth labels are given in parenthesis.

5.5.1. Unlabeled events

Event 1. S3 is characterized by a fundamental process modification not indicated by [35]. Throughout the entire scenario, anomalies are regularly flagged in $v_{H_{T2}}$, $v_{H_{T3}}$ and $v_{H_{T4}}$, as depicted in Fig. 17 for H_{T2}. While the event signature is not known, fulfillment of flag interpretation rules 1 and 2 (see Table 7) points towards a physical failure only affecting the two sub-strings comprising T2-4 (see Fig. 4). For the sake of clarity, $v_{H_{T2}}$, $v_{H_{T3}}$ and $v_{H_{T4}}$ are faded in Fig. 16.

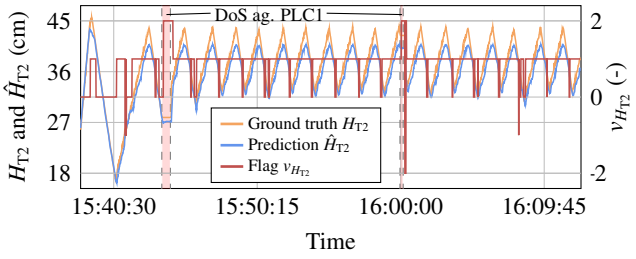


Figure 17: Unlabeled event of H_{T2}, H_{T3} and H_{T4} throughout S3 on the example of H_{T2}.

Event 2. CyPhERS' Stage 1 indicates two further unlabeled anomalies in S3, which likely result from the same event. The anomaly in network traffic of PLC1 is depicted in Fig. 18. Shortly after, a consecutive underfill of H_{T6}, H_{T7} and H_{T8} is indicated. Compliance with flag interpretation rules 1 and 6 suggest a network device failure of PLC1 potentially entailing the underfill of T6-T8 due to interrupted communication of H_{T5} values from PLC1 to PLC2.

These two examples again showcase how CyPhERS provides real-time information for unknown event types including occurrence, affected devices, physical impact, and differentiation between physical failures, network failures, cyber attacks, and cyber-physical attacks.

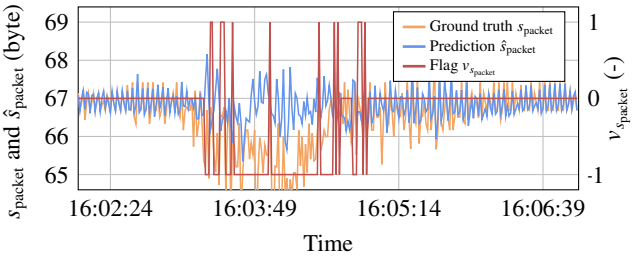


Figure 18: Unlabeled anomaly in s_{packet} of PLC1 in S3.

5.5.2. *Scanning attacks*

Fig. 16 illustrates that anomaly flags during all scanning attacks follow signature C, which gives insights regarding attack type, victim, attacker location, and impact on the physical process[5]. For example, during the first scan (~15:38), signature C, which includes flagging of communication with an unauthorized external device ($v_{n_{IP/MAC}} = 1$) containing unusual TCP flags ($v_{n_{TCP}} = 1$), indicates a scan of PLC1 by an external device not impacting the physical process.

6. Discussion

This section discusses key findings of the concept demonstration in Section 5. In Section 6.1, it is analyzed whether CyPhERS fulfils its intended purpose. Section 6.2 addresses possible concept improvements. Finally, Section 6.3 reflects on the transferability of the demonstration case results to other CPSs.

6.1. *Proof of concept*

The aim of CyPhERS is to provide CPSs operators with relevant information on unknown and known types of attacks and faults for real-time incident response, while being independent of historical event observations. The results in Section 5 proof this thesis. All considered attack and fault types are identified, including localization of victim devices and attacker location as well as determination of the impact on the physical process, through matching anomaly flags provided by CyPhERS' Stage 1 with known event signatures. Moreover, information on further unknown events, which are not officially labeled by the authors of the investigated dataset, are provided, including event occurrence, affected components, physical impact, and differentiation between physical failure, cyber attack, cyber-physical attack and network device failure.

[5] In this context, a wrong label is identified for the last scanning attack.

6.2. Concept improvements

One open issue identified by the concept demonstration is the recovering phase of the anomaly detection and classification pipelines. Primary reason for this is the modification of the ground truth model inputs $x^c_{t-w}, ..., x^c_{t-1}$ in (2) during and through anomalies, which affects the model's capability to predict the normal behavior of a target feature. To solve this issue $x^c_{t-w}, ..., x^c_{t-1}$ in (2) could be replaced with the respective normal behavior predictions $\hat{x}^c_{t-w}, ..., \hat{x}^c_{t-1}$ during flagged anomalies.

Another improvement is seen in automating the process of creating the event signature database. As of now, CyPhERS requires to define signatures by manually applying anomaly flag interpretation rules (see Table 7) on the specific system at hand. In the future, this process should be automated through an application which generates event signatures by providing it with interpretation rules and specifications about the physical and digital components and architectures of a CPS.

6.3. Result transferability to other CPSs

As pointed out in Section 2.3, process volatility and randomness may complicates application of CyPhERS to some CPSs, including power system applications. Since the study case considered in this work exhibits comparatively simple repeating process patterns, a feasibility demonstration of CyPhERS for systems with pronounced volatility and randomness is required. Moreover, to proof applicability for smaller CPSs without dedicated human operator and for more complex processes where manual recognition of signatures would require a high cognitive effort, the automated signature evaluation in Stage 2 needs to be demonstrated.

7. Conclusion and future work

This work introduces CyPhERS, a cyber-physical event reasoning system that provides real-time information about known and unknown types of attacks and faults in CPSs, independent of historical event observations. CyPhERS uses a two-stage process to infer event information, including occurrence, location, root cause, and physical impact. In Stage 1, informative event signatures are created using methods such as cyber-physical data fusion, unsupervised multivariate time series anomaly detection, and anomaly type differentiation. In Stage 2, the event signatures are evaluated either automatically by matching with a signature database of known events or through manual interpretation by the operator. CyPhERS is demonstrated on a cyber-physical water distribution system, where it successfully identifies various attack and fault types, which includes localization of victim devices and attacker location as well as determination of attack or failure type and impact on the physical process. Additionally, CyPhERS provides information on unknown event types such as occurrence, affected components, physical impact, and differentiation between physical failure, cyber attack, and network failure. Future work will focus on demonstrating CyPhERS for systems with pronounced volatility and randomness under consideration of the automated signature evaluation in Stage 2.

Acknowledgement

This work is partly funded by the Innovation Fund Denmark (IFD) under File No. 91363, and by the Helmholtz Association under the program 'Energy System Design'.

References

[1] R. Alguliyev, Y. Imamverdiyev, L. Sukhostat, Cyber-physical systems and their security issues, Computers in Industry 100 (2018) 212–223. doi:https://doi.org/10.1016/j.compind.2018.04.017.

[2] S. Colabianchi, F. Costantino, G. Di Gravio, F. Nonino, R. Patriarca, Discussing resilience in the context of cyber physical systems, Computers & Industrial Engineering 160 (2021) 107534. doi:https://doi.org/10.1016/j.cie.2021.107534.
URL https://www.sciencedirect.com/science/article/pii/S0360835221004381

[3] L. A. Maglaras, K.-H. Kim, H. Janicke, M. A. Ferrag, S. Rallis, P. Fragkou, A. Maglaras, T. J. Cruz, Cyber security of critical infrastructures, ICT Express 4 (1) (2018) 42–45, sI: CI & Smart Grid Cyber Security. doi:https://doi.org/10.1016/j.icte.2018.02.001.

[4] F. Frattini, U. Giordano, V. Conti, Facing cyber-physical security threats by psim-siem integration, in: 2019 15th European Dependable Computing Conference (EDCC), 2019, pp. 83–88. doi:10.1109/EDCC.2019.00026.

[5] N. Müller, C. Ziras, K. Heussen, Assessment of cyber-physical intrusion detection and classification for industrial control systems, in: 2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2022, pp. 432–438. doi:10.1109/SmartGridComm52983.2022.9961010.

[6] A. Ayodeji, Y.-k. Liu, N. Chao, L.-q. Yang, A new perspective towards the development of robust data-driven intrusion detection for industrial control systems, Nuclear Engineering and Technology 52 (12) (2020) 2687–2698.

[7] J. Zhang, L. Pan, Q.-L. Han, C. Chen, S. Wen, Y. Xiang, Deep learning based attack detection for cyber-physical system cybersecurity: A survey, IEEE/CAA Journal of Automatica Sinica 9 (3) (2022) 377–391. doi:10.1109/JAS.2021.1004261.

[8] Y. Luo, Y. Xiao, L. Cheng, G. Peng, D. D. Yao, Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities, ACM Computing Surveys 54 (5) (2021). doi:10.1145/3453155.

[9] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, R. Candell, A survey of physics-based attack detection in cyber-physical systems, ACM Computing Surveys 51 (4) (jul 2018). doi:10.1145/3203245.

[10] X. Cai, Q. Wang, Y. Tang, L. Zhu, Review of cyber-attacks and defense research on cyber physical power system, in: 2019 IEEE Sustainable Power and Energy Conference (iSPEC), 2019, pp. 487–492. doi:10.1109/iSPEC48194.2019.8975131.

[11] B. Lindemann, B. Maschler, N. Sahlab, M. Weyrich, A survey on anomaly detection for technical systems using lstm networks, Computers in Industry 131 (2021) 103498. doi:https://doi.org/10.1016/j.compind.2021.103498.

[12] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, J. Barbosa, Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges, Computers in Industry 123 (2020) 103298. doi:https://doi.org/10.1016/j.compind.2020.103298.
[13] L. Xi, R. Wang, Z. J. Haas, Data-correlation-aware unsupervised deep-learning model for anomaly detection in cyber–physical systems, IEEE Internet of Things Journal 9 (22) (2022) 22410–22421. doi:10.1109/JIOT.2022.3150048.
[14] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S.-K. Ng, Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks, in: I. V. Tetko, V. Kůrková, P. Karpov, F. Theis (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series, Springer International Publishing, Cham, 2019, pp. 703–716.
[15] C. Feng, P. Tian, Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 2858–2867. doi:10.1145/3447548.3467137.
[16] T. T. Huong, T. P. Bac, D. M. Long, T. D. Luong, N. M. Dan, L. A. Quang, L. T. Cong, B. D. Thang, K. P. Tran, Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach, Computers in Industry 132 (2021) 103509. doi:https://doi.org/10.1016/j.compind.2021.103509.
[17] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N. V. Chawla, A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'19/IAAI'19/EAAI'19, AAAI Press, 2019. doi:10.1609/aaai.v33i01.33011409.
[18] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N. V. Chawla, A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data, in: Proceedings of the AAAI conference on artificial intelligence, 2019, pp. 1409–1416. doi:10.1609/aaai.v33i01.33011409.
[19] S. Tuli, G. Casale, N. R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, arXiv preprint arXiv:2201.07284 (2022).
[20] F. Khoshnevisan, Z. Fan, Rsm-gan: A convolutional recurrent gan for anomaly detection in contaminated seasonal multivariate time series, arXiv preprint arXiv:1911.07104 (2019).
[21] D. Hallac, S. Vare, S. Boyd, J. Leskovec, Toeplitz inverse covariance-based clustering of multivariate time series data, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, p. 215–223. doi:10.1145/3097983.3098060.
[22] D. Song, N. Xia, W. Cheng, H. Chen, D. Tao, Deep r -th root of rank supervised joint binary embedding for multivariate time series retrieval, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, p. 2229–2238.
[23] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, p. 387–395. doi:10.1145/3219819.3219845.
[24] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, p. 2828–2837. doi:10.1145/3292500.3330672.
[25] J. M. Navarro, D. Rossi, Hurra! human readable router anomaly detection, in: 2020 32nd International Teletraffic Congress (ITC 32), 2020, pp. 19–28. doi:10.1109/ITC3249928.2020.00011.
[26] X. Niu, J. Li, J. Sun, K. Tomsovic, Dynamic detection of false data injection attack in smart grid using deep learning, in: 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), 2019, pp. 1–6. doi:10.1109/ISGT.2019.8791598.
[27] A. Bezemskij, G. Loukas, R. J. Anthony, D. Gan, Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle, in: 2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS), 2016, pp. 61–68. doi:10.1109/IUCC-CSS.2016.017.
[28] R. Heartfield, G. Loukas, A. Bezemskij, E. Panaousis, Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning, IEEE Transactions on Information Forensics and Security 16 (2021) 1720–1735. doi:10.1109/TIFS.2020.3042049.
[29] A. A. Cook, G. Mısırlı, Z. Fan, Anomaly detection for iot time-series data: A survey, IEEE Internet of Things Journal 7 (7) (2020) 6481–6494. doi:10.1109/JIOT.2019.2958185.
[30] R. R. R. Barbosa, R. Sadre, A. Pras, Towards periodicity based anomaly detection in scada networks, in: Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), 2012, pp. 1–4. doi:10.1109/ETFA.2012.6489745.
[31] R.-J. Hsieh, J. Chou, C.-H. Ho, Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing, in: 2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA), 2019, pp. 90–97. doi:10.1109/SOCA.2019.00021.
[32] S. Kang, S. Sristi, J. Karachiwala, Y.-C. Hu, Detection of anomaly in train speed for intelligent railway systems, in: 2018 International Conference on Control, Automation and Diagnosis (ICCAD), 2018, pp. 1–6. doi:10.1109/CADIAG.2018.8751374.
[33] A. A. Abokifa, K. Haddad, C. Lo, P. Biswas, Real-time identification of cyber-physical attacks on water distribution systems via machine learning–based anomaly detection techniques, Journal of Water Resources Planning and Management 145 (1) (2019) 04018089.
[34] J. Yu, Y. Song, D. Tang, D. Han, J. Dai, Telemetry data-based spacecraft anomaly detection with spatial–temporal generative adversarial networks, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–9. doi:10.1109/TIM.2021.3073442.
[35] L. Faramondi, F. Flammini, S. Guarino, R. Setola, A hardware-in-the-loop water distribution testbed dataset for cyber-physical security testing, IEEE Access 9 (2021) 122385–122396. doi:10.1109/ACCESS.2021.3109465.
[36] M. Conti, N. Dragoni, V. Lesyk, A survey of man in the middle attacks, IEEE Communications Surveys Tutorials 18 (3) (2016) 2027–2051. doi:10.1109/COMST.2016.2548426.
[37] T. Mahjabin, Y. Xiao, G. Sun, W. Jiang, A survey of distributed denial-of-service attack, prevention, and mitigation techniques, International Journal of Distributed Sensor Networks 13 (12) (2017). doi:10.1177/1550147717741463.
[38] E. Bou-Harb, M. Debbabi, C. Assi, Cyber scanning: A comprehensive survey, IEEE Communications Surveys Tutorials 16 (3) (2014) 1496–1519. doi:10.1109/SURV.2013.102913.00020.
[39] M. K. Hasan, A. A. Habib, Z. Shukur, F. Ibrahim, S. Islam, M. A. Razzaque, Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations, Journal of Network and Computer Applications 209 (2023). doi:https://doi.org/10.1016/j.jnca.2022.103540.
[40] L. Cao, X. Jiang, Y. Zhao, S. Wang, D. You, X. Xu, A survey of network attacks on cyber-physical systems, IEEE Access 8 (2020) 44219–44227. doi:10.1109/ACCESS.2020.2977423.
[41] F. Li, X. Yan, Y. Xie, Z. Sang, X. Yuan, A review of cyber-attack methods in cyber-physical power system, in: 2019 IEEE 8th International Conference on Advanced Power System Automation and Protection (APAP), 2019, pp. 1335–1339. doi:10.1109/APAP47170.2019.9225126.
[42] J.-P. A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, M. Malli, Cyber-physical systems security: Limitations, issues and future trends, Microprocessors and Microsystems 77 (2020) 103201. doi:https://doi.org/10.1016/j.micpro.2020.103201.
[43] D. Chakraborty, H. Elzarka, Advanced machine learning techniques for building performance simulation: a comparative analysis, Journal of

Building Performance Simulation 12 (2) (2019) 193–207. doi:10.1080/19401493.2018.1498538.
[44] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
[45] E. Scornet, G. Biau, J.-P. Vert, Consistency of random forests, The Annals of Statistics 43 (4) (2015) 1716 – 1741. doi:10.1214/15-AOS1321.
[46] C. S. Bojer, J. P. Meldgaard, Kaggle forecasting competitions: An overlooked learning opportunity, International Journal of Forecasting 37 (2) (2021) 587–603. doi:https://doi.org/10.1016/j.ijforecast.2020.07.007.
[47] T. Hastie, R. Tibshirani, J. H. Friedman, J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer, 2009.
[48] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.
[49] S. Siami-Namini, N. Tavakoli, A. Siami Namin, A comparison of arima and lstm in forecasting time series, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1394–1401. doi:10.1109/ICMLA.2018.00227.
[50] D. M. Q. Nelson, A. C. M. Pereira, R. A. de Oliveira, Stock market's price movement prediction with lstm neural networks, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1419–1426. doi:10.1109/IJCNN.2017.7966019.
[51] S. Srivastava, S. Lessmann, A comparative study of lstm neural networks in forecasting day-ahead global horizontal irradiance with satellite data, Solar Energy 162 (2018) 232–247. doi:10.1016/j.solener.2018.01.005.
[52] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[53] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. V. Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, M. Dumonal, J. Kościsz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, G. Grosch, Darts: User-friendly modern machine learning for time series (2021). arXiv:2110.03224.