Anomaly Detection of ICS Communication Using Statistical Models

Ivana Burgetová
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
burgetova@fit.vutbr.cz

Petr Matoušek
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
matousp@fit.vutbr.cz

Ondřej Ryšavý
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
rysavy@fit.vutbr.cz

***Abstract*—Industrial Control System (ICS) transmits control and monitoring data between devices in an industrial environment that includes smart grids, water and gas distribution, or traffic control. Unlike traditional internet communication, ICS traffic is stable, periodical, and with regular communication patterns that can be described using statistical modeling. By observing selected features of ICS transmission, e.g., packet direction and inter-arrival times, we can create a statistical profile of the communication based on distribution of features learned from the normal ICS traffic. This paper demonstrates that using statistical modeling, we can detect various anomalies caused by irregular transmissions, device or link failures, and also cyber attacks like packet injection, scanning, or denial of service (DoS). The paper shows how a statistical model is automatically created from a training dataset. We present two types of statistical profiles: the master-oriented profile for one-to-many communication and the peer-to-peer profile that describes traffic between two ICS devices. The proposed approach is fast and easy to implement as a part of an intrusion detection system (IDS) or an anomaly detection (AD) module. The proof-of-concept is demonstrated on two industrial protocols: IEC 60870-5-104 (aka IEC 104) and IEC 61850 (Goose).**

***Index Terms*—anomaly detection, communication patterns, industrial networks, IEC 104, monitoring, smart grid**

I. Introduction

Security and safety of critical infrastructure that includes substations, power plants, water and gas treatment facilities, or traffic control systems become more and more important due to the rising level of automation and intelligent control of industrial processes as proposed by the Industry 4.0 initiative [1]. Deployment of automation in industry induces a higher risk of failures caused by device malfunctioning, lost packets, communication delays, and also cyber attacks initiated from infected machines [2]. To mitigate and prevent internal threats, we need to monitor industrial communication and observe irregularities or suspicious patterns that occur in the traffic. How important this task is can be seen in cyber attacks against Ukrainian power plants in 2016 [3] or more recently on the ransomware attack against the Colonial Gas Pipeline in the U.S. that happened in May 2021 [4]. Both attacks were initiated from infected internal stations.

Industrial systems are well protected against external threats using firewalls and IDS systems that filter communication between ICS network and the internet. Thus, direct attacks against ICS infrastructure are rare. However, an attacker can gain an access to the system through a malware sent to a user via infected e-mail attachment. To detect malware activity, we need to observe internal ICS traffic and trace unusual behavior.

In this work, we study statistical properties of ICS traffic. Previous research demonstrated that Industrial Control System (ICS) and Supervisory Control and Data Acquisition (SCADA) communication is more stable than traditional internet network traffic in the sense that the number of communicating nodes does not change too often [5], [6], end devices use a limited number of communication protocols, and the frequency of exchanged packets is predictable [7]. The regular behavior of ICS communication is an important condition for generating a stable statistical model that represents a distribution of packet features like size, direction, inter-arrival time, etc. By observing transmitted packets, we compare their distribution with the learned models and raise an alarm if the behavior significantly differs from the learned model. Unlike our previous work [8] where we modeled ICS command sequences using probabilistic automata, in this work we focus on timing properties of ICS communication.

Statistical properties of network communication can be observed on different layers of the TCP/IP model. On the IP layer, we can observe timestamps, packet size, direction, delay, etc. On the transport layer, we can monitor flow duration, segment size and inter-arrival time, round-trip time, or retransmissions [6], [9]. Statistically interesting features can be also extracted from the application layer, e.g., distribution of IEC 104 commands or Modbus operations [10]. The main advantage of the statistical approach is that it does not require a deep packet analysis with high demands on processing, so it can be applied for real-time detection. On the other hand, a statistical model is sensitive to outliers which are particular data with exceptionally low probability that may be incorrectly marked as anomalies. The model assumes that underlying data have a particular distribution that is stable over time.

Stability and regularity of ICS traffic was observed for Modbus [11], IEC 104 [7], or DNP3 [6]. In this paper, we closely examine statistical distribution of IEC 104 and Goose traffic and show how it can be used for anomaly detection.

A. Contribution

The paper presents a technique for statistical modeling of ICS communication. We split ICS communication into regions based on packet inter-arrival times and direction. For each region we create a statistical profile that represents ICS communication. The main point of the method is to determine split-points that divide the ICS traffic into regions for accurate modeling of packet distribution. For detection, we employ the three-sigma rule that gives minimum false positives. The proposed technique successfully detects common anomalies like connection loss, injection attack, rogue device, scanning attack, or denial of service. In addition, our technique not only detects an anomaly but is able to identify its type.

B. Structure of the Paper

The paper is structured as follows. Section II presents recent works in the area of statistical anomaly detection in industrial networks. Section III reviews the basics of the statistical approach and describes the main features of IEC 104 and GOOSE protocols later used in our experiments. Section IV explains the core of our method that is how how split-points for inter-arrival time distributions are automatically computed and how statistical models are created. Section V presents results of our experiments. The last section concludes the work.

II. State of the Art

Statistical properties of ICS communication were explored by Barbosa, et al. in [5], [12] where the authors observed periodicity, throughput, and topology changes. Their results show that SCADA communication exhibits periodic behavior at a smaller scale, it has constant throughput over a long period of time, and keeps a stable number of connections. The periodicity is caused by a polling mechanism used to retrieve data from SCADA slaves [13]. The authors demonstrated that attacks like scanning, denial of service, network protocol manipulation, or buffer overflow disturb the traffic periodicity and can be detected. For modeling the SCADA communication, Barbose et al. represented the number of packets belonging to a specific flow by time series. During periodicity learning, they generated a periodogram for each flow by Fast Fourier Transform. For detection, they employed discrete-time Short-Time Fourier Transform that created a spectrogram where they detected changes. Our approach comes out of Barbosa's observations. Instead of monitoring the number of transmitted packets we provide a more subtle modeling using arrival times distribution which is precise and faster in computation.

Valdes and Cheung [11] presented pattern-based and flow-based anomaly detection of ICS communication. Their patterns included source and destination IP addresses and ports. During detection, they observed previous n-occurrences of the pattern and computed the historical probability of the pattern. If the probability was less than the threshold, an alert was generated. Their solution included periodic pattern updates and pruning of rare patterns. The second technique presented by Valdes and Cheung used flow records. Flow records included more attributes: IP addresses, time of the last packet, the average number of bytes per packet, the variance of bytes per packet, mean and variance of packet inter-arrival time. Similarly to pattern-based detection, they compared the traffic with historical records and computed difference. If a record did not exist or differed too much, an alert was raised. They tested their approach on a MODBUS network with periodic data retrieval. They were able to detect anomalies like scanning, modified data, DoS attack, and system degradation. Unfortunately, their paper does not give the number of false positives or implementation details. Our approach does not observe individual flows but creates a model for entire communication between ICS nodes over a large period of time.

Lin and Nadjm-Tehrani [7] observed timing patterns of spontaneous events in IEC 104 communication that are asynchronously generated by a Remote Terminate Unit (RTU). The authors modelled inter-arrival times using Probabilistic Suffix Trees (PSTs) and analyzed phase transitions, predictability, and frequent patterns. They describe inter-arrival times as symbolic sequences that are further smoothed and used to create a PST. Having the PST, they observe phase transitions, i.e., a period of time during which the distribution of inter-arrival times is stable. They define five groups of traffic patterns: strongly cyclic, weakly cyclic, stable, bursty, and transitional communication. Using the probability of communication patterns, they also predict future behavior, i.e., an ability to state that a certain pattern appears in the next segment. Their approach is, however, computationally very intensive. We also deal with IEC 104 communication, but we do not restrict to spontaneous events only and our computation requirements are low comparing to PST.

In their other work, Lin et al. [14] proposes a timing-based anomaly detection system for SCADA networks where they employ inter-arrival time of packets similarly to our approach. They build a statistical model for selected packets of three ICS protocols: request and responses of S7, Modbus, and IEC 104 spontaneous events. Their model includes sampling distribution defined by the sample mean, standard deviation, and Central Limit Theorem. For detection, they use a sliding window where they calculate the sample mean and sample range. They verified the model on normal traffic and on various attacks including flooding, injection, and prediction (spoofing). They reached a 99% detection rate with 1.4% false positives. Unlike this approach, our statistical model divides packets into several regions based on inter-arrival time and direction, which produces more accurate model for anomaly detection.

III. Preliminaries

A. Statistical Models

Statistical anomaly detection is grounded on the assumption that *normal data instances occur in high probability regions of the stochastic model, while anomalies occur in the low probability regions* [15]. Statistical modeling is a popular technique for anomaly detection because statistical methods allow simple and fast outlier detection, especially in one-dimensional space. They assume data points to be spread out according to some distribution, e.g., normal distribution.

Then, a statistical test is performed in order to determine if a particular data point belongs to the model or not. If the probability of a particular data point generated from the learned model is low, then the data point is declared as an anomaly. For a given dataset is possible to define an interval of normal values using the statistical model and applied test.

Based on the previous research [16], [17] and our own experiments we approximate inter-arrival times of ICS communication and the number of transmitted packets within a time window with normal distribution.

A simple outlier detection technique called *three sigma rule* says that for normal distribution roughly 99.7% of data points lie within the interval $\langle m - 3 * \sigma, m + 3 * \sigma \rangle$, where m is the mean and σ is a standard deviation [18]. Another useful technique is the *box plot rule* [19] that defines an interval of normal values using the Inter Quartile Range (IQR). Having the lower quartile Q_1 and upper quartile Q_3 of the distribution, the normal values are within the interval $\langle Q_1 - 1.5 * IQR, Q_3 + 1.5 * IQR \rangle$, where $IQR = Q_3 - Q_1$. For normal distribution, roughly 99.3% of data points lie within this interval.

We tested both approaches [20]. Because the IQR produces more false positives we focused on the three sigma rule.

B. ICS Communication

This section briefly overviews IEC 104 and GOOSE protocols, defines the inter-arrival time and introduces our datasets.

1) IEC 104 [21]: The protocol transmits data in the monitor direction (from the controlled station) and in the control direction (from the controlling station) in the power grid. Data are transmitted either over the link layer (IEC 101) or TCP/IP (IEC 104). IEC 104 communication includes data acquisition that cyclically collects data from controlling stations, interrogation, command transmission, etc.

For statistical modeling, we observe all IEC 104 packets. The monitoring probe collects their inter-arrival times in each direction. They are later used for creating a statistical model (learning phase) and anomaly detection (testing phase).

2) GOOSE [22]: It is an Ethernet-based protocol used for Intelligent Electronic Devices (IED) that transfers time-critical events in substations. The communication model is based on autonomous decentralization where substation events are transported through multicast or broadcast services. GOOSE uses a publish-subscribe communication model where the publisher writes the values into a local buffer at the sending side and the subscribe reads data from a local buffer on the receiving side. GOOSE messages are regularly sent as keep-alives with sending time locally configured. If there are no changes on the publisher side, packets are almost identical. Statistical model aggregates GOOSE packets based on the destination multicast address.

3) Packet inter-arrival time: Packet inter-arrival time Δt is *the amount of time between the arrival of two subsequent packets*. It is computed by a monitoring probe as a difference between timestamps of these two packets. Its value depends on the location of the probe in the network, see Fig. 1, but the distribution stays the same regardless of a probe location.

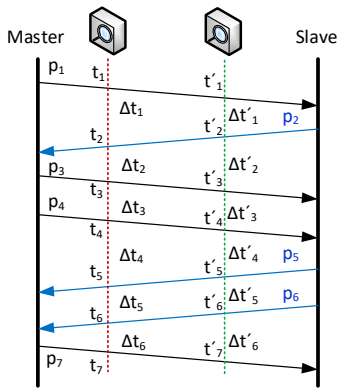


Fig. 1. Measuring inter-arrival times

In case of industrial communication, we can model the inter-arrival time distribution for one direction or for bi-directional traffic. This depends on the underlying ICS protocols. Bi-directional distribution makes sense for IEC 104 master-slave communication while the one-directional distribution model better fits GOOSE publish-subscribe mechanism.

4) Datasets: For our experiments we used several datasets with IEC 104 and GOOSE traffic, see Table I. The first four datasets were created at our university[1], datasets RTU and RICS are from Linköping University, Sweden. GOOSE communication was captured at GIGS Lab in Grenoble, FR.

The table contains the name of the dataset, number of captured packets, number of packets of interest, duration of capturing and the number of communicating ICS devices.

TABLE I
DATASETS WITH IEC 104 (I) AND GOOSE (G) TRAFFIC.

Dataset	Packets		Duration	Dev.
	Total	**IEC/Goose**		
10122018 (I)	102.971	62,676	4h 53 min	4
13122018 (I)	1,433,083	874,697	2 days 23h	14
14-12-18 (I)	35,905	14,342	15h 38min	2
17-12-18 (I)	150,273	58,929	2 days 20h	2
RTU8 (I)	5,788,789	3,117,663	6 days 18h	2
RTU11 (I)	3,491,020	1,828,733	6 days 18h	2
RICS (I)	4,477,807	882,957	12 days 21h	2
Goose (G)	200,583	83,966	19h 26 min	4

IV. STATISTICAL MODELING OF ICS TRAFFIC

In this section, we describe how to build a statistical model of ICS communication using inter-arrival packet times and packet direction. First, we discuss two types of ICS traffic profiles that can be observed based on the underlying traffic. Then we show how to automatically select split-points that divide Δt times of a given communication into regions where Δt values are stable. Lastly, we describe the process of computing the statistical model using the training data.

A. Network Traffic Profiles

Industrial communication is usually limited to a fixed number of communicating nodes connected to the ICS network.

[1] Available at https://github.com/matousp/datasets/scada-iec104 [May 2021].

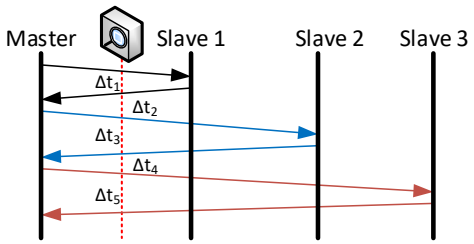


Fig. 2. Master-oriented traffic profiles.

In the master-slave communication model, a master station communicates with one or more slaves, which is typical for protocols like IEC 104, MMS, or Modbus. In the publish-subscribe scheme, the publisher sends data to a set of receiving stations. This model is implemented by GOOSE. Based on the behavior, we define two kinds of profiles for statistical modeling: *master-oriented* and *peer-to-peer-oriented*. Master-oriented profile, see Fig. 2, corresponds to a master that periodically communicates with a set of slaves. In this case, we model all transmissions identified by master's ID (e.g., IP or MAC address). The statistical model represents inter-arrival times of all packets sent or received by the master node.

A peer-to-peer-oriented profile represents communication between two nodes, e.g., IEC 104 master and slave, or GOOSE publisher, see Fig. 1. In this case, peer-to-peer traffic is selected using peer IDs (a pair of MAC/IP addresses + ports) or a multicast destination MAC address (in case of GOOSE). The statistical model is created for each pair of communicating devices. The number of pairs is determined during the learning phase and is stable until a new device connects to the network.

B. Finding Split-Points

As mentioned in the previous research [14], [17], [23], packet inter-arrival time Δt is a useful feature for describing network behavior. In our work, we observe the number of transmitted packets within a time window. For accurate modeling, we add Δt distribution to the statistical model so that we split observed packets into several regions based on Δt values. Then we create a statistical model for each region. As shown later, such detailed modeling detects common anomalies and identifies a particular anomaly that occurred in the network. This is important for network administration. Here, we present three possible ways how to determine split-points for the statistical model.

a) Four equal regions: The first case represents a naive solution that splits the range of inter-arrival times into four equal regions based on the maximal and minimal values observed in the training dataset. Unsurprisingly, this solution is not much suitable for ICS traffic modeling, since Δt values are usually not uniformly distributed as seen in Table II. The table shows inter-arrival time distribution of packets in direction *from master* (fm) and *to master* (tm). This naive solution leads to a situation where majority of observed Δt values fall into the first region, see Fig. 3 (b) and Tab. III.

b) The pre-defined split-points: The second approach aims to give a simple recommendation how to determine suitable split-points for each dataset and direction. Table II shows significant differences in inter-arrival times distribution for individual datasets and directions. It is obvious that reasonable split-points cannot be defined ad-hoc or with some knowledge obtained from the other datasets. Suitable split-points require the analysis of inter-arrival times in the given learning dataset. Due to this fact we reduced the number of split-points and intervals of Δt to simplify this task. We search for one split-point for each direction that provides two additional characteristics of the traffic. With inter-arrival time distribution it seems reasonable to choose some percentile (e.g., median) as a recommended split-point value. However, across our datasets there is no single percentile which would give the best split-point selection for each dataset and direction. The choice that is suitable for one dataset and direction leads to a less stable characteristics for another dataset or direction.

TABLE II
INTER-ARRIVAL TIME DISTRIBUTION IN SELECTED DATASETS.

Dataset	Dir.	min	Q_1	Q_2	Q_3	max
13122018	fm	0.0000	0.0000	0.0003	0.0004	16.1905
	tm	0.0000	0.0002	0.0004	0.0600	10.1331
17-12-18	fm	0.0000	1.9989	3.5909	5.6002	19.9873
	tm	0.0001	1.0091	3.0332	6.0831	19.2696
RTU11	fm	0.0000	0.2109	0.3734	0.4792	2.4896
	tm	0.0000	0.0060	0.0121	0.0145	1.4055
RICS	fm	0.0000	0.0464	0.0830	3.8960	20.0577
	tm	0.0000	0.0073	0.0124	0.1410	10.1876

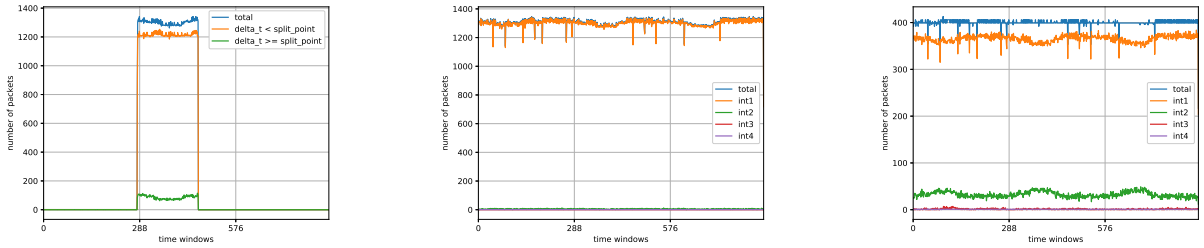
TABLE III
DIFFERENT SPLIT-POINTS FOR DATASET 13122018.

Equal regions				
fm	$\langle 0; 4.05)$ 1300.5	$\langle 4.05; 8.1)$ 5.2	$\langle 8.1; 12.14)$ 0	$\Delta t \geq 12.14$ 1.0
tm	$\langle 0; 2.53)$ 365.1	$\langle 2.53; 5.07)$ 32.5	$\langle 5.07; 7.6)$ 1.2	$\Delta t \geq 7.6.$ 1.0
Auto split-points				
fm	$\Delta t < 0.09$ 1217.8	$\Delta t \geq 0.09$ 89.6	- -	- -
tm	$\Delta t < 0.46$ 347.2	$\Delta t \geq 0.46$ 52.8	- -	- -

TABLE IV
SPLIT-POINT SELECTION IN DATASET 17-12-18 (*from master*).

Split-point		$\Delta t < split - point$		$\Delta t \geq split - point$	
Δt distr.	**value**	**mean**	**std**	**mean**	**std**
Q1	2.00	12.66	3.80	36.82	8.08
Q2	3.60	25.29	8.34	24.19	3.94
mean	4.13	28.89	9.38	20.59	3.33
Q3	**5.66**	37.36	10.99	12.12	**2.83**

c) Split-points automatically derived from the learning dataset: The third case employs the automated method that finds suitable split-points for individual directions of the given dataset. This approach utilizes distribution of Δt times of packets transmitted in the given direction together with the standard deviation. We search for such split-points that filter out the periodic behavior from at least one characteristic. Such split-points are suitable for anomaly detection as they produce stable characteristics, see Fig. 4. Instead of testing the value of each percentile of inter-arrival times as a candidate split-point, we approximate the inter-arrival times distribution with four



(a) One pair of devices **(b)** Direction *from master*. **(c)** Direction *to master*.

Fig. 3. The number of packets transmitted in five minute windows in `13122018` dataset. (a) Communication of one pair of devices (additional characteristic are produced with automatically selected split-point). (b,c) Master-oriented profile, where additional characteristics show the effect of using four equally large intervals of inter-arrival time.

values: quartiles Q1, Q2, Q3 and the mean. We search for the best split-point among these four candidates. Tables IV and V show the candidate split-points for dataset `17-12-18` with *mean* and *standard deviation* of the resulting characteristics.

TABLE V
SPLIT-POINT SELECTION IN DATASET `17-12-18` (*to master*).

Split-point		$\Delta t < split-point$		$\Delta t \geq split-point$	
Δt distr.	**value**	**mean**	**std**	**mean**	**std**
Q1	**1,01**	6,04	2,33	16,78	**2,23**
Q2	3,03	10,92	3,13	11,90	2,86
mean	4,29	14,61	3,34	8,21	3,01
Q3	6,08	16,90	3,37	5,93	3,02

For *from master* direction, see Tab. IV, the value 5.66 (Q3) produces a characteristic with the minimal standard deviation, so it is the best candidate for the split-point. On the other hand, for *to master* direction, see Tab. V, we obtain the most stable characteristic for value 1.01 (Q1).

For some datasets the split-point with minimal deviation divides packets in such way, that the stable characteristic contains only few packets in each time window. If the mean is close to zero, then the testing three sigma rule range would go to negative values and such model would not be capable to detect many types of anomalies. For this reason we put an additional condition for selecting the split-point: we search for (i) minimal standard deviation of the distribution, and (ii) non-zero condition for the sigma rule, i.e., $mean - 3 * \sigma > 0$.

C. Modeling process

In this part, we provide a step-by-step description of the ICS modeling process. Consider an input packet sequence in master-oriented or peer-to-peer-oriented communication model. We build the statistical profile of the traffic as follows:

1) Determine $\Delta t_{i+1} = t_{i+1} - t_i$ for each incoming packet.
2) Partition Δt times by direction to sets ΔT^f and ΔT^t.
3) For each direction $d = \{t, f\}$:
   a) Select the best split-point sp^d.
   b) Gather the characteristics for each time window: the total number of transmitted packets, the number of packets with $\Delta t < sp^d$ (lower region) and the number of packet with $\Delta t \geq sp^d$ (upper region).
   c) Determine the mean and standard deviation for each characteristic.
   d) Find out the range of normal values for each characteristic using the three sigma rule.
   e) Statistical profile P^d is $(sp^d, \langle a_1, a_2 \rangle, \langle l_1, l_2 \rangle, \langle u_1, u_2 \rangle)$ where sp^d is the best split-point. Range $\langle a_1, a_2 \rangle$ denotes the total number of transmitted packet in the time window, $\langle l_1, l_2 \rangle$ the number of packets in lower region and $\langle u_1, u_2 \rangle$ the number of packets in upper region.

We select the best split-point for each set ΔT^d as follows:

1) Compute candidate split points: Q1, Q2, Q3 and the mean of the ΔT^d distribution.
2) For each candidate:
   a) Gather the characteristics for each time window.
   b) Determine the mean and standard deviation.
   c) The best split-point becomes a candidate with minimum σ that satisfies condition $m - 3 * \sigma > 0$.

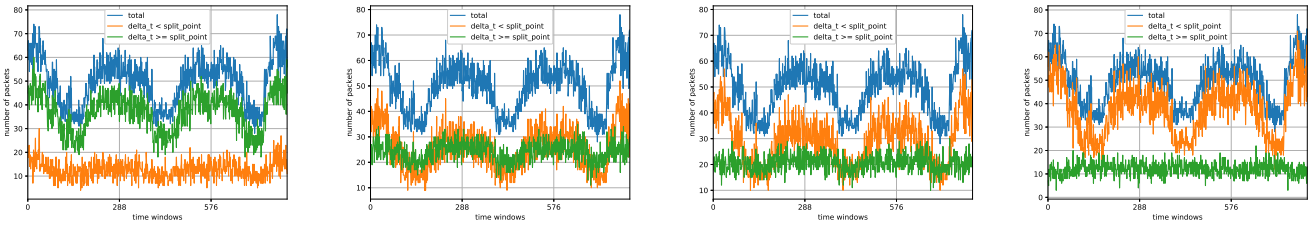
Computational complexity of creating profiles depends on sorting ΔT^d set in order to compute quartiles. Based on the sorting method, the complexity is $\mathcal{O}(n.log\ n)$ or $\mathcal{O}(n^2)$. An example of best-split points selection for dataset *17-12-18* is in Tables IV and V.

V. ANOMALY DETECTION

We tested the stability of our model and detection against common anomalies (failures, attacks) using datasets described in Section III-B4. We employed two detection methods: simple-detection and 3-value-detection. The simple-detection method evaluates each time window independently and compares the number of transmitted packets in this window with the values defined by the statistical profile. An anomaly is detected if a value does not fit the specified range. The 3-value-detection method evaluates three consecutive time windows. This method reduces false positives that occur due to outliers in the dataset. 3-value-detection reports an anomaly only if it is included in at least two of three subsequent time windows.

A. Evaluation of Statistical Profiles

We evaluated our method on datasets described in Section III-B4. We used two-thirds of each dataset to build the profile. The rest of the data was used for testing. Table VI shows the number of false positives windows for each dataset and the accuracy for both methods. It can be seen that the 3-value-detection method provides better accuracy for all datasets.



(a) Split-point Q1=2.00 **(b)** Split-point Q2=3.60 **(c)** Split-point mean=4.13 **(d)** Split-point Q3=5.66

Fig. 4. The number of packets transmitted in five minute windows in `17-12-18` dataset, additional characteristics show the effect of using different split-points (direction *from master*).

TABLE VI
VALIDATION - SUMMARY RESULTS.

Dataset	simple-detection FP/all	simple-detection Acc	3-value-detection FP/all	3-value-detection Acc
10122018	1/20	95%	0/20	100%
13122018	2/285	99.30%	0/285	100%
14-12-18	0/63	100%	0/63	100%
17-12-18	4/273	98.53%	0/273	100%
RTU8	9/650	98.62%	5/650	99.23%
RTU11	16/650	97.54%	0/650	100%
RICS	37/1240	97.02%	11/1240	99.11%
Goose	4/78	94.87%	0/78	100%

TABLE VII
COMPARISON OF THE ANOMALY DETECTION BASED ON STATISTICAL PROFILES AND PROBABILISTIC AUTOMATA.

Attack	Statistical AD	Probabilistic AD		
		$Single$	$Distr_{PT}$	$Distr_{Al}$
Connection loss	✓	×	✓	✓
Injection attack	✓/×	✓	✓	✓
DoS attack	✓	×	×	×
Roque Device	✓	✓	✓	✓
Scanning attack	✓	✓	✓	✓
Switching attack	✓	✓	✓	✓

B. Anomaly Detection Using Statistical Profiles

For anomaly detection, we employed 3-value-detection which is more precise. The statistical profile was created using the whole `17-12-18` dataset with normal traffic. Then we applied the profile on the datasets with anomalies.

Fig. 5 shows, how DoS, scanning and switching attacks changed the characteristics of the traffic (see highlighted parts of the graph with arrows).

Table VII summarizes capability of the proposed method to detect the individual attacks and compares the results with our previous research using probabilistic automata [24]. The table shows that while the DoS attack was not properly detected by probabilistic automata, it can be detected by statistical profiles. On the contrary, statistical approach cannot detect an injection attack, which is covered by the probabilistic approach.

The proposed method of statistical profiles based on inter-arrival times and direction successfully detects common types of ICS attacks. It did not correctly recognize only the first injection attack that did not involve a sufficient number of packets which would significantly differed from the profile.

The results also show that it is useful to build profiles for each direction because some attacks that could be easily hidden in the overall traffic, e.g., the second scanning attack in Fig. 5 (b, e) and the switching attack in Fig. 5 (f). Fig. 5 (a) also shows the benefit of using split-points. In this case, the DoS attack is not detected by the main characteristic (the total number of packets, blue line) but by additional characteristic ($\Delta t \geq split - point$, green line).

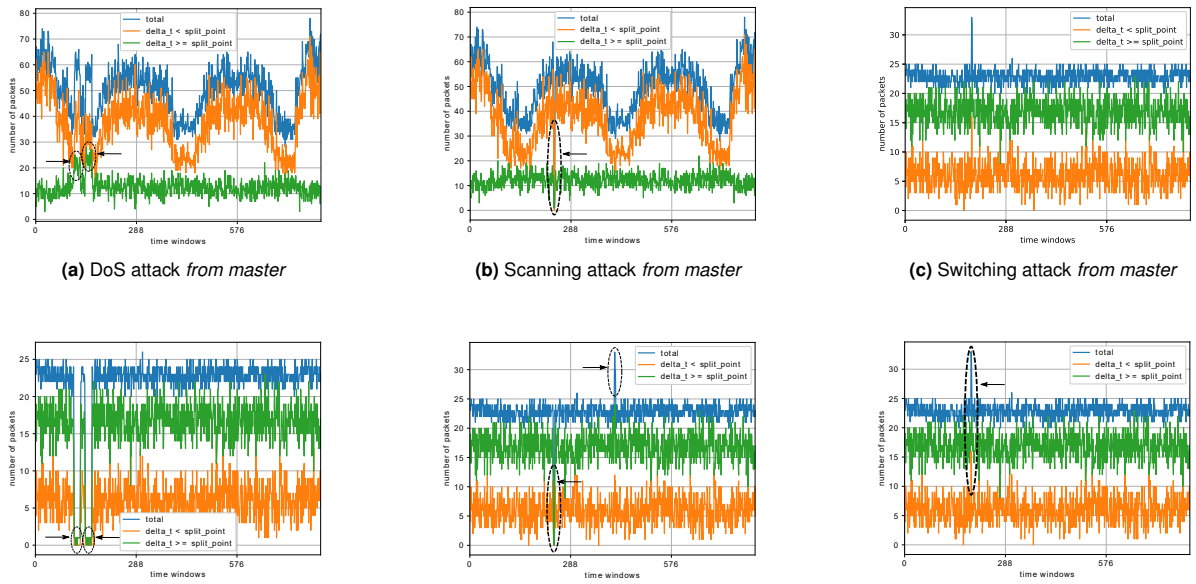
By observing differences from the profile in different regions (total, lower, upper) and directions, we can also identify a type of the attack as depicted in Fig. 6. Full description of our experiments is available at [20].

VI. CONCLUSION

Industrial systems are attractive targets for attackers. Therefore, their cyber security is of paramount importance. The ICS traffic anomaly detection techniques can help identify system malfunctions and even cyber attack activities by finding irregularities in the monitored communication. This paper presents a simple but accurate anomaly detection method applied to IEC 104 and GOOSE traffic. The method builds the statistical model of normal communication by exploiting packet inter-arrival times. Based their statistical distribution, packets are grouped into regions within a monitored window that represent a statistical profile of the communication. The different approaches to construct these ranges have been considered and tested in our experiments. Two important aspects of anomaly detection were considered: fast computation and the small number of false positives. We have identified that false positives can be suppressed using the 3-value-detection approach that compares three consecutive time windows of the monitored traffic to the profile and raises an alert only if an anomaly is detected in at least two of them. We demonstrated that profiles built for each direction of ICS communication could detect the most considered attacks. Finally, the detection capabilities were compared to the AD method based on probabilistic automata.

The conducted experiments demonstrated high accuracy of the proposed method similar to more complex AD methods, while the computation costs of building a profile and evaluating the monitored traffic are substantially lower.

The method was demonstrated on a variety of datasets consisting of IEC 104 and GOOSE traffic. The future work aims at the integration of the method in the ICS traffic monitoring system in order to improve the accuracy and reduce false positives by the combination of various detection methods.



(a) DoS attack *from master* **(b)** Scanning attack *from master* **(c)** Switching attack *from master*

(d) DoS attack *to master* **(e)** Scanning attack *to master* **(f)** Switching attack *to master*

Fig. 5. DoS, scanning and switching attacks detection.

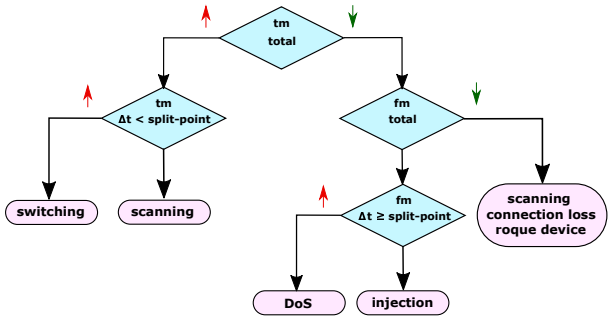


Fig. 6. Identification of attacks based on the changes from expected values.

ACKNOWLEDGMENT

This work is supported by the project "Security Monitoring of ICS Communication (Bonnet)", no. VI20192022138, funded by Ministry of Interior of the Czech Republic.

REFERENCES

[1] K. Schwab, *The Fourth Industrial Revolution*. USA: Crown Publishing Group, 2017.

[2] K. E. Hemsley and D. R. E. Fisher, "History of Industrial Control System Cyber Incidents," no. INL/CON-18-44411-Revision-2, 12 2018. [Online]. Available: https://www.osti.gov/biblio/1505628

[3] Dragos, "CrashOverride. Analysis of the Threat of Electric Grid Operations." Dragos Inc., Tech. Rep., June 2017.

[4] M.-A. Russon, "US fuel pipeline hackers 'didn't mean to create problems'," *BBC News*, May 2021. [Online]. Available: https://www.bbc.com/news/business-57050690

[5] R. R. R. Barbosa, R. Sadre, and A. Pras, "Difficulties in Modeling SCADA Traffic: A Comparative Analysis," in *The 13th International Conference on Passive and Active Measurement*, 2012, pp. 126–135.

[6] D. Formby, A. Walid, and R. Beyah, "A case study in power substation network dynamics," vol. 1, no. 1, Jun. 2017.

[7] C.-Y. Lin and S. Nadjm-Tehrani, "Understanding IEC-60870-5-104 Traffic Patterns in SCADA Networks," in *The 4th ACM Workshop on Cyber-Physical System Security*, ser. CPSS '18, 2018, pp. 51–60.

[8] P. Matoušek, O. Ryšavý, M. Grégr, and V. Havlena, "Flow based monitoring of ICS communication in the smart grid," *Journal of Information Security and Applications*, vol. 54, p. 102535, 2020.

[9] S. S. Jung, D. Formby, C. Day, and R. Beyah, "A first look at machine-to-machine power grid network traffic," in *IEEE International Conference on Smart Grid Communications*, Nov 2014, pp. 884–889.

[10] A. Kleinmann and A. Wool, "A statechart-based anomaly detection model for multi-threaded SCADA systems," in *Int. Conference on Critical Information Infrastructures Security*, 2015, pp. 132–144.

[11] A. Valdes and S. Cheung, "Communication pattern anomaly detection in process control systems," in *2009 IEEE Conference on Technologies for Homeland Security*, May 2009, pp. 22–29.

[12] R. R. R. Barbosa, R. Sadre, and A. Pras, "A first look into SCADA network traffic," in *2012 IEEE Network Operations and Management Symposium*, April 2012, pp. 518–521.

[13] ——, "Towards periodicity based anomaly detection in SCADA networks," in *Proceedings of IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA)*, Sept 2012, pp. 1–4.

[14] C.-Y. Lin, S. Nadjm-Tehrani, and M. Asplund, "Timing-based anomaly detection in SCADA networks," in *International Conference on Critical Information Infrastructures Security*. Springer, 2017, pp. 48–59.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009.

[16] C. Perkins, *RTP: Audio and Video for the Internet*. Addison-Wesley, 2003.

[17] A. Santos Da Silva, C. C. Machado, R. V. Bisol, L. Z. Granville, and A. Schaeffer-Filho, "Identification and Selection of Flow Features for Accurate Traffic Classification in SDN," in *IEEE 14th Int. Symposium on Network Computing and Applications*, 2015, pp. 134–141.

[18] F. Pukelsheim, "The Three Sigma Rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[19] J. W. Tukey, *Exploratory data analysis*. Addison-Wesley, 1977.

[20] I. Burgetová and P. Matoušek, "Statistical Methods for Anomaly Detection in Industrial Communication," Brno University of Technology, Tech. Rep. IT-TR-2021-01, 2021.

[21] P. Matoušek, "Description and analysis of IEC 104 Protocol," Brno University of Technology, Tech. Rep. FIT-TR-2017-12, 2017.

[22] ——, "Description of IEC 61850 Communication," Brno University of Technology, Tech. Rep. FIT-TR-2018-01, 2018.

[23] P. Varga, "Analyzing Packet Interarrival Times Distribution to Detect Network Bottleneck," in *IFIP EUNICE: Networks and Applications Towards a Ubiquitously Connected World*, vol. 196, 2006, pp. 134–141.

[24] P. Matoušek, V. Havlena, and L. Holík, "Efficient Modelling of ICS Communication For Anomaly Detection Using Probabilistic Automata," in *IFIP/IEEE International Symposium on Integrated Network Management*, 2021, pp. 1–9.