Statistical Methods for Anomaly Detection in Industrial Communication

Technical Report, FIT BUT

Petr Matoušek and Ivana Burgetová and Nelson Makau Mutua

Technical Report no. FIT-TR-2021-01
Faculty of Information Technology, Brno University of Technology

Last modified: October 20, 2022

Contents

Chapter 1

Introduction

Protection of industrial communication systems against cyber attacks has become a great challenge during the past years due to the convergence of Operational Technologies (OT) and Information Technologies (IT), adoption of the TCP/IP to industrial networks, and the rising level of automation and intelligent control of industrial processes. Security and safety of critical infrastructure systems that include power plants, substations, water and gas distribution, traffic control systems, etc., can be implemented on various levels.

One way is to monitor network communication of Industrial Control Systems (ICS) and analyze its typical communication patterns. In our previous work [18] we extended flow-based monitoring system standardized by IPFIX protocol with selected application-level data extracted from ICS protocols. This enhancement provides additional monitoring data related to ICS packets which can be further analyzed using anomaly detection methods.

This report focuses on application of selected statistical methods to anomaly detection of ICS protocols deployed in smart grids, namely IEC 104, GOOSE and MMS. Using extended IPFIX monitoring of ICS protocols we are able to enhance visibility of ICS transmission and disclose anomalies that differ from expected communication. Industrial network stations are typically pre-configured hardware devices that operate in master-slave mode and exhibits stable and periodic communication patterns over a long time. Due to the stability of ICS communication, statistical models present a natural way for detection of common ICS anomalies including cyber security threats, device malfunctioning, network congestion, etc. In addition, monitoring time properties of ICS/SCADA protocols is extremely important to secure proper operation of critical industrial processes which work in real-time environment.

For probabilistic modeling of network behavior we employ the following statistical features: distribution of packet inter-arrival times, packet size, and packet direction. Unlike previous works that apply statistical-based anomaly

detection on SCADA networks, we do not observe the above mentioned feature on IP or TCP layers, but we model behavior of application protocols which gives a more precise insight into ICS behavior.

This report presents the results of our experiments with three statistical methods: the Box Plot, Three Sigma Rule and Local Outlier Factor (LOF) which worked best for ICS datasets.

The main advantage of the statistical model is that it does not require high processing power and time to extract packet features and build the model, so it can be easily implemented on a monitoring probe or IDS device. On the other hand, statistical methods are sensitive to outliers which are particular data with exceptionally low probability that can be incorrectly marked as anomalies. Hence, an important question for statistical modeling of network communication is how to represent probabilistic distribution of a given data set so that the model is precise enough and includes even samples with low probability, and at the same time is able to correctly detect any anomaly. The level of detection accuracy is usually controlled by a threshold variable which should be determined with respect to a specific environment.

1.1 Structure of the Report

The report is structured as follows. Chapter 2 overviews preliminaries covering the theory of used statistical methods and a description of industrial protocols used in our experiments. Chapter 3 discusses results of previous work focused on utilizing statistical methods for anomaly detection of ICS and SCADA traffic. Chapter 4 forms the core of the report. It provides a deep insight into processing of ICS data, building a probabilistic model of communication, and anomaly detection. Our experiments are provided with dataset created at our University or obtained from our partners, see Section 4.1. The last chapter concludes our results and discusses possible deployment. The full reports of our experiments can be found in Appendices A and B.

1.2 Acknowledgement

This work was funded by project “Security Monitoring of ICS Communication (Bonnet)” (2019–2022), no. VI20192022138, provided by Ministry of Interior of the Czech Republic.

Chapter 2

Preliminaries

2.1 Statistical Methods

Since an outlier detection is a non-trivial task, many different methods have been developed to address this issue. These methods differ in the principle used, in the computational complexity and they can be suitable for different datasets. In our study we focus on statistical methods, because they are able to detect outliers in reasonable time and they are suitable for our datasets.

Statistical methods assume data spreading according to some distribution or probability model (e.g. normal distribution). Then, the probability of a particular data point being generated can be estimated from this model. Data points with exceptionally low probability can be labeled as an outliers (anomalies). There exists several statistical tests that allows outlier evaluation. For our purposes we tested two basic test for fast outlier detection: the Three Sigma Rule [19] and Box Plot Rule based on interquartile range (IQR) [23].

2.1.1 Three Sigma Rule

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 [20]. Remaining points (roughly 0.3% of data points) are labeled as outliers by this method.

2.1.2 Box Plot Rule based on Interquartile Range

Interquartile Range (IQR) outlier detection relies on different measures of dispersion of the data. It utilizes percentiles and is defined as the difference between the 75th (Q_3) and 25th (Q_1) percentiles of the data. With this measure, the range of normal values is defined as $\langle Q_1 - 1.5 * IQR, Q_3 + 1.5 * IQR \rangle$, while points outside this range are marked as outliers. For normal distribution, roughly 99.3% of data points lie within this interval.

2.1.3 Limitations of Basic Tests

Described statistical methods are loaded by three important limitations. First of all, they can be used for analysis of data bescribed by single attribute (one value). Secondly, they can be applied on unimodal data distribution. Finally, statistical methods suspect normal data distribution.

2.2 LOF Method for Outlier Detection

Local Outlier Factor (LOF) is a popular method for anomaly detection. It compares the local density of the target object with the local densities of its neighbors. If the density around the target data point is significantly lower, the object is marked as an outlier. The main advantage of this method is that it can correctly distinguish outliers even in datasets that are a mixture of clusters of points with different densities.

For our experiments we employed the LOF implementation from scikit-learn python library. There are two basic modes of this method, which are suitable for different situations:

- *Outlier detection:* simply detect the outliers in the given dataset. For each target data point it considers all other data points from the dataset. Using this mode the anomalies cannot form the clusters, otherwise they are not detected by the method.
- *Novelty detection*: in this mode it is possible to distinguish the training set without outliers and the testing set, where outliers may occur. Then, for each point from the testing dataset only the data points from training set are considered. Therefore, the outliers in the testing set can form clusters.

For our experiments we used *novelty detection* mode since the attacks in our dataset lasted for a longer period of time and led to a larger number of similar data points that represented them.

LOF method outputs can be significantly affected by the choice of two key parameters:

- *n_ neighbors:* determines the number of neighbors that are considered for local density estimation.
- *contamination:* allows to specify the expected proportion of outliers in the dataset. It affects the threshold which is used for the final labeling of the points.

For our experiments, we decided to use the default value *'auto'* for the parameter *contamination*. Since the other values forced the method to mark more or less outliers and suppressed the natural labeling of the data. And

they also led to many false positive points (false alarms) or to some false negative points (missed attacks).

TODO: přidat obrázek, který ukazuje vliv parametru auto?

The best values for the *n_ neighbors* parameter for each dataset we searched during the validation phase with respect to the number of produced false positive points. We tested the values from the interval 3 to 30. The bast values for each dataset are listed in tables in section 4.3.

2.3 Industrial Communication and Security

Industrial communication typically includes control and monitoring transmissions that are exchanged between Intelligent Electronic Devices (IEDs), Human-Machine Interfaces (HMIs), control stations, and gateways. Connected devices typically use standardized ICS protocols like IEC 104, MMS or GOOSE [12]. The communication is often not secured which makes it an easy target for cyber attacks. In the recent years, industrial systems experienced several damaging attacks on critical infrastructure [**?**, 11, 5, 22]. Such attacks were often driven by malware installed on an internal control station. The malware usually employs industrial communications to discover ICS network resources, requests execution of unauthorized commands, collects sensitive data, or even manipulates ICS processes, see Fig. 2.1, that is not easy to detect.

To better understand behavior of ICS protocols, we give here a short overview of ICS protocols IEC 104 and GOOSE that we later use in our experiments. In addition, we also define the inter-arrival time that we observe for modelling ICS behavior.

Protocols IEC 104

The protocol IEC 104 [15] 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 periodically 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).

Protocol GOOSE

GOOSE [16] is an Ethernet-based protocol used for Intelligent Electronic Devices (IED) that transfers time-critical events in substations. The com-

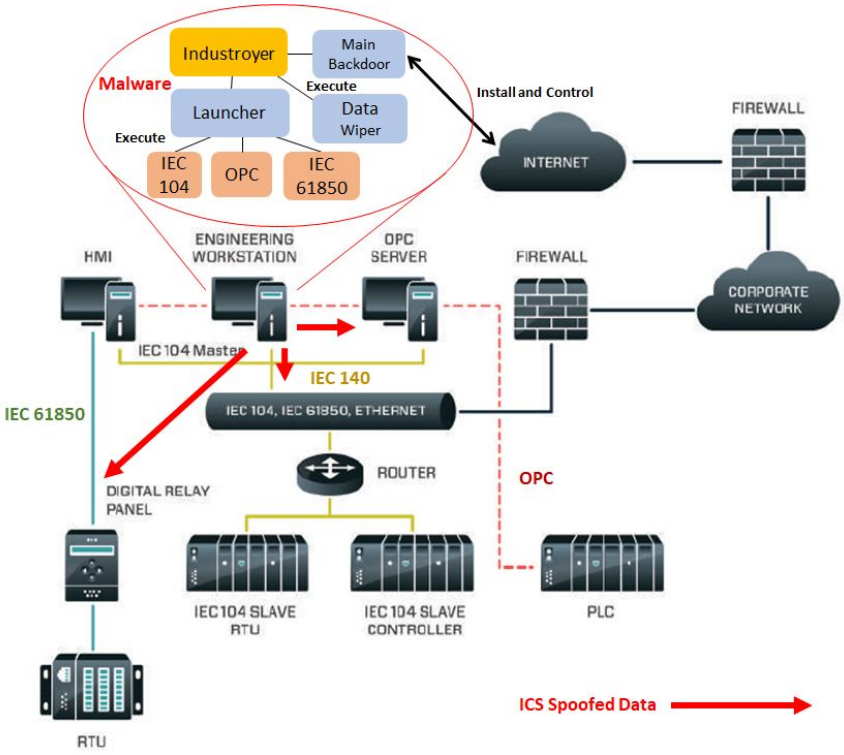


Figure 2.1: Industroyer Attack on Power Grid in 2016 [8].

munication 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.

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. 2.2, but the distribution stays the same regardless of a probe location.

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

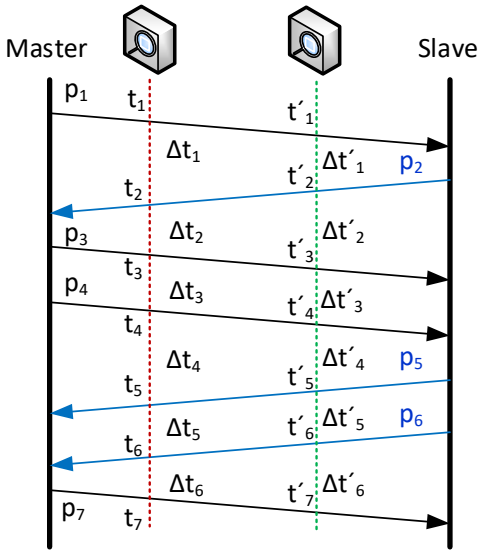


Figure 2.2: Measuring inter-arrival times

model better fits GOOSE publish-subscribe mechanism.

Chapter 3

Related Work

Statistical-based anomaly detection is one of the widely used techniques [1, 9]. The basic idea of statistical methods is to detect significant deviations of observed behavior from the normal one. Successful statistical modeling requires stable and predictable behavior of modeled traffic. Stability and regularity of ICS/SCADA communication was previously studied and demonstrated for major industrial protocols like Modbus [24], IEC 104 [13], or DNP3 [10].

In our own work [17] we observed regularity of Internet of Things traffic and created a simple statistical model for representing resource usage of Constrained Application Protocol (CoAP) [21]. The CoAP resource was described by a pair *operation* (e.g., PUT command) and *resource URL address* (e.g., floor_light). In each time window we observed the number of packets and octets associated with the resource and created a usage profile related to the specific resource and device. The model was created by application of expectation-maximization (EM) algorithm [7] and represented as a joint probability function and computed threshold value. The obtained results showed hit ratio (recall) about 75 to 90% with false positive ration about 2 to 6.4%. Since ICS traffic is more stable and regular than CoAP communication, we applied a statistical model with simpler computation which, however, gives quite precise results.

Statistical properties of ICS communication were widely explored by Barbosa, et al. in [2, 4] where the authors compared periodicity, throughput and topology changes in SCADA and SNMP traffic. Their results show that SCADA communication exhibits periodic behavior at a smaller scale, has constant throughput over a long period of time, and keeps a stable number of connections. Its periodicity is caused by a polling mechanism used to retrieve data from SCADA slaves [3]. The authors demonstrated that attacks like scanning, denial of service, network protocol manipulation, or buffer overflow disturb the periodicity, thus, it can be detected by anomaly detection. For modeling the SCADA communication, Barbose et al. use

time series representing the number of packets belonging to a specific flow. During periodicity learning, they generate a periodogram for each flow using Fast Fourier Transform. In detection phase, using discrete-time Short-Time Fourier Transform they create a spectrogram for monitoring changes in periodicity. Our approach comes out of Barbosa's observations. Instead of monitoring a simple number of transmitted packets we provide a more subtle classification using arrival times distribution. This is faster in computation while providing similar results.

Valdes and Cheung [24] introduced pattern-based and flow-based anomaly detection of ICS communication. Their patterns include source and destination IP addresses and ports. During detection, they monitor previous n-occurrences of the pattern and compute the historical probability of the pattern. If the probability is less than the given threshold, an alert is generated. Their solution includes a periodic update of the patterns and pruning the rare patterns. The second technique presented by Valdes and Cheung uses flow records for anomaly detection. Flow records include more attributes like source and destination addresses, the time of the last packet, the average number of bytes per packet, the variance of bytes per packet, or mean and variance of packet inter-arrival time. Similarly to pattern-based detection, they compare the traffic with historical flow records and compute a difference. If a record does not exist or differs too much, the alert is raised. They tested the approach on MODBUS network with periodic data retrieval. They were able to detect anomalies like scanning, modified data, denial of service, and system degradation. Unfortunately, their results do not show the number of false positives and implementation. Our approach does not observe individual flows but creates a model for entire communication between groups of communicating ICS nodes.

Lin and Nadjm-Tehrani [13] analyzed timing patterns of spontaneous events of the IEC 104 protocol which are asynchronously generated by an RTU. The authors model inter-arrival times of IEC 104 packets using Probabilistic Suffix Trees (PSTs) and analyze phase transitions, predictability, and frequent patterns. They describe inter-arrival times as sequences of symbols representing groups of "similar" inter-arrival times. The symbolic sequences are further processed (smoothing, finding boundaries) and used to create a PST. Having the PST, the authors define a phase transition, i.e., a period of time during which the distribution of inter-arrival times is stable. They found five groups of traffic patterns based on phase transitions: strongly cyclic, weakly cyclic, stable, bursty, and transitional communication. Using the probability of communication patterns, they predict future behavior, i.e., that a certain pattern would appear in the next segment. The approach is, however, computationally very intensive. We also deal with IEC 104 communication, but we do not restrict to spontaneous events only but model all IEC 104 packets. We use a simpler statistical model with lower computational requirements.

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

Chapter 4

Anomaly Detection

In this chapter, we describe the proposed statistical method for anomaly detection in industrial communication. As we mentioned in 2.1.3, statistical methods are loaded by few limitations. Therefore, it is necessary to describe the communication flow by a set of suitable characteristics. These chacteristics should be stable enough to provide reasonable model of normal traffic and they should be also affected by possible attacks on these systems. For such description, it is possible to define the range of usual values for each variable in normal traffic for given communication flow. Then, during monitoring of the system, we can use these ranges to detect the anomalies.

We start this section with the description of the datasets utilized in our experiments. Then, we explain how we process the traffic data, gather the characteristics and build the traffic description. Subsequently, we describe how this description can be utilized for traffic monitoring and anomaly detection. Finally, we describe performed experiments and obtained results.

4.1 Datasets

In our study, we focused on two industrial protocols: GOOSE and IEC104. This section contains the description of the used datasets.

For our experiments we used several datasets with IEC 104 and GOOSE traffic, see Table 4.1. 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.

4.1.1 IEC104

For our experiments, we deal with the IEC 60870-5-104 (shortly IEC104) protocol, that is widely used in smart grids for substation control. We uti-

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

lized seven different datasets of ICS flows for IEC 104 protocol listed in Table 4.1.

Dataset	Packets	Time	Devices	Organization
10122018-104Mega	104,534	4h 53min	4	VUT
13122018-mega104	1,460,829	71h 17min	14	VUT
mega104-14-12-18	14,597	15h 38min	2	VUT
mega104-17-12-18	58,931	67h 55min	2	VUT
KTH-RTU8	3,463,632	162h 30min	2	RTSLab
KTH-RTU11	1,836,723	162h 30min	2	RTSLab
RICS	883,183	309h 40min	2	RTSLab

Table 4.1: IEC104 Datasets. Column *Devices* shows the number of communicating devices in the given dataset.

In order to test the ability of attack detection we also used a special simulated set of records of ICS flow. These records were obtained by injection into or removing communication from the record `mega104-17-12-18`. We experimented with six types of attacks:

- Injection attack
- Connection loss
- DoS attack
- Rogue devices
- Scanning attack
- Switching attack

The details about these attacks are summarized in table XXXX.
TODO:Doplnit tabulku, která by shrnovala útoky
čas, trvání, počet přidaných/odstaněných paketů?

4.1.2 GOOSE

We tested the suitability of the proposed method also for GOOSE protocol. We validated our method on the records listed in table 4.2. In the case of GOOSE communication datasets, we focused on flows instead of packets.

4.2 Data pre-processing

Statistical anomaly detection methods require stable description of the given IEC traffic. According to [6] it is useful to describe the communication between two devices as a flow of packets, where for each packet i two properties

Dataset	Flows	Time	Devices
gics-goose	2,177	19h 26min	4
goose-mms3	2,566	43h 2min	4

Table 4.2: GOOSE Datasets. Column *Devices* shows the number of communicating devices in the given dataset.

are considered: its size s_i and the inter-arrival time between previous and current packet Δt_i. Hovewer, in statistical modelling we do not care about the individual packets, rather we characterize the flow by some summary statistics. Basically, we describe the given communication by the amount of packets transmitted during five minute window in the given direction between considered devices. To make our characterization more subtle and precise, we utilize more features obtained by splitting the number of transmitted packets into the groups of packets according to their inter-arrival time. In order to get really stable characteristics, several issues needed to be solved:

1. *Partitioning the traffic into two directions (from the master, towards the master).* As usual in ICS/SCADA communication, there is one device (master) that communicate with all other devices in each of our datasets. Originally, we partitioned each dataset into the communications of each pair of devices. Hovewer, these partitioning has proved unsuitable for some datasets (namely 10122018-104Mega and 13122018-mega104). In such datasets, each pair of devices communicate only for short time period and therefore, we were not able to find stable description of the communication for these pairs of devices (see fig. 4.1). One possible solution allows two normal state of such communication. However, this solution will prevent the detection of many attacks, as they lead to a decrease in the number of transmitted packets. After closer examination, we realized that in these datasets at each time, the master device communicate with only one device, and also the communication of the master is continuous. Therefore, we split the traffic into two directions - *from master* and *to master* (see fig. A.1).

2. *Determining inter-arrival time for each packet.* Due to the division of the traffic into two directions, there are two possibilities to determine an inter-arrival time for each packet: before the division of the traffic and after it. In principle, both possibilities can be used, however with the determination of inter-arrival time before the division we achieved a slightly better results in anomaly detection (we detected some attacks in both directions). In such case, missing or added packets in one direction will affect an inter-arrival time of the packets in the other

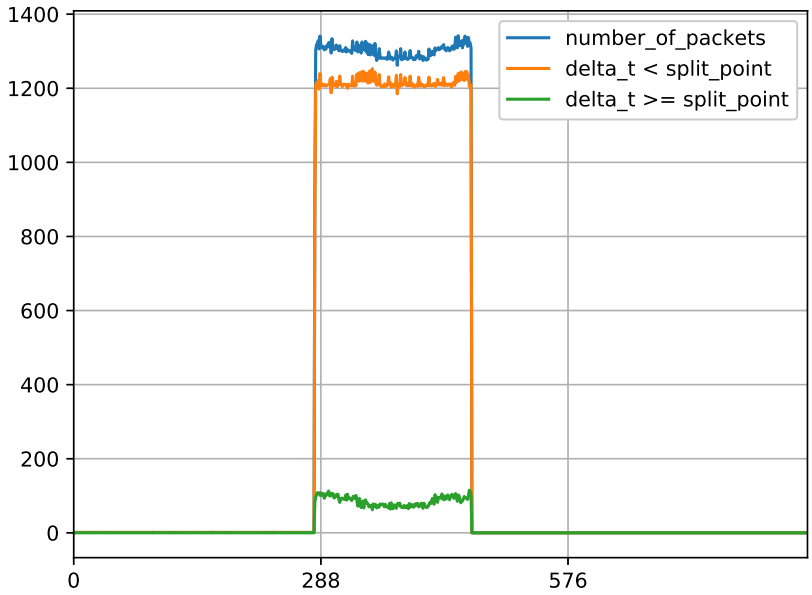


Figure 4.1: Graph of the amounts of transmitted packets in five minute windows in `13122018-mega104` dataset (conversation of one pair of devices in the direction *from master*).

direction and therefore we can detect such change in both directions.

3. *Finding suitable split-point for the given IEC communication.* In our research we tested several ways how to split the amount of transmitted packets according to their inter-arrival time to obtain more subtle characterization of the given traffic.

   First of all, we utilized maximal inter-arrival time observed in the given dataset and determined four equally large intervals that cover all possible inter-arrival times of the given dataset. Not surprisingly, such intervals are not suitable for the traffic characterization. Typically, inter-arrival time of few packets is much grater then inter-arrival time of the rest of the packets (see table 4.3). With such intervals, the majority of packets falls into one or two intervals of the inter-arrival time. On the other hand, very few packets fall into the remaining intervals, so they do not provide any additional information about the given traffic (see fig. 4.2).

   Secondly, we searched for some predefined spilt-points that would reasonably divide the packets of all datasets. We also reduced the number of split-points and intervals of the inter-arrival time that we search for,

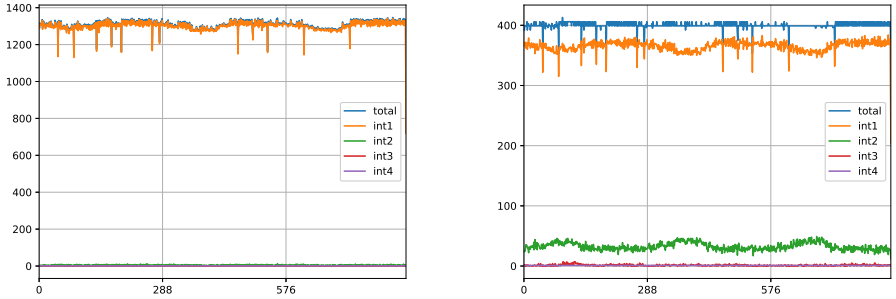
Dataset	Dir.	min	25%	50%	75%	max
10122018-104Mega	fm	0.00000	0.0000	0.0003	0.0005	8.2033
	tm	0.0000	0.0002	0.0004	0.0598	5.2006
13122018-mega104	fm	0.0000	0.0000	0.0003	0.0004	16.1905
	tm	0.0000	0.0002	0.0004	0.0600	10.1331
mega104-14-12-18	fm	0.0000	1.6701	3.2010	5.2896	19.7166
	tm	0.0000	1.0076	3.0301	6.0784	19.2687
mega104-17-12-18	fm	0.0000	1.9989	3.5909	5.6002	19.9873
	tm	0.0001	1.0091	3.0332	6.0831	19.2696
KTH-RTU8	fm	0.0000	0.2025	0.2044	0.2184	1.2111
	tm	0.0000	0.0142	0.0145	0.0146	15.5452
KTH-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 4.3: Inter-arrival time distribution in individual datasets and directions (five-number summary).

since the total amount of packets transmitted in five minute window in some datasets is not large enough to be divided into more than three intervals. Unfortunately, split-points that are useful for some datasets does not divide the packets of another datasets at all (see fig. 4.3). In addition, split-points suitable for one direction do not work well for the other direction (see fig. 4.4). The difficulty of finding split-point suitable for all datasets and direction is also apparent from the distribution of the inter-arrival times in individual datasets and directions. These distributions differ significantly between individual datasets and directions.

Given these facts, we suggest the method that automatically finds the suitable split-points for individual direction of the given traffic. Also, we recommend to reduced the number of split-points and intervals of the inter-arrival time that we search for in order to reduce the complexity of this task. Therefore we search for one split-point for each direction that provide two additional characteristics for the given direction of the traffic.

In order to set up the split-point automatically we suggest to utilize the distribution of the inter-arrival time of packets transmitted in the given direction of the traffic and also the standard deviation of the resulting distribution of the packets. Undoubtedly, different split-points produce different characteristics. Our experiments show the potential of some split-points to filter-out the periodic behavior from at least on of the resulting characteristic. Such split-points are suitable for the statistical



(a) Direction *from master*. (b) Direction *to master*.

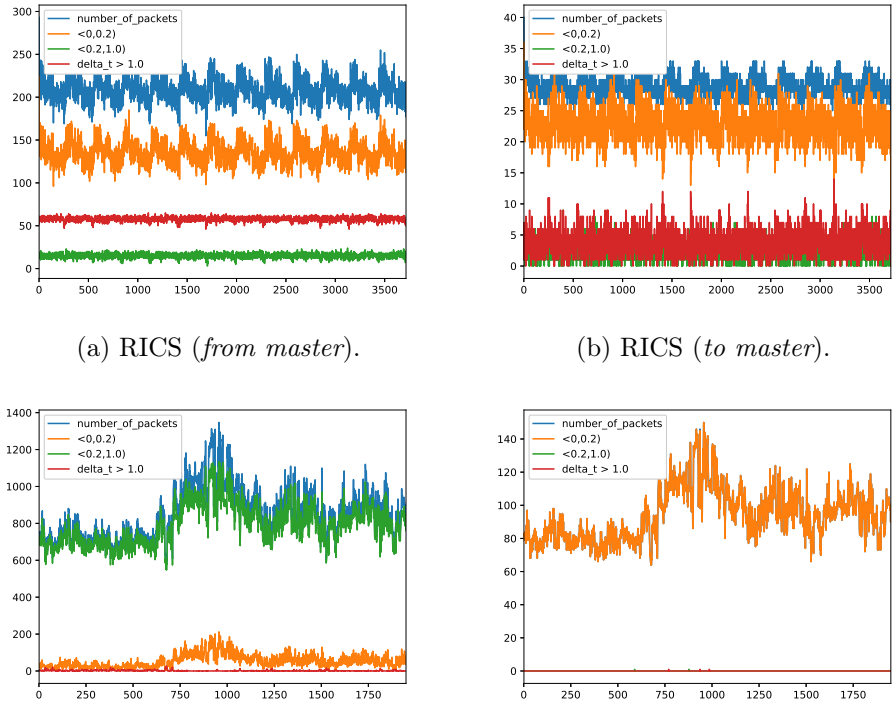
Figure 4.2: Graph of the amount of packets transmitted in five minute windows in `13122018-mega104` dataset, additional characteristics show the effect of using four equally large intervals of inter-arrival time.

anomaly detection methods as they produce stable characteristics (see fig. 4.5 and 4.6). Unfortunately, it is not clear where to look for a suitable split-point in the distribution of the measured inter-arrival times. While for some datasets the median of the measured inter-arrival times is an accurate split-point, for others it is a value close to the quartile Q1 or Q3 (see table 4.4 and 4.5). Therefore, we recommend to test four different values of split-point (Q1, Q2, mean and Q3) and select the one that produce such packet distribution, where one of the characteristics is the most stable one (characteristic with the smallest standard deviation).

Split-point		$\Delta t < split-point$		$\Delta t \geq split-point$	
Δt distribution	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**

Table 4.4: Split-point selection in dataset `mega104-17-12-18` for the direction *from master*. Table shows the possible value of split-points and *mean* and *standard deviation* of the resulting characteristics. In this case, the value 5.66 is used as split-point as it produce characteristic with the smallest standard deviation. (Δt distribution is derived from the first 48 hours of the captured traffic in order to decrease the influence of the periodicity.)

One additional issue should be mentioned in the context of suitable split-points choice. For some datasets the split-points which lead to most stable characteristics divide the amount of packets in such man-



(a) RICS (*from master*).

(b) RICS (*to master*).

(c) KTH-RTU11 (*from master*).

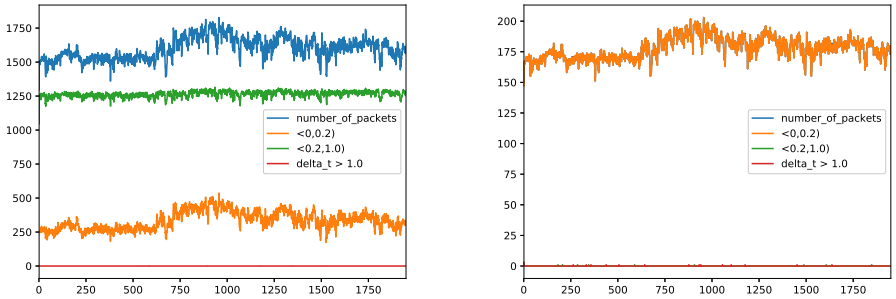
(d) KTH-RTU11 (*to master*).

Figure 4.3: Graphs of the amount of packets transmitted in five minute windows in `RICS` dataset (*(a)* and *(b)*) and `KTH-RTU11` dataset (*(c)* and *(d)*), additional characteristics show the effect of using the same predefined split-points of inter-arrival time for both datasets.

ner, that the stable characteristic contain only few packets in each five minute window. Such characteristic is not effective in anomaly detection with statistical methods. This is the case of the characteristics their mean is not grater than the triple of standard deviation. Then, the range of normal values exceeds below zero and the characteristic is not capable to detect many types of attacks. Therefore, we suggest to add a second condition for selection of a suitable split-point: choose a split-point that produce such characteristics, one of which is the one with the smallest possible standard deviation and at the same time the following condition applies to it: $mean - 3\sigma > 0$.

Due to previous observations, we process individual datasets and gather traffic characteristics as follows:

- Consider an input consisting of the sequence $T = (t_1^d, t_2^d, ..., t_n^d)$, where $d \in \{t, f\}$ denotes the direction *to master* and *from master* and t_i



(a) KTH-RTU8 (*from master*). (b) KTH-RTU8 (*to master*).

Figure 4.4: Graphs of the amount of packets transmitted in five minute windows in `KTH-RTU8` dataset, additional characteristics show the effect of using the same predefined split-points of inter-arrival time for both directions.

Split-point		$\Delta t < split-point$		$\Delta t \geq split-point$	
Δt distribution	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

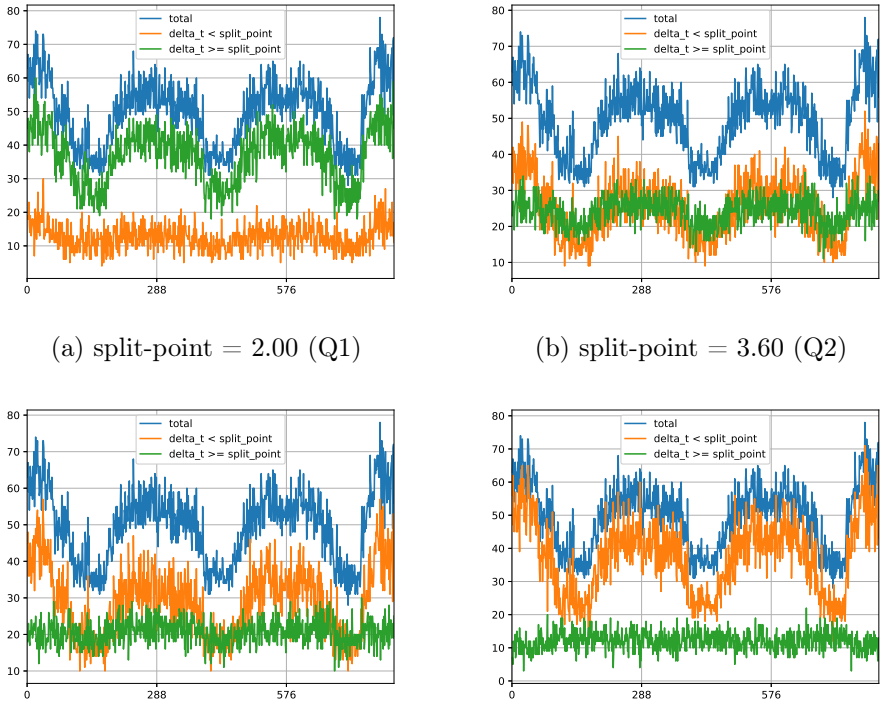
Table 4.5: Split-point selection in dataset `mega104-17-12-18` for the direction *to master*. Table shows the possible value of split-points and *mean* and *standard deviation* of the resulting characteristics. In this case, the value 1.01 is used as split-point as it produce characteristic with the smallest standard deviation. (Δt distribution is derived from the first 48 hours of the captured traffic in order to decrease the influence of the periodicity.)

denotes the time of capturing for the individual transmitted packets.

1. Determine inter-arrival time for each packet $\Delta t_i^d = t_i^d - t_{i-1}$ and create a new sequence $\Delta T = (\Delta t_1^d, \Delta t_2^d, ..., \Delta t_n^d)$ that hold the inter-arrival time for each packet.

2. Partition the input sequence and the sequence of inter-arrival times into two subsequences according to the direction of individual values: $T^f = (t_i^d : d = f \wedge 1 \leq i \leq n)$ and $T^t = (t_i^d : d = t \wedge 1 \leq i \leq n)$, $\Delta T^f = (\Delta t_i^d : d = f \wedge 1 \leq i \leq n)$ and $\Delta T^t = (\Delta t_i^d : d = t \wedge 1 \leq i \leq n)$. Subsequences T^f and ΔT^f represent the direction *from master* and subsequences T^t and ΔT^t represent the direction *to master*.

3. Consider the distribution of the values in ΔT^f and ΔT^t subsequences and find the set of candidates for split-points for each distributions:



(a) split-point = 2.00 (Q1)

(b) split-point = 3.60 (Q2)

(c) split-point = 4.13 (mean)

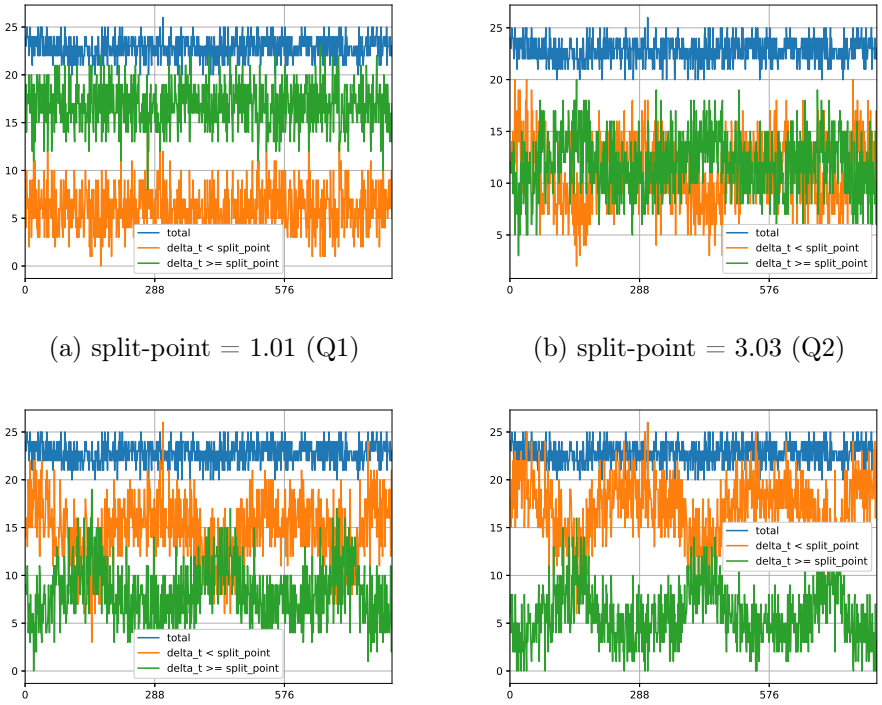
(d) split-point = 5.66 (Q3)

Figure 4.5: Graph of the amount of packets transmitted in five minute windows in `mega104-17-12-18` dataset, additional characteristics show the effect of using different split-points (direction from master).

$D^f = \{Q1^f, Q2^f, mean^f, Q3^f\}$ and $D^t = \{Q1^t, Q2^t, mean^t, Q3^t\}$, where $Q1$ is the first quartile, $Q2$ is the median, $Q3$ is the third quartile and $mean$ is arithmetic mean.

4. For each direction and each $sp \in D^d$ find five minute characteristics of the traffic as the amount of packets transmitted during each five minute window with specified inter-arrival time:
   $S^{d,L}_{sp} = (a_1, ..., a_m : a_j = |t^d_i|, k * w \leq t_i < (k+1) * w, k = 0...\frac{t_n}{w} \quad \wedge \Delta t^d_i < sp)$ and
   $S^{d,U}_{sp} = (b_1, ..., b_m : b_j = |t^d_i|, k * w \leq t_i < (k+1) * w, k = 0...\frac{t_n}{w} \wedge \Delta t^d_i \geq sp)$,
   where w is the size of time window ($w = 300$ for five minute window) and $|t^d_i|$ denotes the number of elements of sequence T^d that satisfies specified conditions.

5. For each direction find the set of mean values (two values for each



(a) split-point = 1.01 (Q1)

(b) split-point = 3.03 (Q2)

(c) split-point = 4.29 (mean)

(d) split-point = 6.08 (Q3)

Figure 4.6: Graph of the amount of packets transmitted in five minute windows in `mega104-17-12-18` dataset, additional characteristics show the effect of using different split-points (direction *to master*).

candidate split-point):

$M^d = \{m_{sp}^{d,e} : e \in \{L, U\}, sp \in D^d\}$,

where $m_{sp}^{d,e} = \frac{1}{m}\sum_{j=1}^{m} a_j$ and $a_j \in S_{sp}^{d,e}$.

And also set of standard deviations:

$DEV^d = \{\sigma_{sp}^{d,e} : e \in \{L, U\}, sp \in D^d\}$,

where $\sigma_{sp}^{d,e} = \sqrt{\frac{1}{n}\sum_{j=1}^{m}(a_j - m_{sp}^{d,e})^2}$ and $a_j \in S_{sp}^{d,e}$.

6. For each direction find the smallest $\sigma_{sp}^{d,e}$ in DEV^d that also satisfies condition: $m_{sp}^{d,e} - 3 * \sigma_{sp}^{d,e} > 0$. Save the sp^d value as the selected split-point for the given direction (ssp^d).

7. For each direction find a summary five minute characteristic of the traffic:

   $S^{d,T} = (c_1, ..., c_m : c_j = |t_i^d|, k * w < t_i \leq (k+1) * w, k = 0...\frac{t_n}{w})$

and find the mean $m^{d,T}$ and standard deviation $\sigma^{d,T}$ for this characteristic.

8. Using 3-sigma rule, filter out the outlier values from $S^{d,T}$ and $S^{d,e}_{ssp}$ characteristics generated with selected split-point (filter out the points that does not fit in the ranfe $< m - 3 * \sigma, m + 3 * \sigma >$):
   $S^{d,e,R}_{ssp} = (a'_1, ..., a'_o : a'_j \in S^{d,e}_{ssp} \wedge (m^{d,T}_{ssp} - 3 * \sigma^{d,T}_{ssp}) \leq a'_j \leq (m^{d,T}_{ssp} + 3 * \sigma^{d,T}_{ssp}), j = 1...o)$.
   $S^{d,T,R} = (c'_1, ..., c'_o : c'_j \in S^{d,T} \wedge (m^{d,T} - 3 * \sigma^{d,T}) \leq c'_j \leq (m^{d,T} + 3 * \sigma^{d,T}), j = 1...o)$,

9. Find the mean and standard deviation for each reduced characteristic: $m^{d,T,R}$, $\sigma^{d,T,R}$, $m^{d,L,R}_{ssp}$, $\sigma^{d,L,R}_{ssp}$, $m^{d,U,R}_{ssp}$ and $\sigma^{d,U,R}_{ssp}$.

10. Determine the boundaries of intervals of normal values with the 3-sigma rule for all three characteristics of the traffic:
    $to_1 = m^{d,T,R} - 3 * \sigma^{d,T,R}$, $to_2 = m^{d,T,R} + 3 * \sigma^{d,T,R}$,
    $sm_1 = m^{d,L,R}_{ssp} - 3 * \sigma^{d,L,R}_{ssp}$, $sm_2 = m^{d,L,R}_{ssp} + 3 * \sigma^{d,L,R}_{ssp}$,
    $ge_1 = m^{d,U,R}_{ssp} - 3 * \sigma^{d,U,R}_{ssp}$, $ge_2 = m^{d,U,R}_{ssp} + 3 * \sigma^{d,U,R}_{ssp}$.

11. Build up a final description for the given direction of the traffic as a 4-tuple consisting of the value of selected split-point and of a triple of ranges of normal (expected) values: $(ssp, < to_1; to_2 >, < sm_1; sm_2 >, < ge_1; ge_2 >)$, where $< to_1; to_2 >$ specify the range of normal values of the total amount of the transmitted packets in the five minute window, $< sm_1; sm_2 >$ specify the range of normal values of the amount of packets with $\Delta t < ssp$ and $< ge_1; ge_2 >$ specify the range of the normal values of the amount of the packets with $\Delta t \geq ssp$.

Figures A.1 - A.7 show the collected characteristics for individual datasets (total amount of packets for each five minute window and amounts of packets transmitted in two ranges of their inter-arrival time). Table 4.6 shows the descriptions of the traffic for individual datasets that we build up for the validation purposes. Each traffic is described by a couple of description defined previously - one for each direction. In order to define the ranges of normal values for individual characteristics we utilize 3-sigma rule. We also tested the ranges of normal values based on IQR, but these intervals proved to be too narrow to accept characteristics of normal communication flow.

4.2.1 Traffic monitoring and anomaly detection

Traffic monitoring system based on our statistical method has to gather the statistical information for individual five minute windows. For each five minute window, it is necessary to collect information about the number of

transmitted packets and about their inter-arrival times. Traffic in each direction (*from master* and *to master*) is monitored separately, however inter-arrival time should be determined in both-direction traffic. The descriptions for both directions of the given traffic contain all information essential for such traffic monitoring. First component of each description determines how to divide the packets according to their inter-arrival time into two groups. Then, the amount of packets in each group and also total amount of transmitted packets are compared with the ranges of normal values contained in the description. If any of the values does not fit into the specified range, an anomaly is detected. This is the principle of the simple-detection method.

Besides this simple-detection method, monitoring system can also utilize 3-value-detection method. The purpose of this method is to allow short anomalies after them the traffic properties returns back to normal values and to reduce the number of false positives window. In this method, we consider three consecutive five minute windows. An anomaly is reported only if at least two of the three windows detect the values outside the specified range for some characteristic of the traffic.

Dataset	Dir.	Description
10122018-104Mega	fm	(0.10,<1200.78;1413.44>,<1104.74;1311.42>,<79.51;119.30>)
	tm	(0.40,<367.10;427.92>,<314.46;375.33>,<47.91;54.49>)
13122018-mega104	fm	(0.09,<1260.54;1357.95>,<1171.15;1270.40>,<45.17;131.89>)
	tm	(0.46,<390.15;411.42>,<336.42;359.35>,<46.27;59.52>)
mega104-14-12-18	fm	(5.28,<17.74;82.24>,<0.27;72.10>,<4.82;22.70>)
	tm	(1.01,<19.39;26.28>,<-1.43;12.03>,<11.09;23.98>)
mega104-17-12-18	fm	(5.66,<20.30;76.94>,<4.70;68.05>,<3.90;20.62>)
	tm	(1.01,<19.39;26.22>,<-1.01;12.26>,<10.67;23.64>)
KTH-RTU8	fm	(0.186,<1329.00;1858.74>,<86.53;524.54>,<1230.18;1346.55>)
	tm	(0.014,<147.58;206.59>,<10.04;58.02>,<92.85;193.18>)
KTH-RTU11	fm	(0.211,<368.79;1302.63>,<-231.92;644.10>,<541.62;713.65>)
	tm	(0.006,<40.97;144.76>,<9.69;36.95>,<11.34;127.58>)
RICS	fm	(1.35,<169.52;248.56>,<114.27;191.58>,<50.29;62.05>)
	tm	(0.14,<24.56;33.61>,<14.25;29.47>,<0.84;13.56>)

Table 4.6: Description of the normal traffic in our datasets.

4.2.2 GOOSE protocol specifics

The proposed method is suitable also for other ICS communication protocols. In this section, we describe the application of the method on GOOSE communication. In this case, we focus on modelling flow properties rather than packets or virtual flows as in case of IEC 104. When analyzing GOOSE flows we need to re-define the inter-arrival time. Since we are able to record

the start time and the end times of each flows, we can define the inter-arrival time of individual flows as the difference between timestamps of the subsequent flows. With this abstraction, the inter-arrival time includes the duration of the flow. However, the duration of flows is constant, so we can determine the inter-arrival time in this simple way.

	fe80::221:c1ff:fe25:8a2	fe80::209:8eff:fefa:c045
Q1	69.3031	60.0071
Q2	69.3032	60.0080
mean	69.3032	60.0148
Q3	69.3033	60.0089

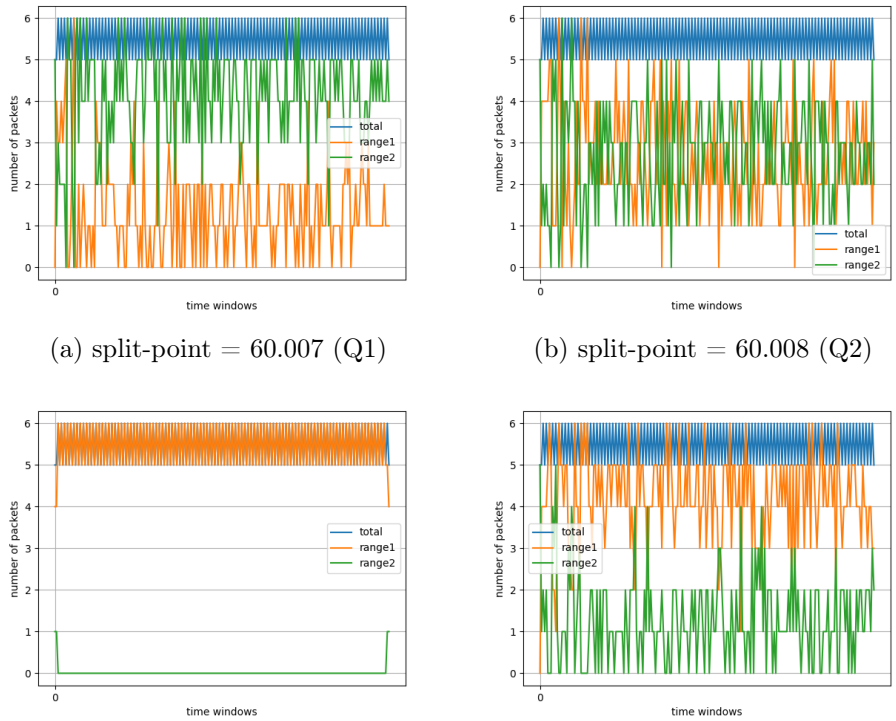
Table 4.7: Inter-arrival time (of flows) distribution for two publishers from goose-mms3 dataset (rounded to 4 decimal places).

The communication in GOOSE protocol differs from communication in IEC104 in many ways. These differences greatly simplify the modelling process:

- *Publisher-subscriber mode* of GOOSE communication allows to skip the second step (splitting the communication based on the direction) of the modelling process and leads to the model for one-sided communication.
- *Regular inter-arrival times* enable to omit the additional characteristics obtained by splitting flows based on selected split-point. The inter-arrival times in the observed datasets are almost stable and unchangeable for the given flow. Therefore all quartiles and mean value vary in thousandths of a second (see table 4.7). Even though we can apply proposed method and find the suitable split-point for the given communication, such split-point produce rather random split of the individual flows (see fig. 4.7). Furthermore, the main characteristic is so stable that additional characteristics no longer provide any benefit. Altogether, we suggest to model the GOOSE communication only with the main characteristic (total number of flows captured in the specified time window) and its range of normal values (given by $mean \pm 3 * \sigma$).

On the other hand, the high stability of the inter-arrival times of GOOSE flows brings the problem of zero standard deviation of the gathered characteristic (see fig. 4.8a). This leads to too narrow range of normal values that is very sensitive to even very small changes in the flow distribution. This problem arises if the size of the time window is almost infinitely divisible by the average inter-arrival time. Time windows with the size disjoint with average inter-arrival time provide more robust range of normal values for the given traffic (see fig. 4.8b).

While inter-arrival times of flows in Goose communication seems to be too stable to refine the statistical model of the given communication, other



(a) split-point = 60.007 (Q1)

(b) split-point = 60.008 (Q2)

(c) split-point = 60.015 (mean)

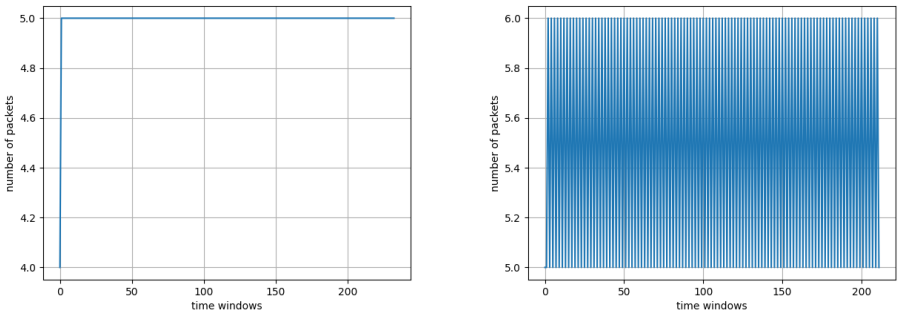
(d) split-point = 60.009 (Q3)

Figure 4.7: Graph of the amount of packets transmitted in five minute windows in `gics-goose` dataset for publisher fe80::209:8eff:fefa:c045 (blue line), additional characteristics (green and orange) show the effect of using different split-points.

flows properties allows to extend the statistical model. Our analysis revealed that the size of the flow expressed in packets or bytes almost does not change over time. As a result, this property can be used for more robust statistical modeling of Goose traffic. In addition to the characteristic indicating the number of flows in a time window, an interval of normal values for the volume of flows (in packets or bytes) transmitted in a time window can also be determined. Such extension allows detection of flows whose size is changed, although the frequency of sending them remain unchanged (see fig. 4.9).

4.3 Experiments

Performed experiments were designed in order to test the applicability of statistical methods (especially 3-sigma method) for anomaly detection in industrial communication. These experiments can be divided into two groups



(a) window size 300s (b) window size 330s

Figure 4.8: Effect of time window size in `gics-goose` dataset for publisher fe80::209:8eff:fefa:c045. The median of the inter-arrival times in this communication is 60.008 seconds.

- validation tests and anomaly detection experiments.

At first, we examined proposed description of the communication flow. We tested the stability of this description throughout our datasets. For these validation tests we divided the available datasets into two parts. We used the first part (2/3 of the captured traffic) to find the description of the given communication flow. Then, we tested established description and its ranges on the second part (1/3 of the flow) of the given communication flow. We tested both detection methods described in section 4.2.1. We searched for detection method that would lead to almost perfect fit of the second part of the communication flow into the given ranges. Such method will not produce the false positives during the detection. On the other hand, this method must also be able to detect possible attacks on the given infrastructure.

The anomaly detection experiments focus on the ability to detect an attack on the given infrastructure. We use the description of the communication flow and the detection method selected by the previous experiments.

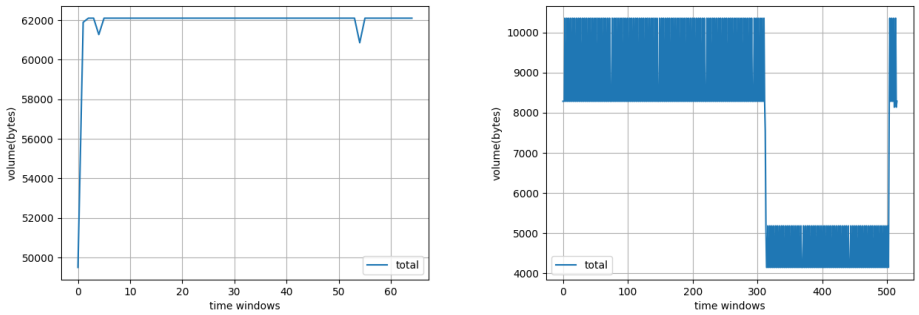
Subsections 4.3.1 and 4.3.2 describe performed experiments for IEC104 protocol and obtained results.

We also tested the proposed description of the communication flow with LOF method. Obtained results are provided in section 4.3.3.

4.3.1 3-sigma validation tests

Validation tests were performed to confirm the applicability of statistical methods and correctness of defined ranges for individual features (characteristics) that we use to describe the communication flow.

In order to find the range of expected values for each feature, we analyze the first two thirds of the communication flow for all datasets. For each flow,



(a) Stable flows volume (b) Unstable flows volume

Figure 4.9: Volume of flows characteristic in `mms3-goose` dataset. The left part shows the usual goose communication, while the right part shows the communication containing the anomaly that changed the volume of the flows in the specified time window (300 seconds).

we gather the statistics (features) as described in section 4.2 and build up its description.

Validation of the determined ranges is performed on the last third of each communication flow. Table 4.8 list all five minute windows in our datasets, their characteristics does not fit into determined ranges of expected values. This simple-detection method marks many five minute windows as anomalies, even if only on the basis of one feature. However, we can see that the characteristics of the communication flow usually return back to specified interval in next five minute window. This small deviations from determined range of normal values can be caused by switching devices or by some network delay. Such behaviour is quite common in normal traffic (see figure A.1).

There exist more possibilities, how to solve this problem. One possibility is to enlarge the determined range of expected normal values. However, this solution may allow too large range of normal values to prevent anomaly detection. Another approach utilizes floating window that allows to test if the values return back to predefined range. Therefore, we propose detection method the utilizes 3-values window, where 2 of three values have to fit into determined range of expected values.

Table 4.9 list five minutes windows detected by 3-value-detection method as anomalies. 3-value-detection method produces much less false positives elements and therefore is more suitable for anomaly detection. Tables 4.10 and 4.11 shows the accuracy of both detection methods. Table 4.10 shows the accuracy of the individual features separately. Table 4.11 summarizes the overall accuracy.

Dataset	Dir.	Char.	List of windows
10122018-104Mega	tm	$\Delta t \geq sp$	40
13122018-mega104	fm	total	618, 725
		$\Delta t < sp$	618, 725
13122018-mega104	tm	total	618, 725
		$\Delta t < sp$	618, 725
mega104-17-12-18	fm	total	784
		$\Delta t < sp$	784
		$\Delta t \geq sp$	654, 754
mega104-17-12-18	tm	$\Delta t < sp$	795
		$\Delta t \geq sp$	795
KTH-RTU8	fm	$\Delta t \geq sp$	1492, 1497, 1527 - 1529, 1812 - 1813, 1818, 1924
KTH-RTU11	fm	$\Delta t \geq sp$	1417, 1421, 1429, 1448, 1456, 1460, 1463, 1467, 1482, 1650, 1745, 1749, 1932, 1940
	tm	$\Delta t < sp$	1482, 1632, 1833
RICS	fm	total	2564, 2609, 3458, 3498
		$\Delta t < sp$	2609, 3458, 3498
		$\Delta t \geq sp$	2543, 2555, 2574, 2682, 2841, 2849, 2850, 2860, 2942, 3130, 3140, 3141, 3160, 3237, 3263, 3420, 3421, 3423, 3426, 3429, 3431, 3474, 3707, 3712, 3713
RICS	tm	total	3716
		$\Delta t < sp$	2543, 2578, 3140, 3207, 3420, 3498, 3716
		$\Delta t \geq sp$	2543, 2553, 2575, 2869, 3001, 3138, 3140, 3716

Table 4.8: Simple-detection method validation - list of five minute windows that does not fit into predefined range of values. Only datasets and characteristics with false positive elements are included.

Dataset	Direction	Char.	List of windows
KTH-RTU8	fm	$\Delta t \geq sp$	1526 - 1528, 1811 - 1812
RICS	fm	$\Delta t \geq sp$	2848 - 2849, 3139 - 3140, 3419 - 3421, 3429, 3711 - 3712
RICS	tm	$\Delta t \geq sp$	3138

Table 4.9: 3-value-detection method validation - list of first five minute window of 3-value window for which 2 of the three values do not fit into predefined range of values. Start position of 3-value window is reported. Only datasets and characteristics with false positive elements are included.

			Simple-detection		**3-value-detection**	
Dataset	Dir.	Char.	FP/all	Acc	FP/all	Acc
10122018-104Mega	fm	any	0/20	100%	0/20	100%
10122018-104Mega	tm	$\Delta t \geq sp$	1/20	95%	0/20	100%
13122018-mega104	fm	total	2/285	99,30%	0/285	100%
		$\Delta t < sp$	2/285	99,30%	0/285	100%
13122018-mega104	tm	total	2/285	99,30%	0/285	100%
		$\Delta t < sp$	2/285	99,30%	0/285	100%
mega104-14-12-18	fm	any	0/63	100%	0/63	100%
mega104-14-12-18	tm	any	0/63	100%	0/63	100%
mega104-17-12-18	fm	total	1/273	99.63%	0/273	100%
		$\Delta t < sp$	1/273	99.63%	0/273	100%
		$\Delta t \geq sp$	2/273	99.27%	0/273	100%
mega104-17-12-18	tm	$\Delta t < sp$	1/273	99.63%	0/273	100%
		$\Delta t \geq sp$	1/273	99.63%	0/273	100%
KTH-RTU8	fm	$\Delta t \geq sp$	9/650	98.62%	5/650	99.23%
KTH-RTU8	tm	any	0/650	100%	0/650	100%
KTH-RTU11	fm	$\Delta t \geq sp$	14/650	97.85%	0/650	100%
KTH-RTU11	tm	$\Delta t < sp$	3/650	99.54%	0/650	100%
RICS	fm	total	4/1240	99.68%	0/1240	100%
		$\Delta t < sp$	3/1240	99.76%	0/1240	100%
		$\Delta t \geq sp$	25/1240	97.98%	10/1240	99.19%
RICS	tm	total	1/1240	99.92%	0/1240	100%
		$\Delta t < sp$	7/1240	99.44%	0/1240	100%
		$\Delta t \geq sp$	8/1240	99.34%	1/1240	99.92%

Table 4.10: Validation results - results for individual characteristics.

Dataset	Simple-detection FP/all	Simple-detection Acc	3-value-detection FP/all	3-value-detection Acc
10122018-104Mega	1/20	95%	0/20	100%
13122018-mega104	2/285	99.30%	0/285	100%
mega104-14-12-18	0/63	100%	0/63	100%
mega104-17-12-18	4/273	98.53%	0/273	100%
KTH-RTU8	9/650	98.62%	5/650	99.23%
KTH-RTU11	16/650	97.54%	0/650	100%
RICS	37/1240	97.02%	11/1240	99.11%

Table 4.11: Validation - summary results.

4.3.2 3-sigma anomaly detection

In the previous section, we show that our description of the traffic together with 3-value-detection method is able to describe the normal behaviour of the given communication flow with sufficient accuracy. In this section, we describe the results obtained by processing datasets with simulated attacks. Figures A.8 - A.13 show the collected characteristics for `mega104-17-12-18` datasets with simulated attacks (total amount of packets for each five minute window and amounts of packets transmitted in two ranges of their inter-arrival time).

For these tests, we build up a description of the traffic from the whole `mega104-17-12-18` dataset. For the anomaly detection we applied 3-value-detection method. Tables 4.12, 4.13 and 4.14 list the 5 minute windows that were revealed as anomalies.

The results show that the method is able to detect the majority of simulated attacks. Our method does not correctly recognize only one attack - first injection attack. This attack does not involve the amount of transmitted packets significantly and therefore it is not detectable by this method. Other attacks were correctly detected at least in one direction of the communication flow.

4.3.3 LOF validation and anomaly detection

We applied the LOF novelty detection method to the same IEC104 traffic description as our 3-sigma method. We applied LOF method on data points with three attributes: total amount of packets for each five minute window and amounts of packets transmitted in two ranges of their inter-arrival time. On the final labeling of the LOF method we again applied 3-value-detection method to filter out short deviation in the traffic. We search for such 3 consecutive values where at least two of them were labeled as outliers by LOF method. Both direction of the traffic we treated separately.

Table 4.15 shows the best value of *n_neighbors* parameter, the number

Dir.	Char.	Connection loss		Injection attack	
		310-312	498-510	59-60	365-368
fm	total	311-312 ↓	499-511 ↓	-	-
	$\Delta t < sp$	-	500-510 ↓	-	-
	$\Delta t \geq sp$	-	499-510 ↓	-	-
tm	total	311-313 ↓	499-511 ↓	-	367-369 ↓
	$\Delta t < sp$	-	-	-	-
	$\Delta t \geq sp$	311-312 ↓	499-511 ↓	-	-

Table 4.12: Connection loss and injection attack detection. The header of the columns list the real five minute windows in which the attack occurred, lines denote windows in which the attack was detected by individual characteristics. Arrows indicate whether the amount of packets was above or below the range of specified values.

Dir.	Char.	DoS attack		Roque device
		110-128	142-161	8-13
fm	total	-	-	10-14↓
	$\Delta t < sp$	-	-	10-14 ↓
	$\Delta t \geq sp$	112-114, 117-121, 125-128 ↑	145-161 ↑	10-14 ↓
tm	total	111-130 ↓	143-162 ↓	9-14 ↓
	$\Delta t < sp$	-	-	-
	$\Delta t \geq sp$	111-129 ↓	143-162 ↓	9-14 ↓

Table 4.13: DoS attack and roque device detection. The header of the columns list the real five minute windows in which the attack occurred, lines denote windows in which the attack was detected by individual characteristics. Arrows indicate whether the amount of packets was above or below the range of specified values.

of false positive windows and the accuracy for each IEC104 dataset. The results of LOF and 3-sigma method are roughly comparable, however LOF method produces slightly more false positive windows and it requires finding a suitable *n_ neighbors* parameter value. For some datasets the right value of *n_ neighbors* parameter is crucial, since bad value can result in an increase in the number of false positive windows to more than six times (up to 41 FP for `mega104-17-12-18` dataset).

In addition, we encountered a problem with the LOF method during anomaly detection. In this case LOF method produced many false positive points. We examined the results of the LOF method and found that the method is sensitive to changes in the density of data points. Since our dataset with normal traffic contains many identical points (duplicates), the LOF method identified as outliers such points around which the frequency

Dir.	Char.	Scanning attack		Switching attack
		239-242	413-417	190-192
fm	total	240-242 ↓	-	-
	$\Delta t < sp$	241-242 ↓	-	-
	$\Delta t \geq sp$	240-242 ↓	-	-
tm	total	240-243 ↓	414-417 ↑	191-192 ↑
	$\Delta t < sp$	-	-	191-192 ↑
	$\Delta t \geq sp$	240-243 ↓	-	-

Table 4.14: Scanning and switching attack detection. The header of the columns list the real five minute windows in which the attack occurred, lines denote windows in which the attack was detected by individual characteristics. Arrows indicate whether the amount of packets was above or below the range of specified values.

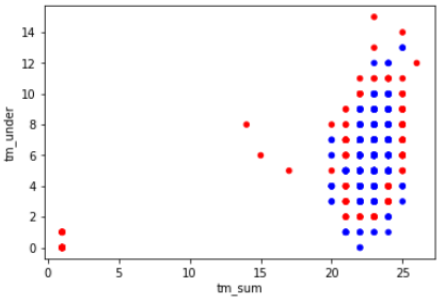
	3-sigma method		**LOF**		
Dataset	FP/all	Acc	FP/all	Acc	best k
10122018-104Mega	0/20	100%	0/20	100%	$k = 28, 29$
13122018-mega104	0/285	100%	4/285	98.60%	$k = 4$
mega104-14-12-18	0/63	100%	0/63	100%	$k = 10...12, 18...20, 29$
mega104-17-12-18	0/273	100%	6/273	97.80%	$k = 6$
KTH-RTU8	5/650	99.23%	0/650	100%	$9 \leq k \leq 11$
KTH-RTU11	0/650	100%	0/650	100%	$k \geq 6$
RICS	11/1240	99.11%	18/1240	98.55%	$k = 3, 23, 24$

Table 4.15: LOF validation results compared to 3-sigma method results.

of occurrence of points decreased significantly (see fig. 4.10). Such labeling is not desired behavior for the traffic monitoring and anomaly detection.

To avoid the problem with duplicates, we filtered them out and performed the validation test The results of these experiments together with the bast values of the parameter *n_neighbors* are summarized in table 4.16. The results of anomaly detection performed on the reduced set with *n_neighbors* parameter set to value 6 are presented in table 4.17.

The ability of LOF method to detect the individual attacks is the same as of 3-sigma method. It is not possible to detect the first injection attack by both methods due to the used description of the communication flow. LOF method on the reduced dataset produces the slightly better result than 3-sigma method in the term of false positive windows. On the other hand, the 3-sigma method employs a simpler model and the evaluation of time windows is extremely clear and fast.



(a) LOF labeling

	19	20	21	22	23	24	25	26
13					1		2	
12					1	3		1
11				1	2	6	1	
10				6	14	16	4	
9			3	14	21	19	9	
8		1	2	14	36	24	9	
7		2	10	31	34	32	6	
6		2	11	46	48	22	11	
5		1	21	43	43	29	3	
4		6	10	31	33	10	3	
3		2	9	30	21	11	1	
2		1	4	6	10	4		
1			6	5	2	1		
0				2				

(b) Frequency of point in the cluster

Figure 4.10: LOF method behavior on the `mega104-17-12-18` dataset with DoS attack and with duplicates (*to master* direction). As outliers (red points) are marked points around which the frequency of occurrence of points decreased significantly. The frequency of the points in the cluster is depicted in (b).

	LOF		
Dataset	FP	Acc	best k
10122018-104Mega	0	100%	$k \geq 9$
13122018-mega104	0	100%	$k \geq 21$
mega104-14-12-18	0	100%	$\forall k$
mega104-17-12-18	0	100%	$\forall k$
KTH-RTU8	0	100%	$9 \leq k \leq 24$
KTH-RTU11	0	100%	$k \geq 6$
RICS	0	100%	$\forall k$

Table 4.16: LOF validation results on reduced dataset (without duplicates).

Attack	3-sigma		LOF	
	TM	FM	TM	FM
Connection loss 1	✓	✓	✓	✓
Connection loss 2	✓	✓	✓	✓
Injection attack 1	×	×	×	×
Injection attack 2	×	✓	×	✓
DoS attack 1	✓	✓	✓	✓
DoS attack 2	✓	✓	✓	✓
Roque Device	✓	✓	✓	✓
Scanning attack 1	✓	✓	✓	✓
Scanning attack 2	×	✓	×	✓
Switching attack	×	✓	×	✓

Table 4.17: Comparison of the anomaly detection with 3-sigma method and LOF method.

4.3.4 Summary

Performed experiments show that the total amount of packets transmitted in five minute window is useful feature of the communication flow. In order to provide more subtle description it is possible to divide the amount of transmitted packets into groups defined by Δt_i intervals.

3-sigma rule can be used to define the ranges of normal values for the designed features. 3-value-detection method is capable to detect almost all simulated attack while produce only reasonably small number of false positive windows.

On the other hand, in some datasets, we can see, that the range of expected values is quite wide. Usually, this is true for communication flows that show some periodicity. In such case, more precise method based on the detected period might be more appropriate.

Chapter 5

Conclusion

Bibliography

[1] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A Survey of Network Anomaly Detection Techniques. *J. Netw. Comput. Appl.*, 60(C):19–31, January 2016.

[2] R. R. R. Barbosa, R. Sadre, and A. Pras. A first look into SCADA network traffic. In *2012 IEEE Network Operations and Management Symposium*, pages 518–521, April 2012.

[3] R. R. R. Barbosa, R. Sadre, and A. Pras. Towards periodicity based anomaly detection in SCADA networks. In *Proceedings of IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–4, Sept 2012.

[4] Rafael R. R. Barbosa, Ramin Sadre, and Aiko Pras. Difficulties in Modeling SCADA Traffic: A Comparative Analysis. In *The 13th International Conference on Passive and Active Measurement*, pages 126–135, 2012.

[5] Smart Grid Cybersecurity Committee. Guidelines for Smart Grid Cybersecurity. Technical Report NISTIR-7628r1, National Institute of Standards and Technology, 2014.

[6] Manuel Crotti, Maurizio Dusi, Francesco Gringoli, and Luca Salgarelli. Traffic Classification Through Simple Statistical Fingerprinting. *SIGCOMM Comput. Commun. Rev.*, 37(1):5–16, January 2007.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[8] Dragos. CrashOverride. Analysis of the Threat of Electric Grid Operations. Technical report, Dragos Inc., June 2017.

[9] Gilberto Fernandes, J. Rodrigues, L. F. Carvalho, J. Al-Muhtadi, and M. L. Proença. A comprehensive survey on network anomaly detection. *Telecommunication Systems*, 70:447–489, 2019.

[10] David Formby, Anwar Walid, and Raheem Beyah. A case study in power substation network dynamics. 1(1), June 2017.

[11] Kevin E. Hemsley and Dr. Ronald E. Fisher. History of Industrial Control System Cyber Incidents. (INL/CON-18-44411-Revision-2), 12 2018.

[12] Eric D. Knapp and Joel Thomas Langill. *Industrial Network Security. Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems.* Syngress, 2015.

[13] Chih-Yuan Lin and Simin Nadjm-Tehrani. Understanding IEC-60870-5-104 Traffic Patterns in SCADA Networks. In *The 4th ACM Workshop on Cyber-Physical System Security*, CPSS '18, pages 51–60, 2018.

[14] Chih-Yuan Lin, Simin Nadjm-Tehrani, and Mikael Asplund. Timing-based anomaly detection in SCADA networks. In *International Conference on Critical Information Infrastructures Security*, pages 48–59. Springer, 2017.

[15] Petr Matoušek. Description and analysis of IEC 104 Protocol. Technical Report FIT-TR-2017-12, Brno University of Technology, 2017.

[16] Petr Matoušek. Description of IEC 61850 Communication. Technical Report FIT-TR-2018-01, Brno University of Technology, 2018.

[17] Petr Matoušek, Ondřej Ryšavý, and Matěj Grégr. Security Monitoring of IoT Communication Using Flows. In *Proceedings of the 6th Conference on the Engineering of Computer Based Systems*, ECBS '19, pages 1–9. Association for Computing Machinery, 2019.

[18] Petr Matoušek, Ondřej Ryšavý, Matěj Grégr, and Vojtěch Havlena. Flow based monitoring of ICS communication in the smart grid. *Journal of Information Security and Applications*, 54:102535, 2020.

[19] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers.* John Wiley and Sons, 7th edition, 2018.

[20] Friedrich Pukelsheim. The Three Sigma Rule. *The American Statistician*, 48(2):88–91, 1994.

[21] Z. Shelby, K. Hartke, and C. Bromann. *The Constrained Application Protocol (CoAP).* IETF RFC 7252, June 2014.

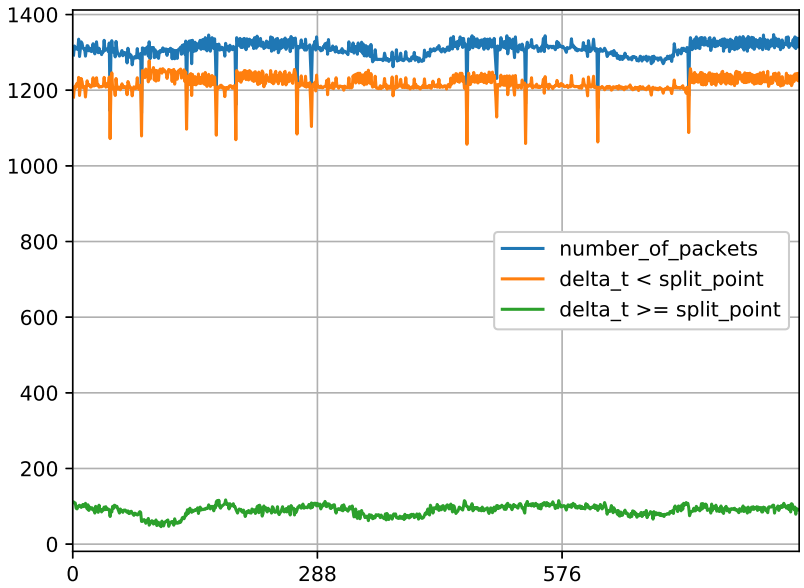
[22] Keith Stouffer, Victoria Pillitteri, Marshall Abrams, and Adam Hahn. Guide to Industrial Control Systems (ICS) Security. Technical Report NIST-SP-800-82r2, National Institute of Standards and Technology, 2015.

[23] John W. Tukey. *Exploratory data analysis.* Addison-Wesley, 1977.

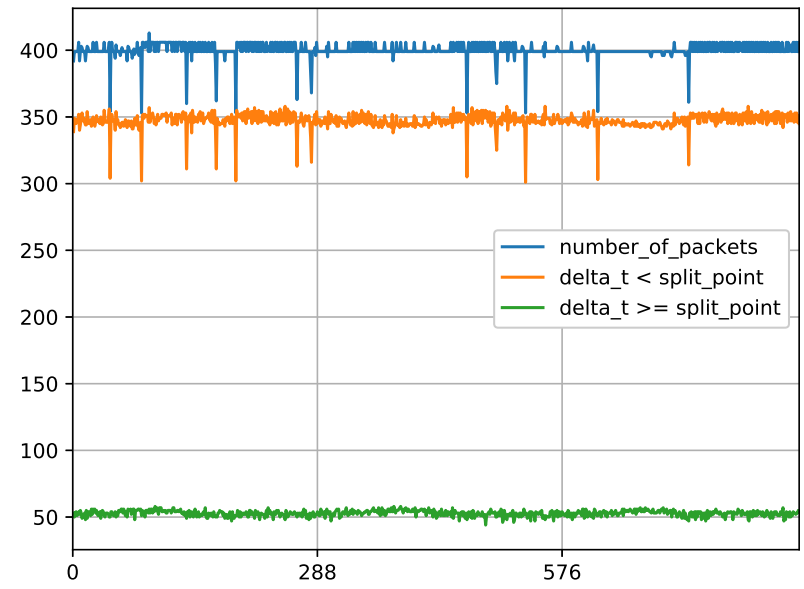
[24] A. Valdes and S. Cheung. Communication pattern anomaly detection in process control systems. In *2009 IEEE Conference on Technologies for Homeland Security*, pages 22–29, May 2009.

Appendix A

Figures with automatic split-points

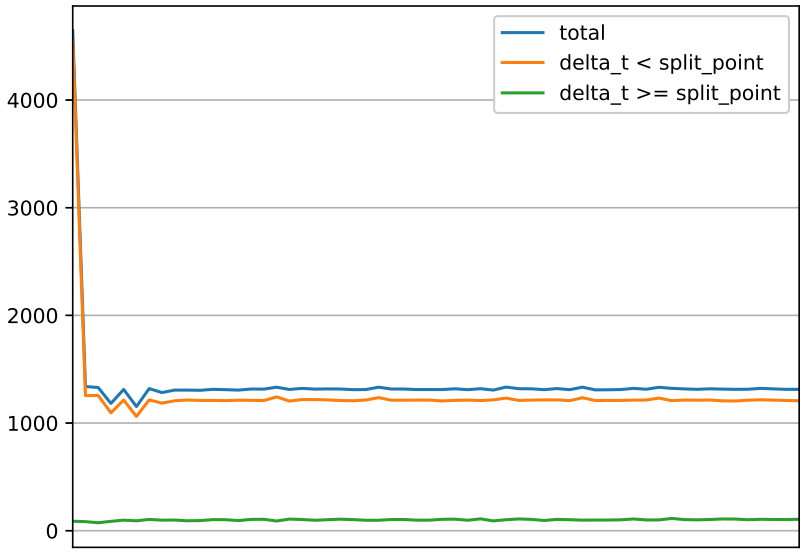


(a) Characteristics of the traffic from the master device.

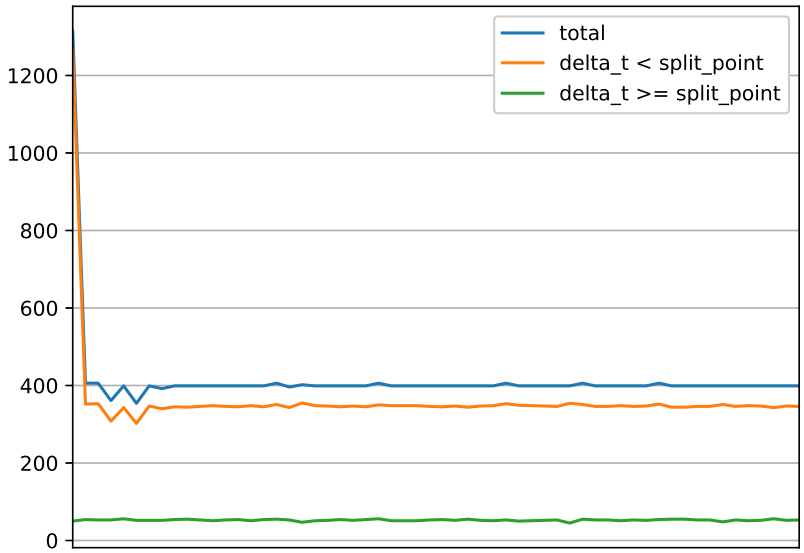


(b) Characteristics of the traffic to the master device.

Figure A.1: Graphs of the amounts of transmitted packets in five minute windows in `13122018-mega104` dataset (whole communication splitted by the direction, automatic setting of split points).

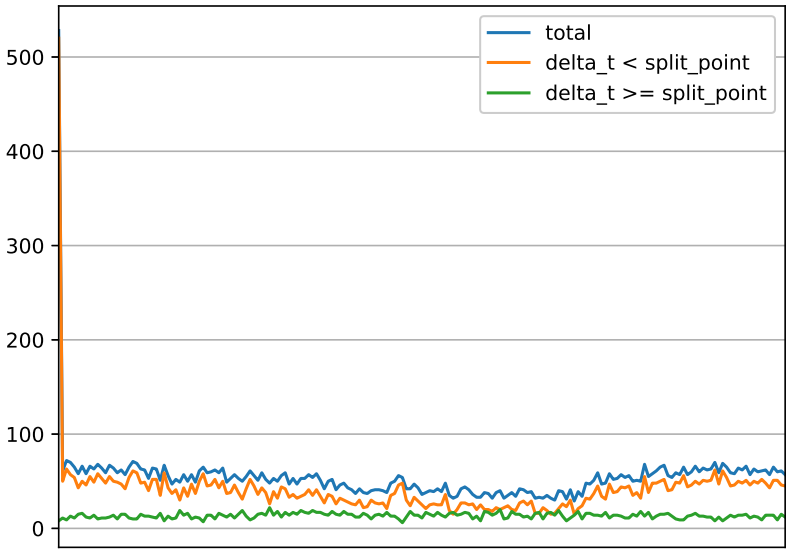


(a) Characteristics of the traffic from the master device.

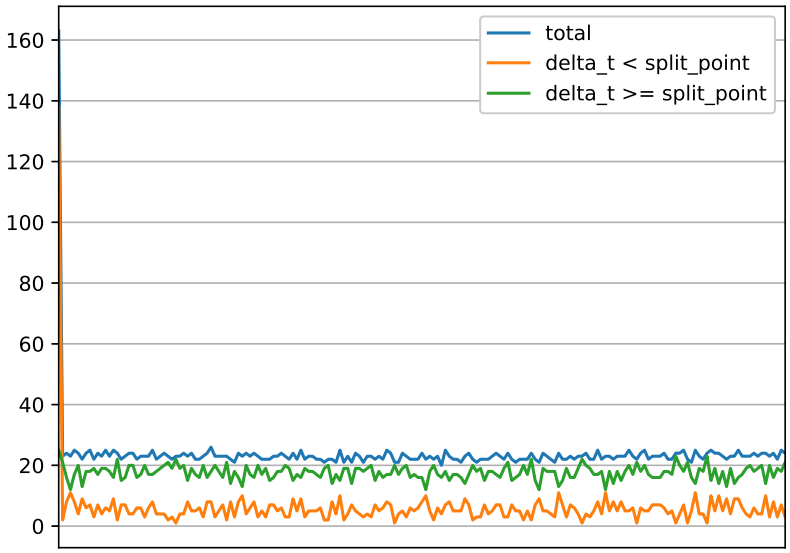


(b) Characteristics of the traffic to the master device.

Figure A.2: Graphs of the amounts of transmitted packets in five minute windows in `10122018-104Mega` dataset (whole communication splitted by the direction, automatic setting of split points).

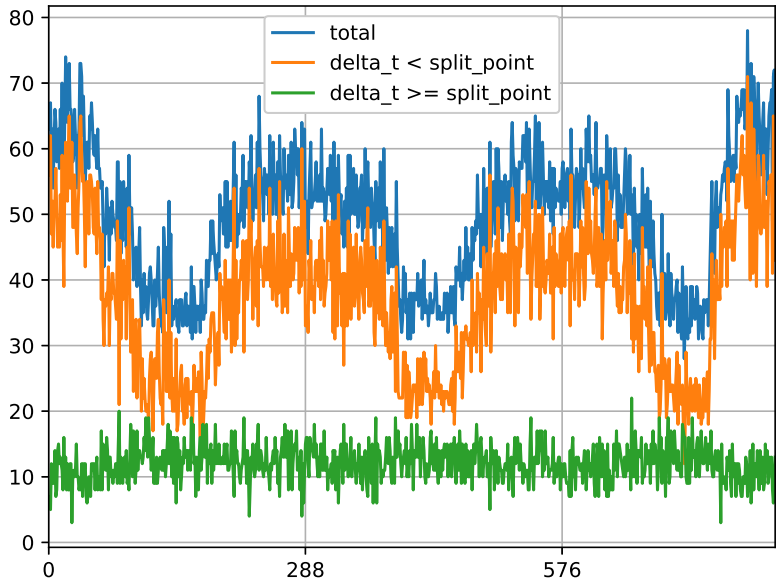


(a) Characteristics of the traffic from the master device.

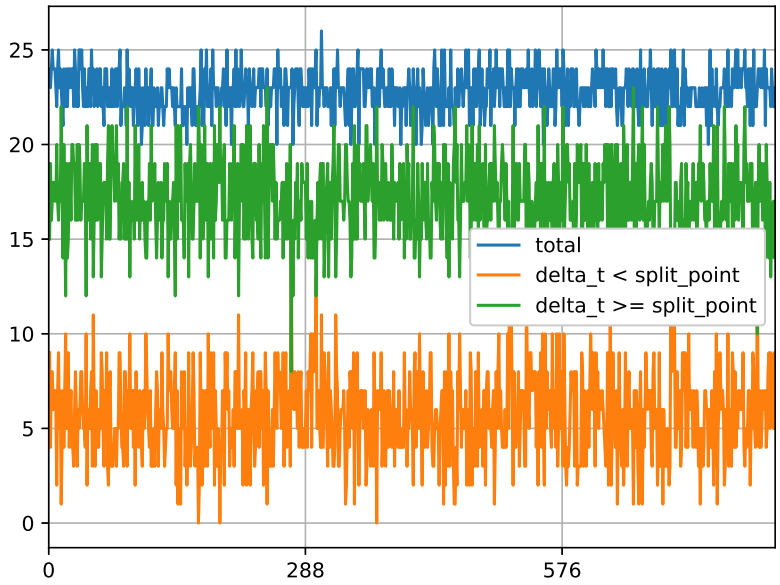


(b) Characteristics of the traffic to the master device.

Figure A.3: Graphs of the amounts of transmitted packets in five minute windows in `mega104-14-12-18` dataset (whole communication splitted by the direction, automatic setting of split points).

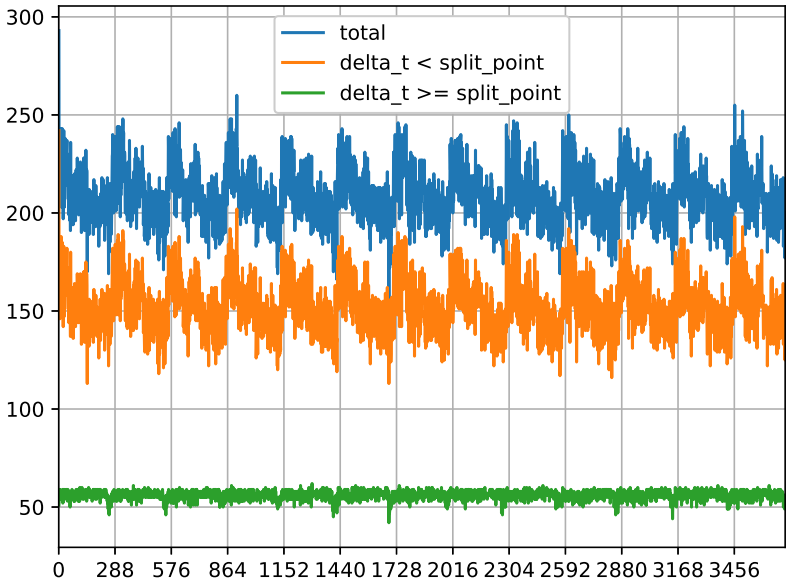


(a) Characteristics of the traffic from the master device.

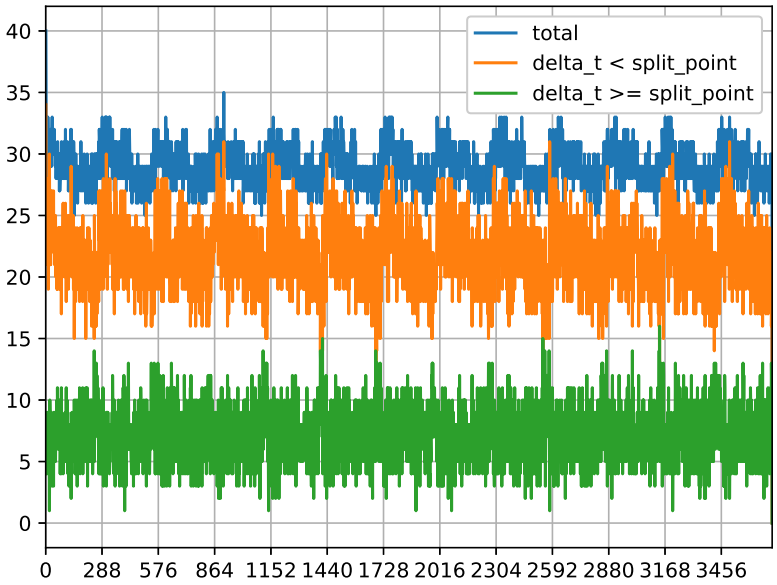


(b) Characteristics of the traffic to the master device.

Figure A.4: Graphs of the amounts of transmitted packets in five minute windows in `mega104-17-12-18` dataset (whole communication splitted by the direction, automatic setting of split points).

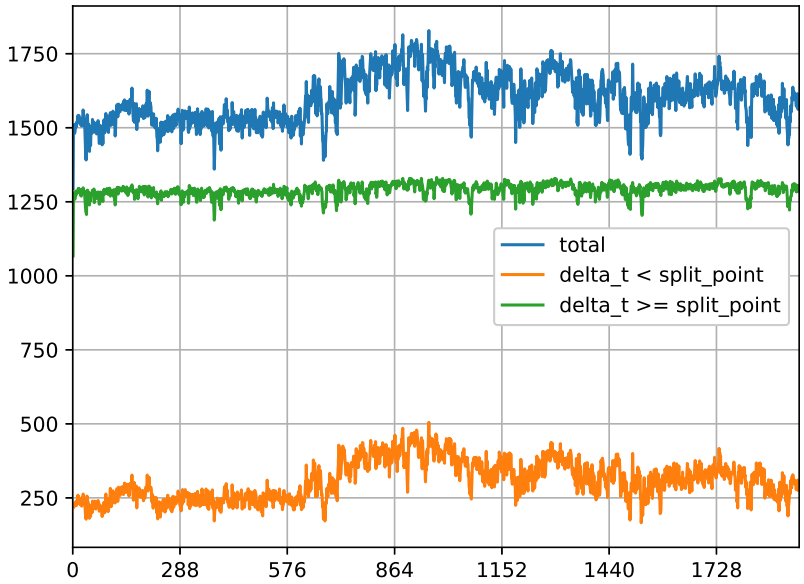


(a) Characteristics of the traffic from the master device.

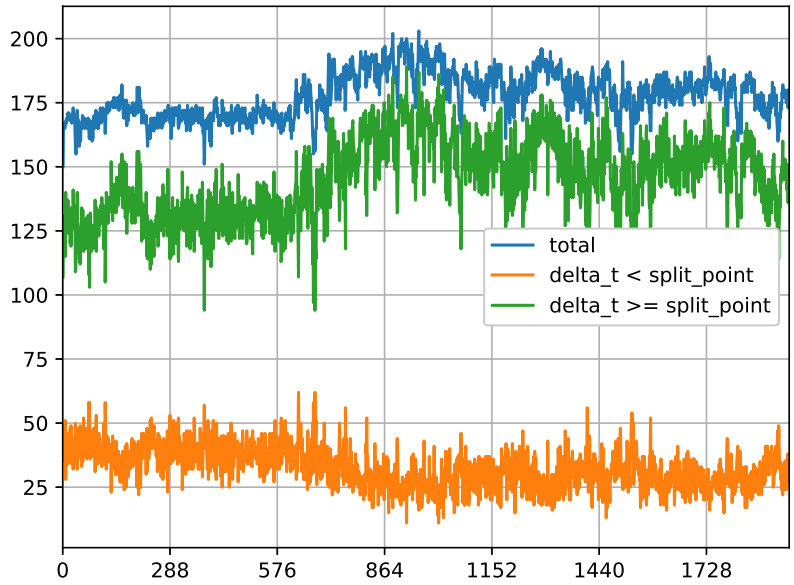


(b) Characteristics of the traffic to the master device.

Figure A.5: Graphs of the amounts of transmitted packets in five minute windows in `RICS` dataset (whole communication splitted by the direction, automatic setting of split points).

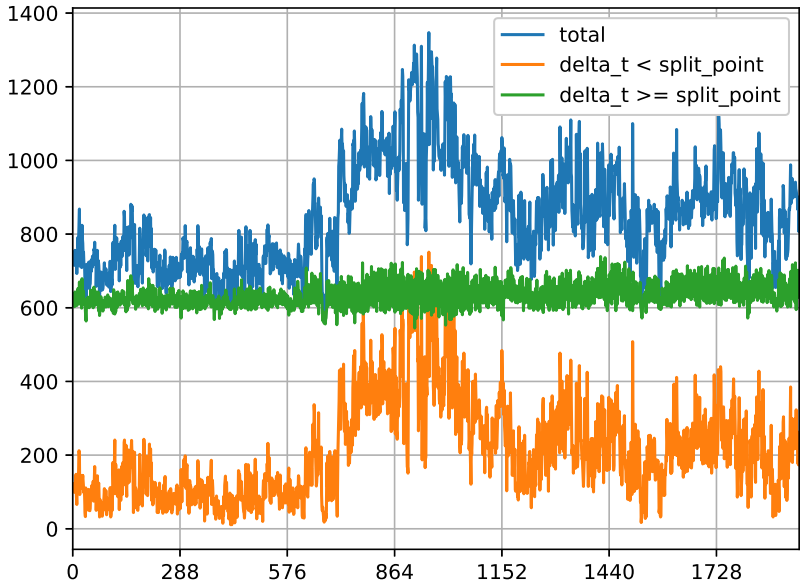


(a) Characteristics of the traffic from the master device.

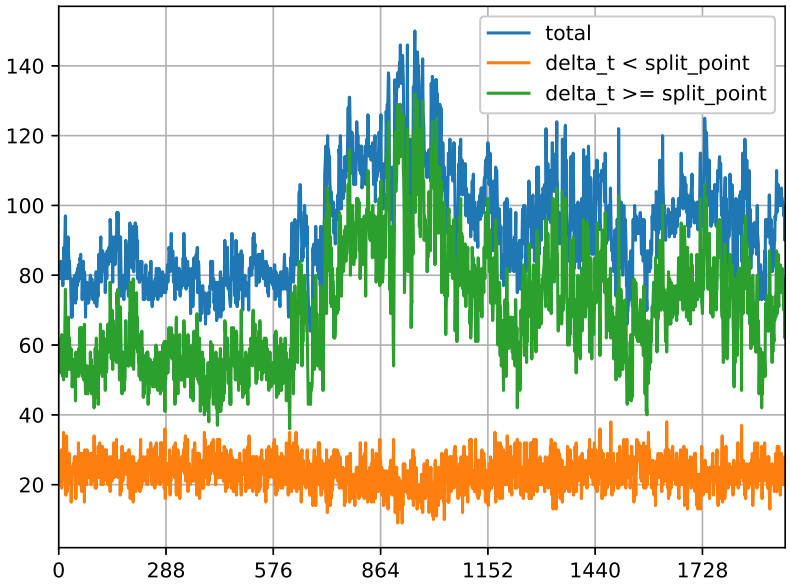


(b) Characteristics of the traffic to the master device.

Figure A.6: Graphs of the amounts of transmitted packets in five minute windows in `KTH-RTU8` dataset (whole communication splitted by the direction, automatic setting of split points).

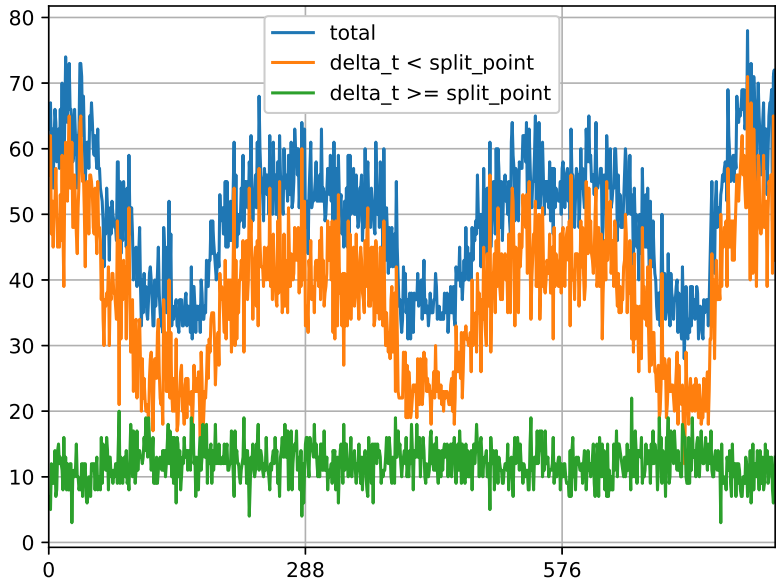


(a) Characteristics of the traffic from the master device.

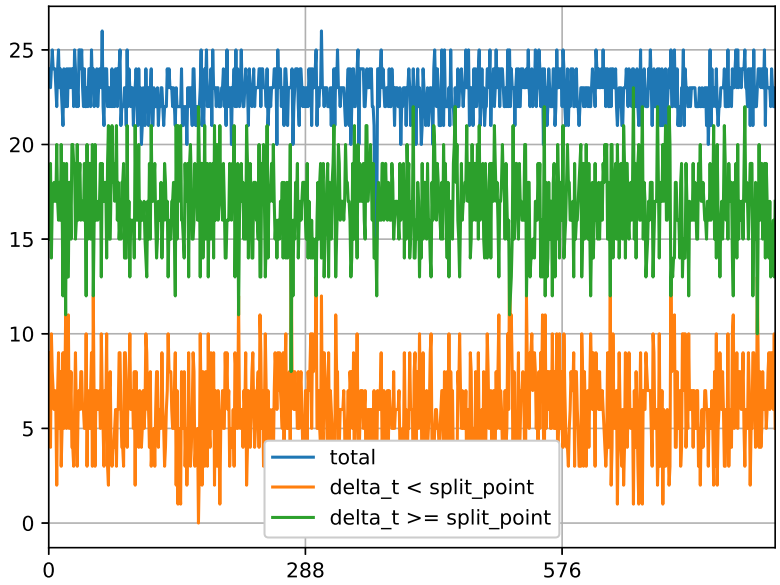


(b) Characteristics of the traffic to the master device.

Figure A.7: Graphs of the amounts of transmitted packets in five minute windows in `KTH-RTU11` dataset (whole communication splitted by the direction, automatic setting of split points).

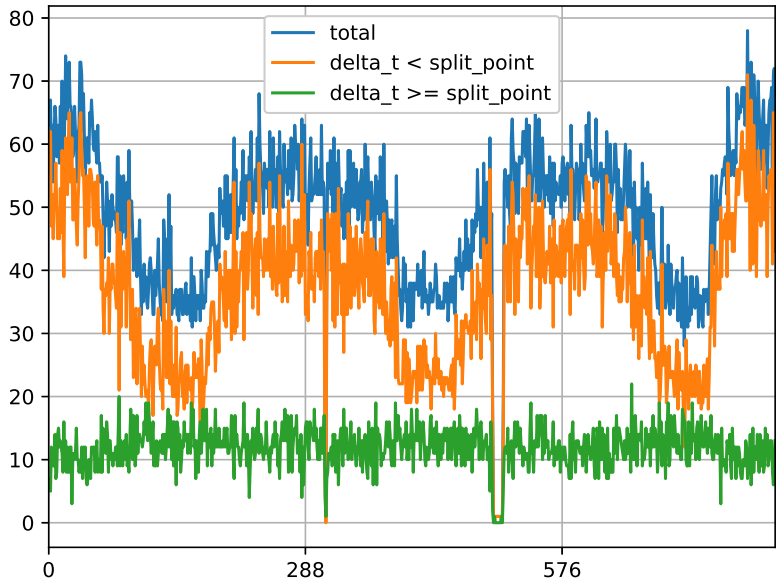


(a) Characteristics of the traffic from the master device.

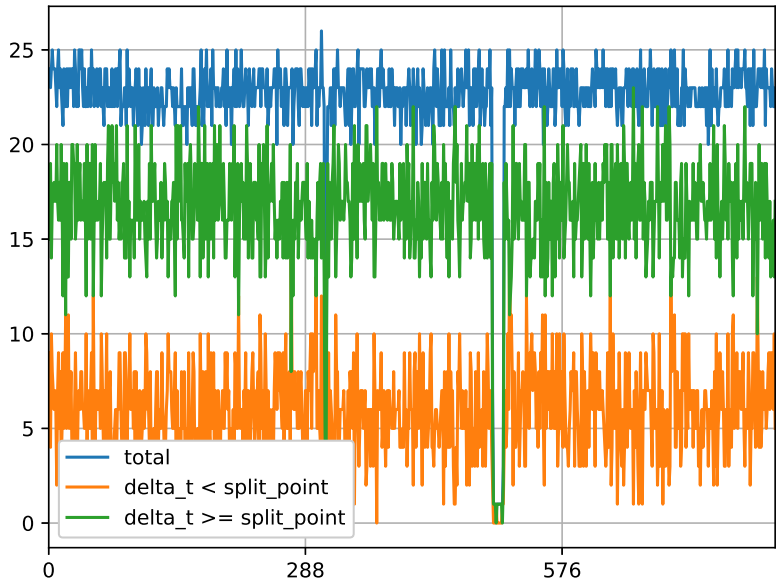


(b) Characteristics of the traffic to the master device.

Figure A.8: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with injection attack (whole communication splitted by the direction, automatic setting of split points).

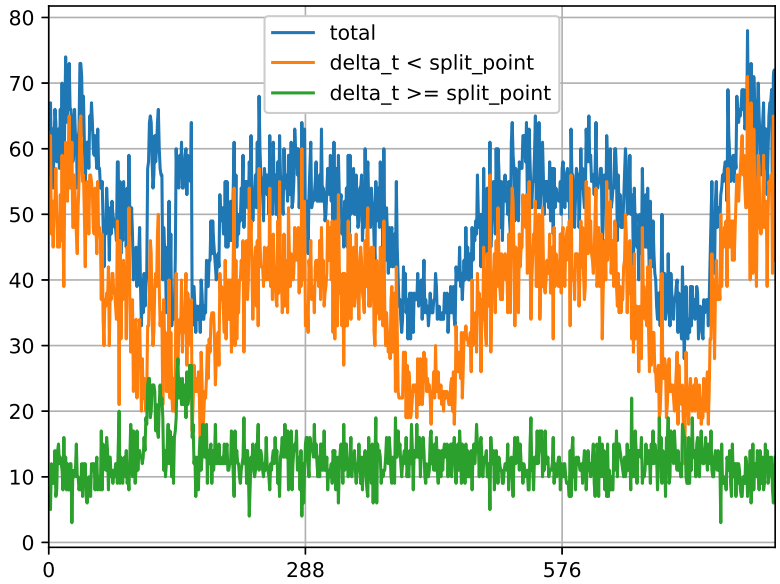


(a) Characteristics of the traffic from the master device.

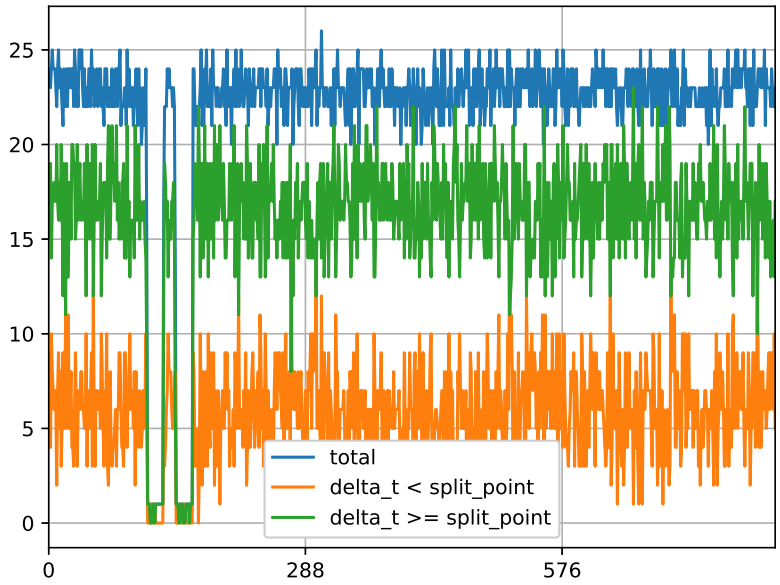


(b) Characteristics of the traffic to the master device.

Figure A.9: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with connection loss attack (whole communication splitted by the direction, automatic setting of split points).

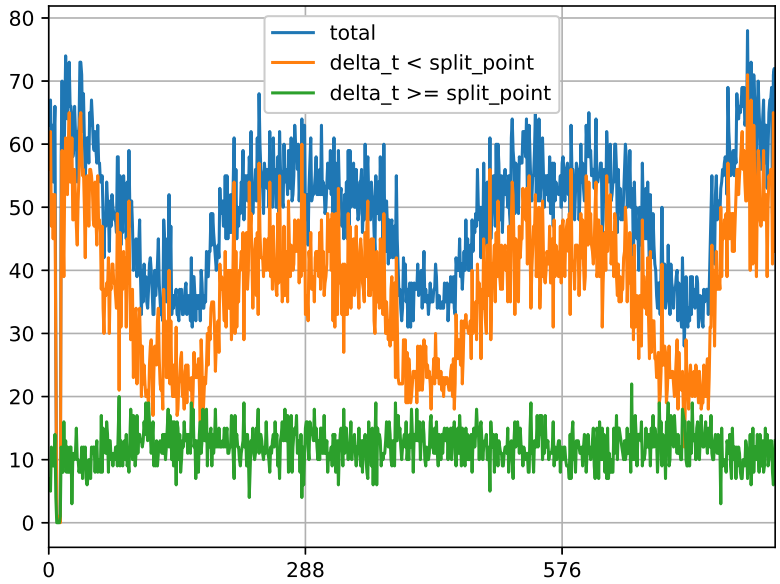


(a) Characteristics of the traffic from the master device.

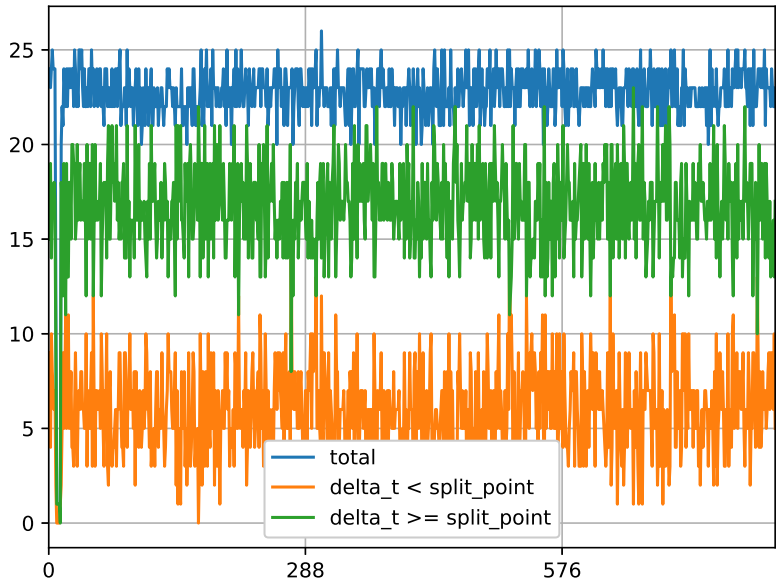


(b) Characteristics of the traffic to the master device.

Figure A.10: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with DoS attack (whole communication splitted by the direction, automatic setting of split points).

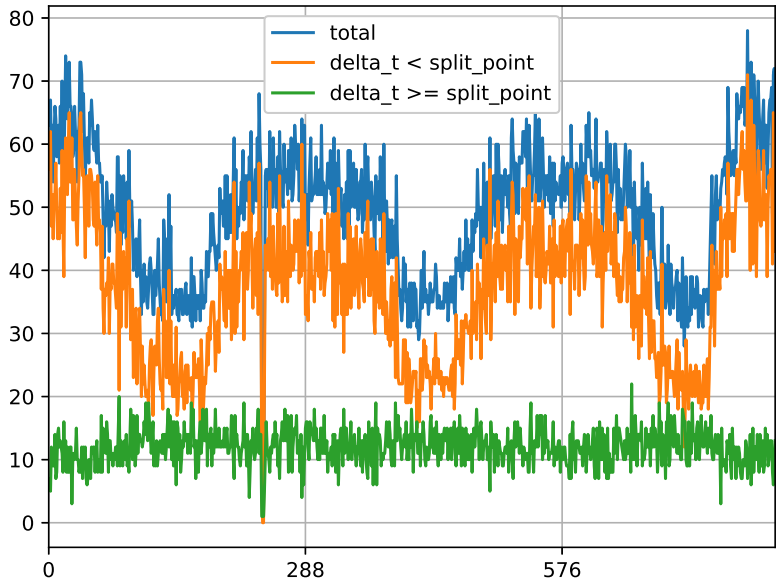


(a) Characteristics of the traffic from the master device.

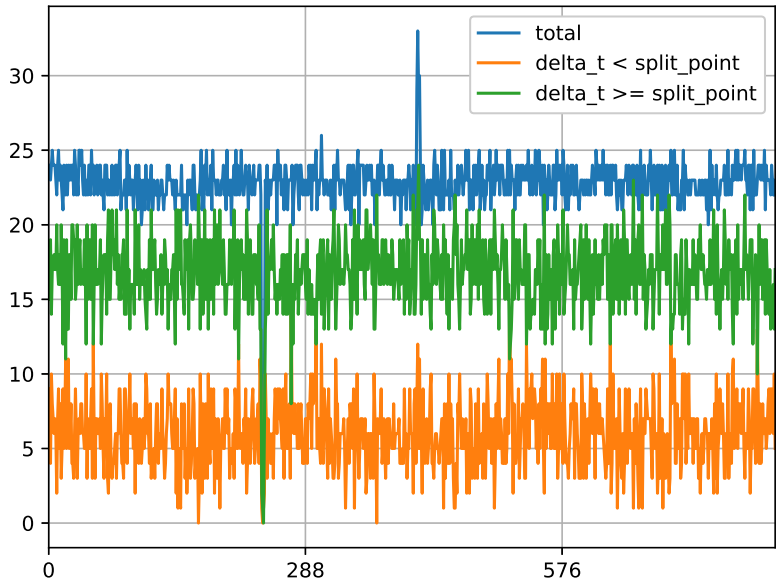


(b) Characteristics of the traffic to the master device.

Figure A.11: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with rogue devices attack (whole communication splitted by the direction, automatic setting of split points).

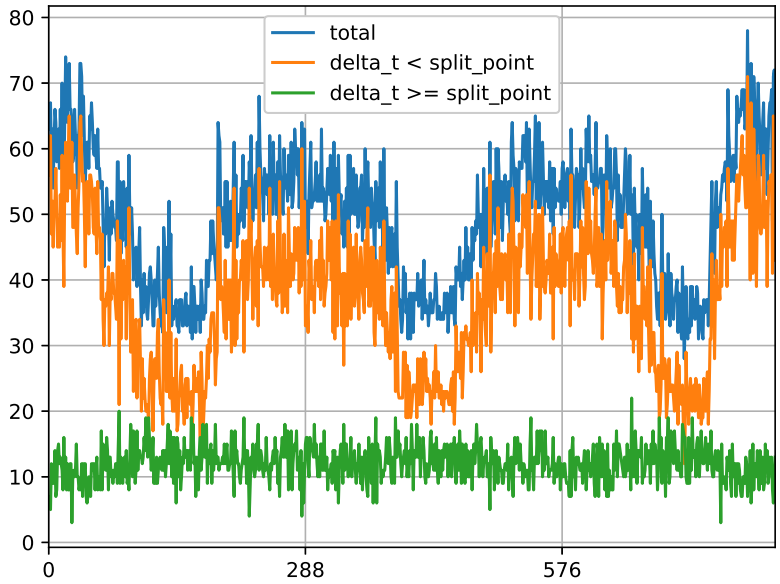


(a) Characteristics of the traffic from the master device.

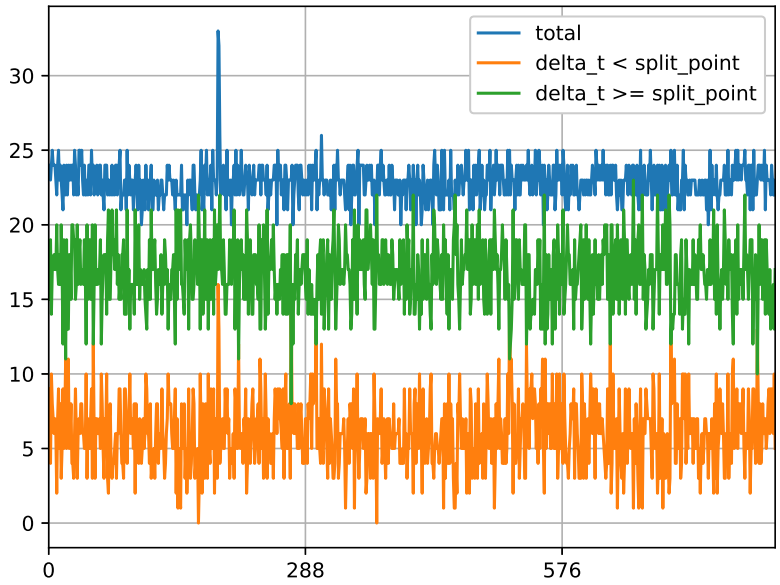


(b) Characteristics of the traffic to the master device.

Figure A.12: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with scanning attack (whole communication splitted by the direction, automatic setting of split points).



(a) Characteristics of the traffic from the master device.



(b) Characteristics of the traffic to the master device.

Figure A.13: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with switching attack (whole communication splitted by the direction, automatic setting of split points).

Appendix B

Figures and tables with manual split-points

Hovewer, splitting points of inter-arrival defined above, are not exactly the most appropriate for `mega104-17-12-18` dataset. The total amount of packets transmitted in five minute windows of these communication flow is much smaller than in other datasets. Subsequently, usual inter-arrival time is greater. We found experimentally more suitable splitting point: 1.0 and 5.0. Figures B.8 - B.14 show the characteristics of the dataset `mega104-17-12-18` utilizing splitting points 1.0 and 5.0 and also this dataset with different types of attacks.

Figure B.4 shows that Δt intervals used to provide more subtle distribution of the number of transmitted packets are not optimal. We can see, that the majority of the packets are transmitted after $\Delta t \geq 1.0$. Basically, intervals $< 0, 0.2)$, $< 0.2, 1.0)$ and $\Delta t \geq 1.0$ are useful for datasets with greater amount of packets transmitted in each 5 minute window. The number of transmitted packets in five minute window in `mega104-17-12-18` dataset fluctuate around 50 packets in the direction *from master* and around 23 packets in the opposite direction. Therefore intervals, that allow grater Δt_i might describe given communication flow more properly. Table B.6 list the 5 minute windows that were revealed as anomalies with the following Δt_i intervals: $< 0, 1.0)$, $< 1.0, 5.0)$ and $\Delta t \geq 1.0$. The results show that these intervals allow the detection all but one attacks with no false positive windows.

Dataset	**Dir.**	**Char.**	**List of windows**
13122018-mega104	fm	total	618, 725
		$\Delta t \in < 0, 0.2)$	618, 725
13122018-mega104	tm	total	618, 725
		$\Delta t \in < 0, 0.2)$	618, 725
		$\Delta t \in < 0.2, 1.0)$	738, 850
mega104-14-12-18	fm	$\Delta t \in < 0, 0.2)$	125
mega104-14-12-18	tm	$\Delta t \in < 0.2, 1.0)$	143
		$\Delta t \geq 1.0$	143
mega104-17-12-18	fm	total	784
		$\Delta t \in < 0, 0.2)$	683, 686
		$\Delta t \in < 0.2, 1.0)$	781, 795, 811
mega104-17-12-18	tm	$\Delta t \in < 0, 0.2)$	698, 702, 714
		$\Delta t \in < 0.2, 1.0)$	761, 767, 775
		$\Delta t \geq 1.0$	713
KTH-RTU8	fm	$\Delta t \in < 0.2, 1.0)$	1497, 1527 - 1529
		$\Delta t \geq 1.0$	1345, 1354, 1507
KTH-RTU8	tm	$\Delta t \in < 0.2, 1.0)$	1488, 1607, 1848 - 1849
		$\Delta t \geq 1.0$	1452, 1637
KTH-RTU11	fm	$\Delta t \geq 1.0$	1526 - 1529, 1539, 1813, 1896
RICS	fm	total	2564, 2609, 3458, 3498
		$\Delta t \in < 0, 0.2)$	3458, 3498
		$\Delta t \in < 0.2, 1.0)$	2572, 2659, 2855, 2934, 3057, 3124, 3439, 3692, 3710
		$\Delta t \geq 1.0$	2557, 2561, 2846, 2849, 2860, 2942, 3130, 3138, 3140, 3236, 3420, 3429, 3474, 3712
RICS	tm	total	3716
		$\Delta t \in < 0, 0.2)$	2543, 2578, 3140, 3160, 3420, 3498, 3716
		$\Delta t \in < 0.2, 1.0)$	2575, 2582, 2871, 3139, 3147, 3242, 3439, 3526, 3567
		$\Delta t \geq 1.0$	2543, 2673, 2841, 2849, 2860, 2933, 3001, 3006, 3140, 3160, 3263, 3419, 3420, 3421, 3427, 3429, 3434, 3474, 3712

Table B.1: Simple validation - list of five minute windows that does not fit into predefined range of values (manual setting of split-points).

Dataset	Direction	Characteristic	List of windows
KTH-RTU8	from master	$\Delta t \in < 0.2, 1.0)$	1528, 1529, 1530
KTH-RTU8	to master	$\Delta t \in < 0.2, 1.0)$	1849, 1850
KTH-RTU11	from master	$\Delta t > 1.0$	1527, 1528, 1529, 1530
RICS	from master	$\Delta t > 1.0$	3140
RICS	to master	$\Delta t > 1.0$	3420, 3421, 3422, 3429

Table B.2: 3-value window validation - list of first five minute window of 3-value window for which 2 of the three values do not fit into predefined range of values (manual setting of split-points). Only datasets with false positive elements are included.

Dataset	Dir.	Char.	**Simple validation** FP	Acc	**3-value validation** FP	Acc
10122018-104Mega	fm	all	0	100%	0	100%
10122018-104Mega	tm	all	0	100%	0	100%
13122018-mega104	fm	total	2	99,30%	0	100%
		$\Delta t \in <0, 0.2)$	2	99,30%	0	100%
13122018-mega104	tm	total	2	99,30%	0	100%
		$\Delta t \in <0, 0.2)$	2	99,30%	0	100%
		$\Delta t \in <0.2, 1.0)$	2	99,30%	0	100%
mega104-14-12-18	fm	$\Delta t \in <0, 0.2)$	1	98.41%	0	100%
mega104-14-12-18	tm	$\Delta t \in <0.2, 1.0)$	1	98.41%	0	100%
		$\Delta t \geq 1.0$	1	98.41%	0	100%
mega104-17-12-18	fm	total	1	99.63%	0	100%
		$\Delta t \in <0, 0.2)$	2	99.27%	0	100%
		$\Delta t \in <0.2, 1.0)$	3	98.90%	0	100%
mega104-17-12-18	tm	$\Delta t \in <0, 0.2)$	3	98.90%	0	100%
		$\Delta t \in <0.2, 1.0)$	3	98.90%	0	100%
		$\Delta t \geq 1.0$	1	99.63%	0	100%
KTH-RTU8	fm	$\Delta t \in <0.2, 1.0)$	4	99.38%	3	99.53%
		$\Delta t \geq 1.0$	3	99.53%	0	100%
KTH-RTU8	tm	$\Delta t \in <0.2, 1.0)$	4	99.38%	2	99.69%
		$\Delta t \geq 1.0$	2	99.69%	0	100%
KTH-RTU11	fm	$\Delta t \geq 1.0$	7	98.92%	4	99.38%
RICS	fm	total	4	99.68%	0	100%
		$\Delta t \in <0, 0.2)$	2	99.84%	0	100%
		$\Delta t \in <0.2, 1.0)$	9	99.27%	0	100%
		$\Delta t \geq 1.0$	14	98.87%	1	99.9%
RICS	tm	total	1	99.92%	0	100%
		$\Delta t \in <0, 0.2)$	7	99.44%	0	100%
		$\Delta t \in <0.2, 1.0)$	9	99.27%	0	100%
		$\Delta t \geq 1.0$	19	98.47%	4	99.7%

Table B.3: Validation results - results for individual characteristics (manual setting of split-points).

Dataset	**Simple validation**		**3-value validation**	
	FP	Acc	FP	Acc
10122018-104Mega	0	100%	0	100%
13122018-mega104	4	98.60%	0	100%
mega104-14-12-18	2	96.83%	0	100%
mega104-17-12-18	13	95.24%	0	100%
KTH-RTU8	13	98.00%	5	99.23%
KTH-RTU11	7	98.92%	4	99.38%
RICS	49	96.05%	5	99.60%

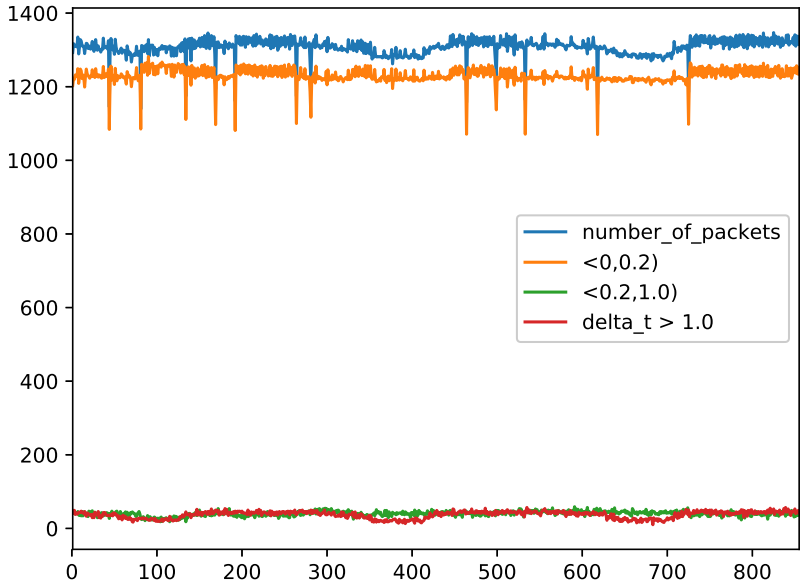
Table B.4: Validation-summary results (manual setting of split-points).

Dataset	Dir.	Char.	List of windows
Connection-lost	fm	total	309-310, 497-509
		$\Delta t \in < 0, 0.2)$	**156**
		$\Delta t \geq 1.0$	309-310, 498-508
	tm	total	309-311, 497-509
		$\Delta t \geq 1.0$	310-311, 497-509
Injection-attack	fm	$\Delta t \in < 0, 0.2)$	**156**
	tm	total	365-367
		$\Delta t \geq 1.0$	365-367
DoS-attack	fm	$\Delta t \in < 0, 0.2)$	110-111, 119-122, 124-126, 144, 146-151, 155-159
	tm	total	109-128, 141-160
		$\Delta t \geq 1.0$	109-128, 141-160
Rogue-devices	fm	total	8-12
		$\Delta t \in < 0, 0.2)$	**156**
		$\Delta t \geq 1.0$	8-12
	tm	total	7-12
		$\Delta t \geq 1.0$	7-12
Scanning-attack	fm	total	238-240
		$\Delta t \in < 0, 0.2)$	**156**
		$\Delta t \geq 1.0$	238-240
	tm	total	238-241, 412-415
		$\Delta t \in < 0.2, 1.0)$	413-414
		$\Delta t \geq 1.0$	238-241
Switching-attack	fm	$\Delta t \in < 0, 0.2)$	**156**
	tm	total	189-190
		$\Delta t \in < 0.2, 1.0)$	189-190

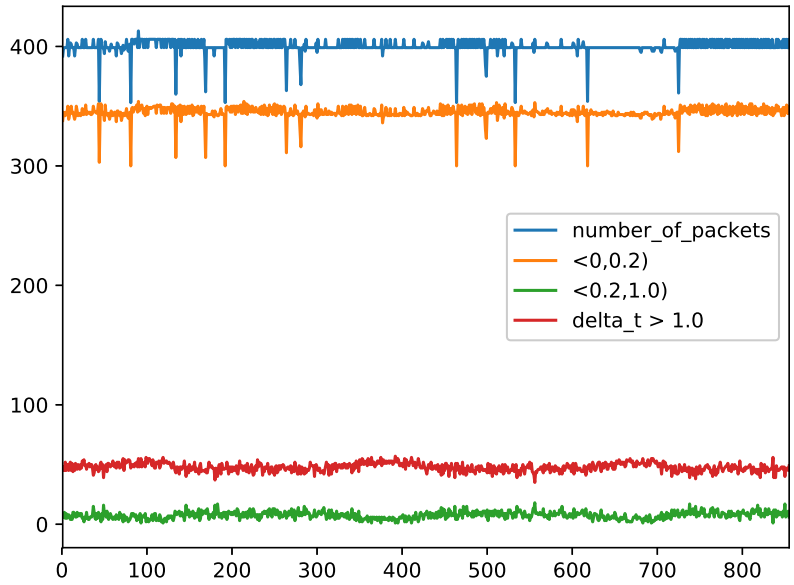
Table B.5: Attacks detection - results obtained with split points 0.2 and 1.0

Dataset	Dir.	Char.	List of windows
Connection-lost	fm	total	309-310, 497-509
		$\Delta t \geq 5.0$	309-310, 497-508
	tm	total	309-311, 497-509
		$\Delta t \in <1.0, 5.0)$	498-508
Injection-attack	tm	total	365-367
DoS-attack	fm	$\Delta t \geq 5.0$	110-112, 115-119, 124-125, 143-159
	tm	total	109-128, 141-160
		$\Delta t \in <1.0, 5.0)$	110-127, 142-160
Rogue-devices	fm	total	8-12
		$\Delta t \geq 5.0$	7-12
	tm	total	7-12
		$\Delta t \in <1.0, 5.0)$	8-12
Scanning-attack	fm	total	238-240
		$\Delta t \geq 5.0$	238-241
	tm	total	238-241, 412-415
		$\Delta t \in <0, 1.0)$	414
		$\Delta t \in <1.0, 5.0)$	239-240
Switching-attack	fm	$\Delta t \in <0, 1.0)$	189-191
	tm	total	189-190
		$\Delta t \in <0, 1.0)$	189-190

Table B.6: Attacks detection - results obtained with split points 1.0 and 5.0

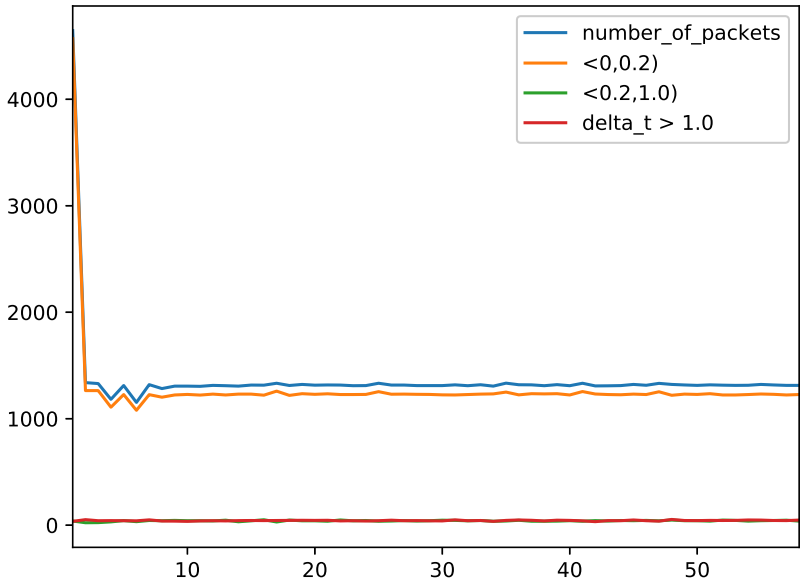


(a) Characteristics of the traffic from the master device.

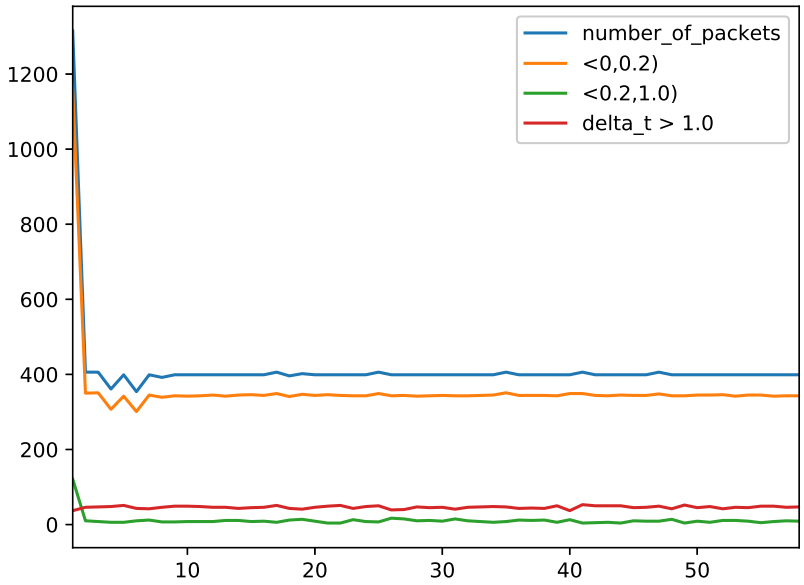


(b) Characteristics of the traffic to the master device.

Figure B.1: Graphs of the amounts of transmitted packets in five minute windows in 13122018-mega104 dataset (whole communication splitted by the direction, manual setting of split points).

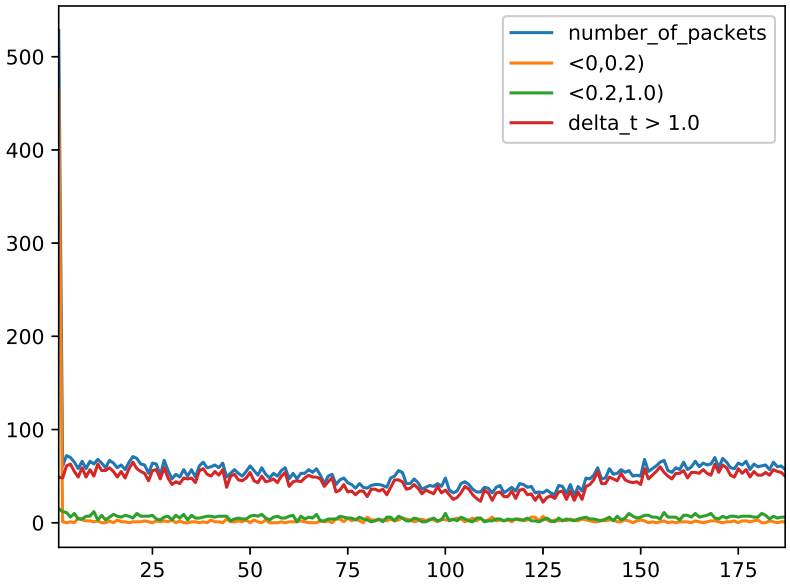


(a) Characteristics of the traffic from the master device.

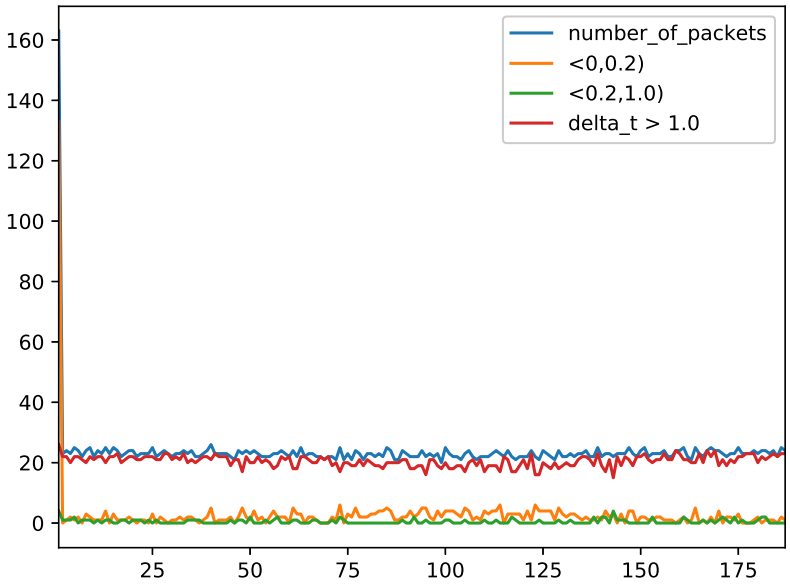


(b) Characteristics of the traffic to the master device.

Figure B.2: Graphs of the numbers of packets transmitted in five minute windows in 10122018-104Mega dataset (whole communication splitted by the direction, manual setting of split points).

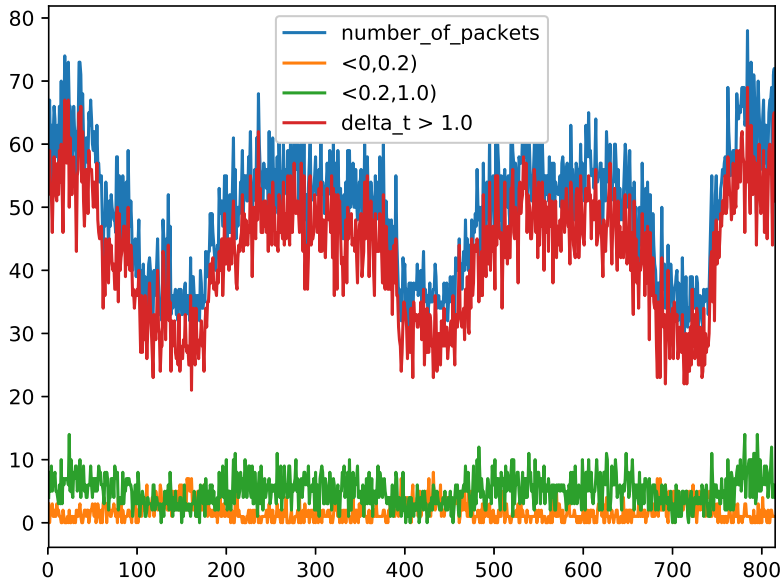


(a) Characteristics of the traffic from the master device.

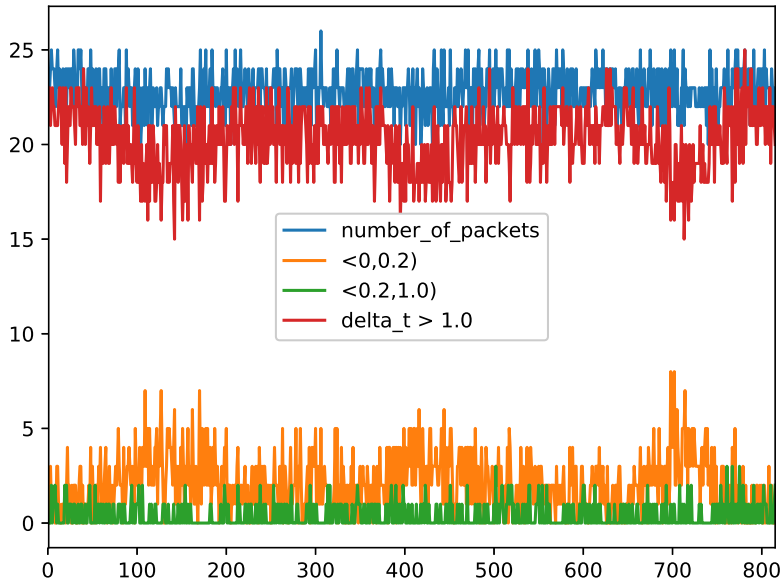


(b) Characteristics of the traffic to the master device.

Figure B.3: Graphs of the numbers of packets transmitted in five minute windows in mega104-14-12-18 dataset (whole communication splitted by the direction, manual setting of split points).

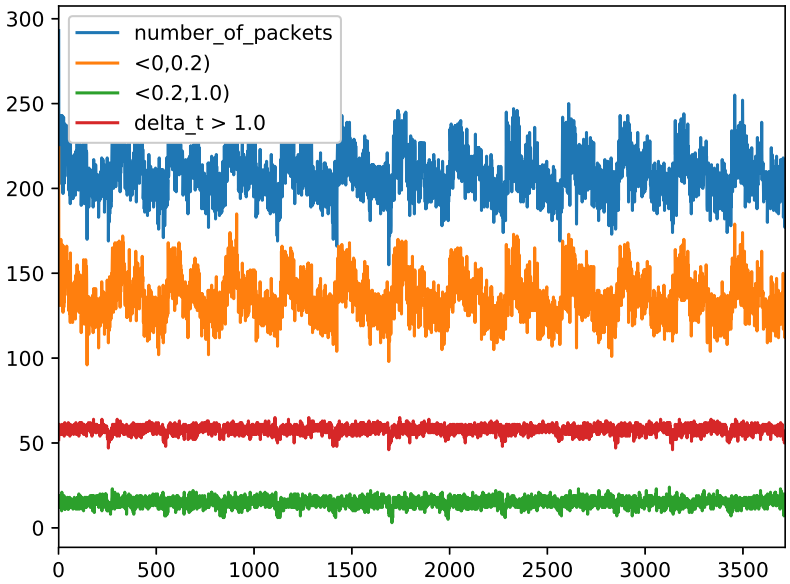


(a) Characteristics of the traffic from the master device.

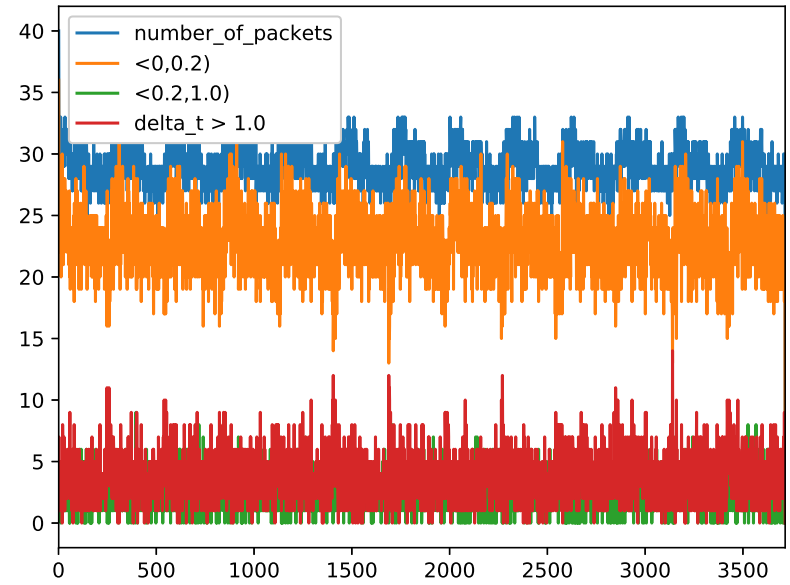


(b) Characteristics of the traffic to the master device.

Figure B.4: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset (whole communication splitted by the direction, manual setting of split points).

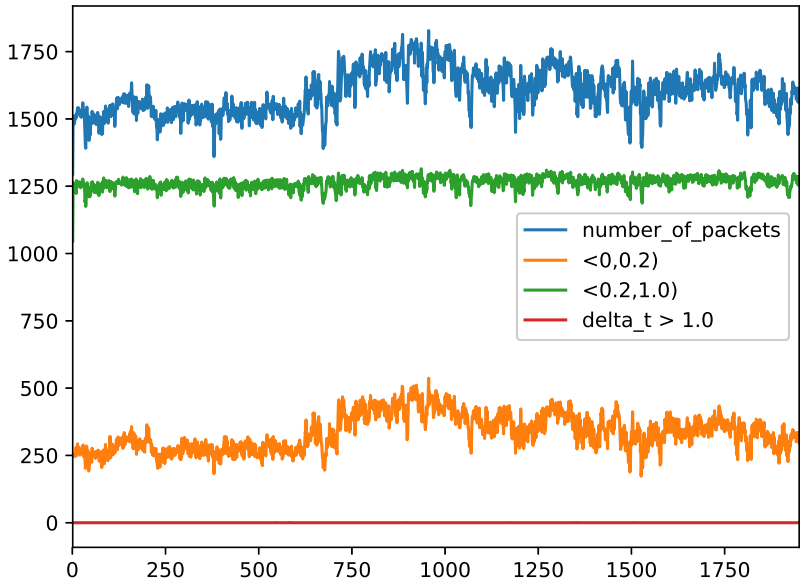


(a) Characteristics of the traffic from the master device.

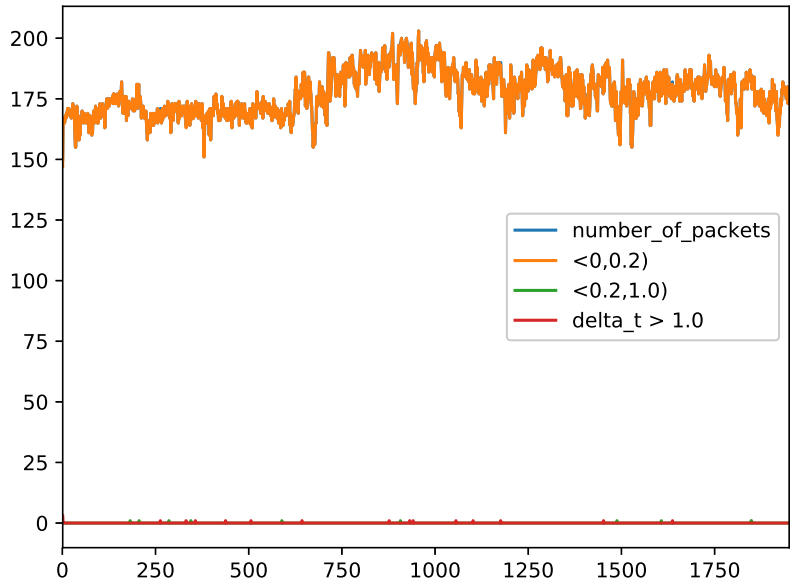


(b) Characteristics of the traffic to the master device.

Figure B.5: Graphs of the numbers of packets transmitted in five minute windows in RICS dataset (whole communication splitted by the direction, manual setting of split points).

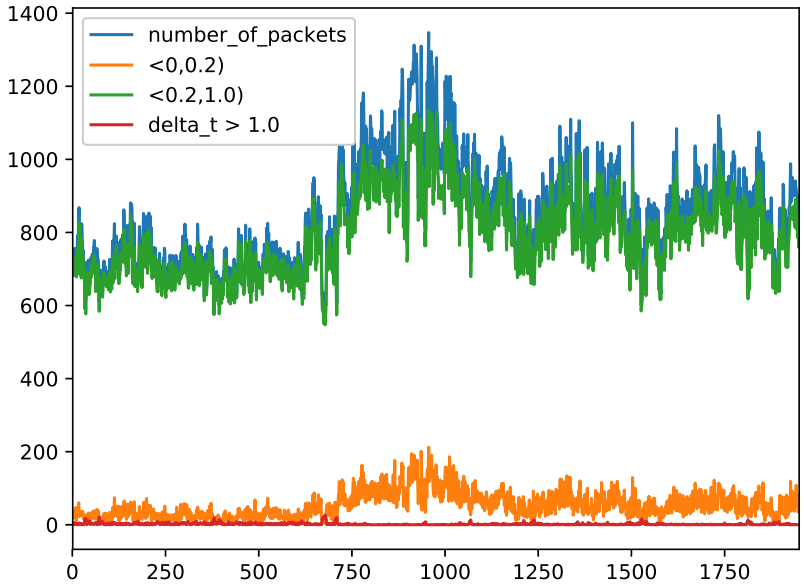


(a) Characteristics of the traffic from the master device.

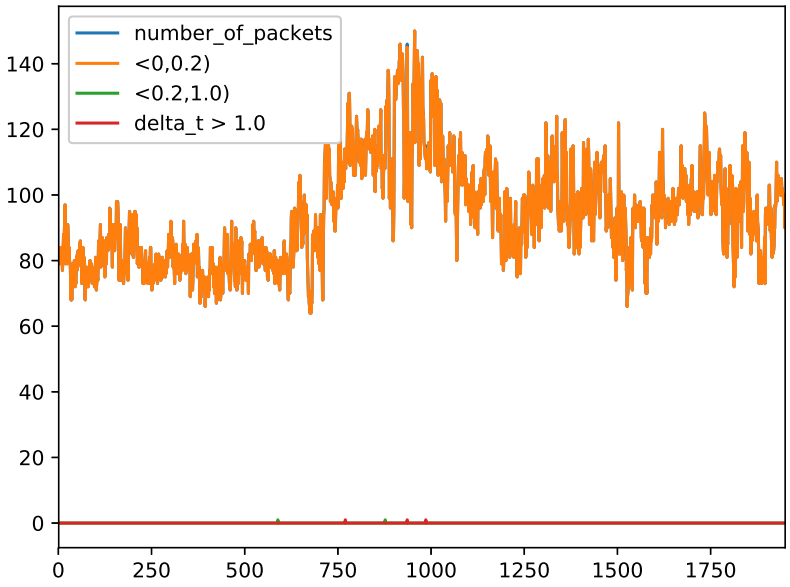


(b) Characteristics of the traffic to the master device.

Figure B.6: Graphs of the numbers of packets transmitted in five minute windows in KTH-RTU8 dataset (whole communication splitted by the direction, manual setting of split points).

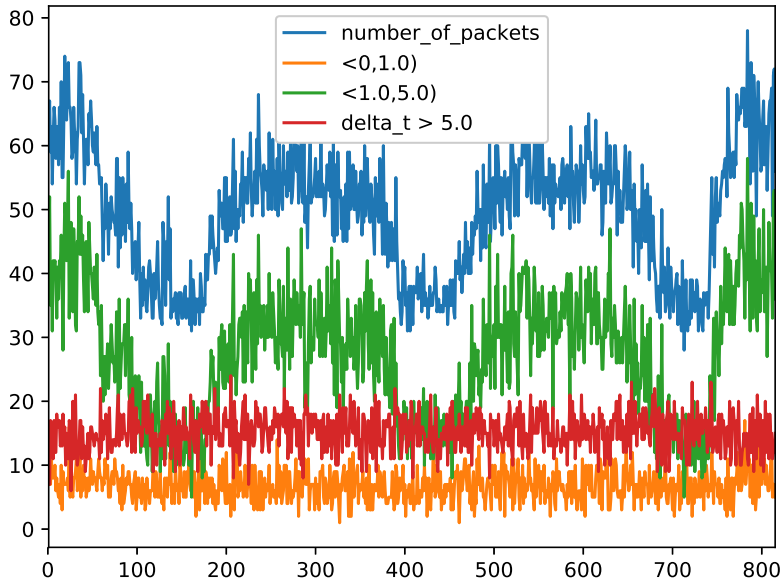


(a) Characteristics of the traffic from the master device.

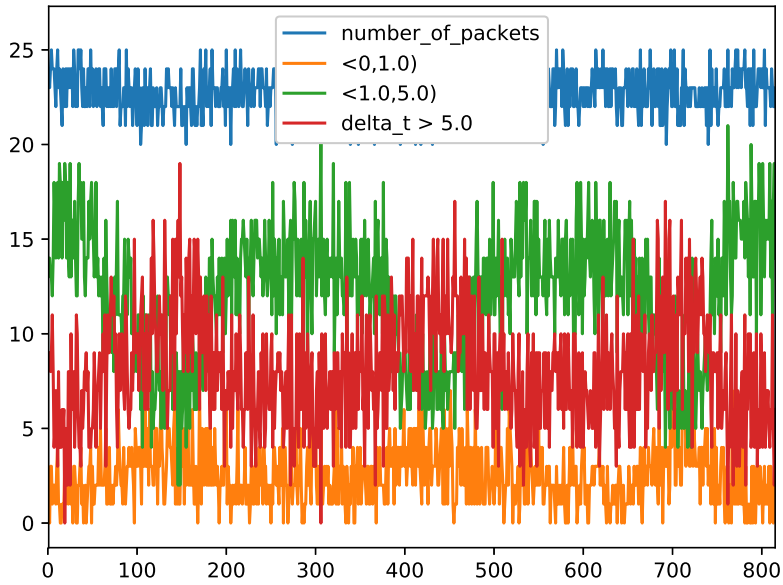


(b) Characteristics of the traffic to the master device.

Figure B.7: Graphs of the numbers of packets transmitted in five minute windows in KTH-RTU11 dataset (whole communication splitted by the direction, manual setting of split points).

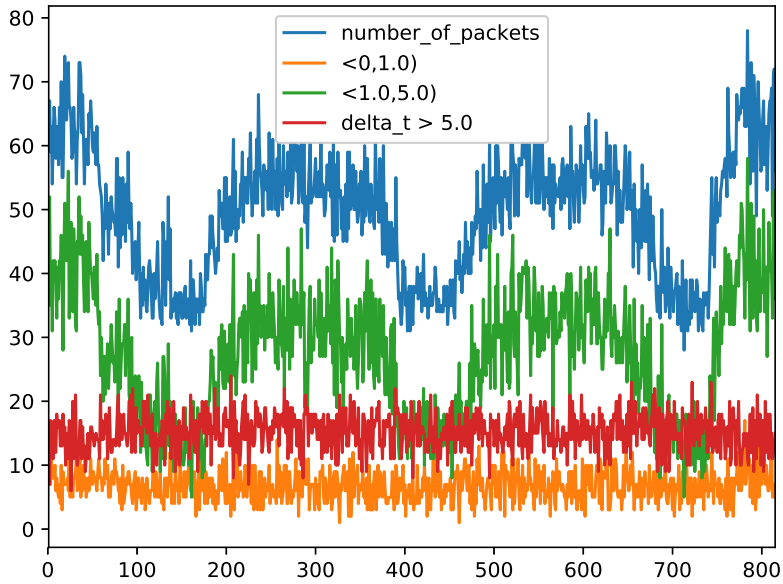


(a) Characteristics of the traffic from the master device.

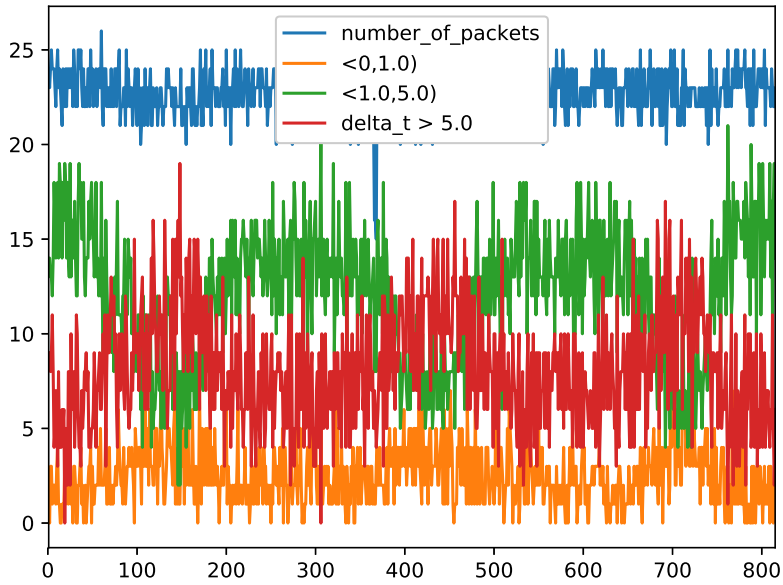


(b) Characteristics of the traffic to the master device.

Figure B.8: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with splitting points 1.0 and 5.0 (whole communication splitted by the direction, manual setting of split points).

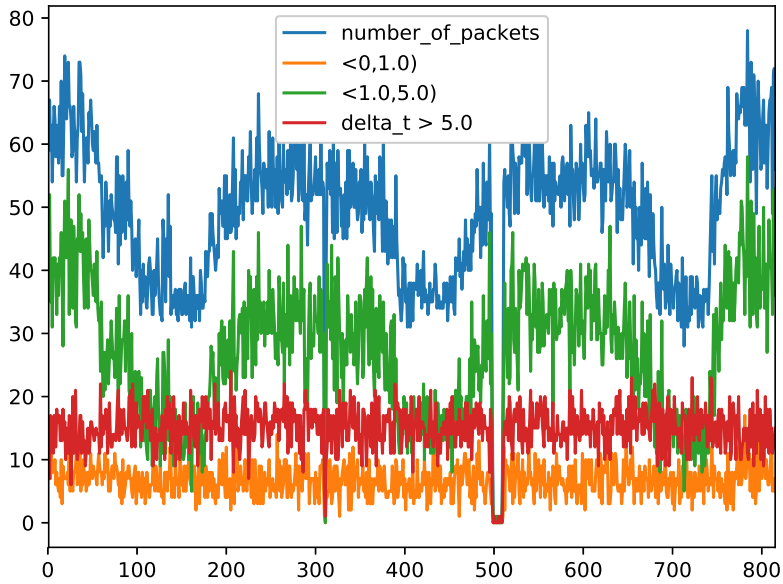


(a) Characteristics of the traffic from the master device.

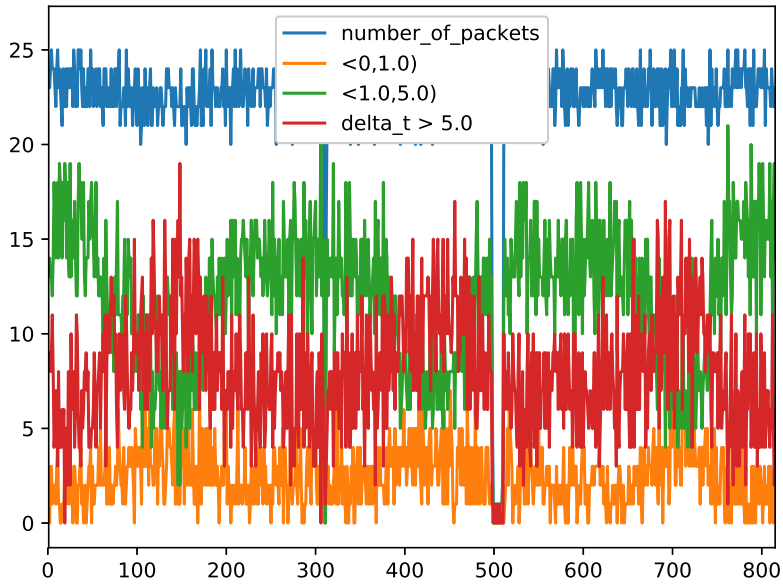


(b) Characteristics of the traffic to the master device.

Figure B.9: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with injection attack (whole communication splitted by the direction, manual setting of split points).

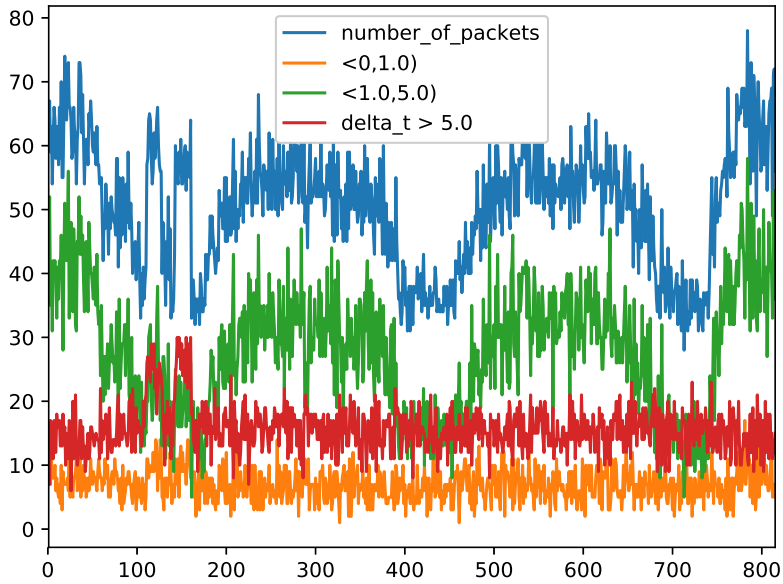


(a) Characteristics of the traffic from the master device.

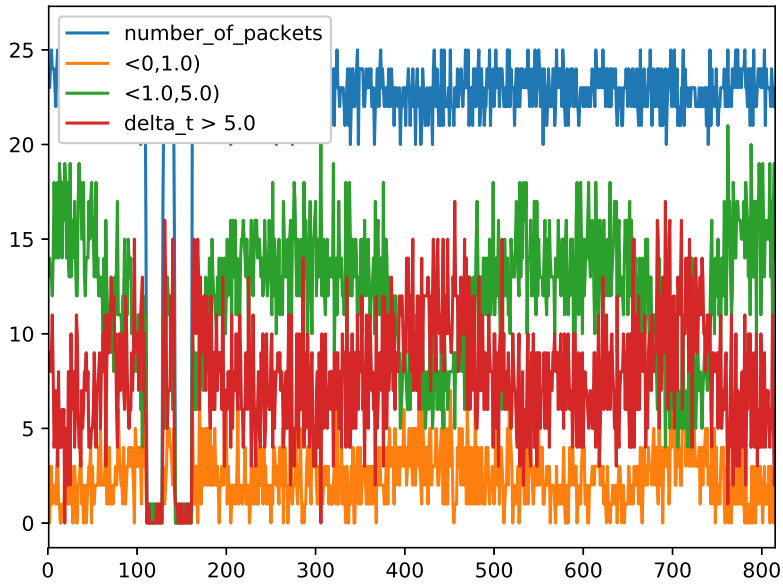


(b) Characteristics of the traffic to the master device.

Figure B.10: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with connection loss attack (whole communication splitted by the direction, manual setting of split points).

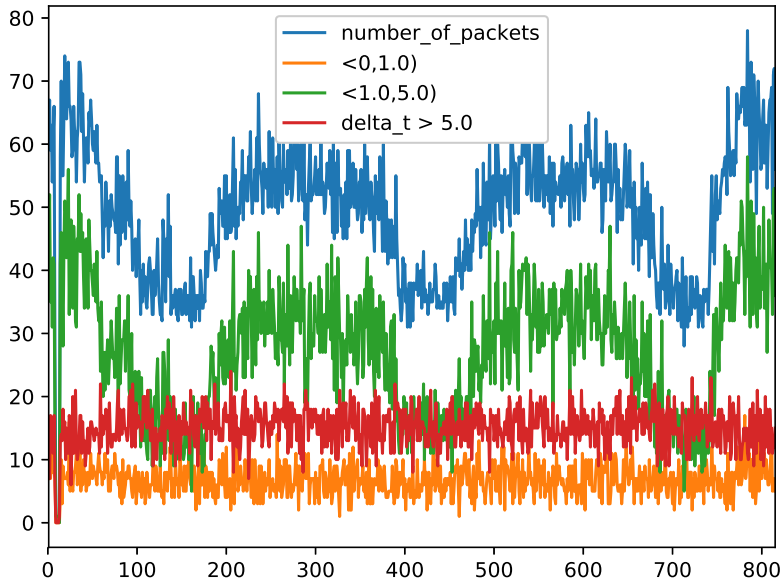


(a) Characteristics of the traffic from the master device.

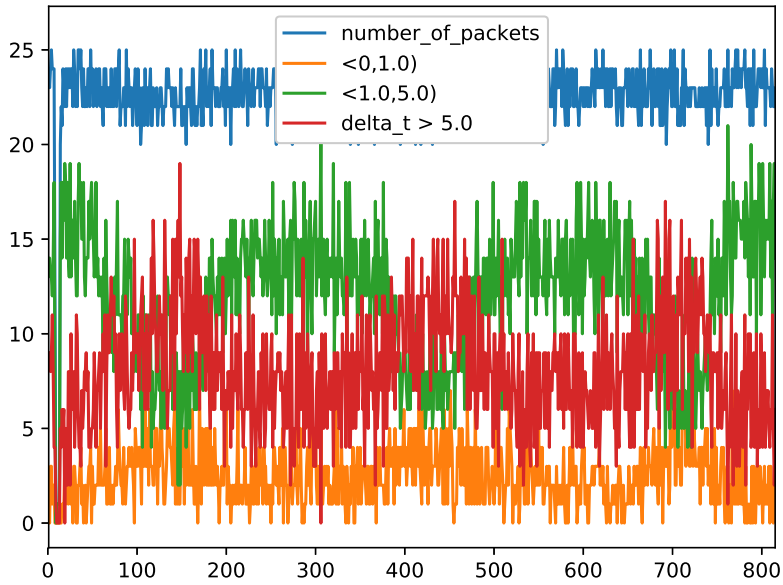


(b) Characteristics of the traffic to the master device.

Figure B.11: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with DoS attack (whole communication splitted by the direction, manual setting of split points).

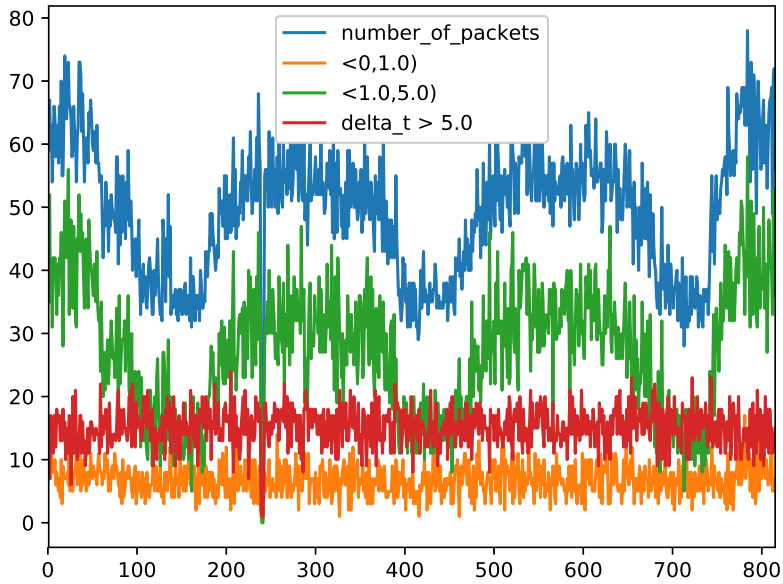


(a) Characteristics of the traffic from the master device.

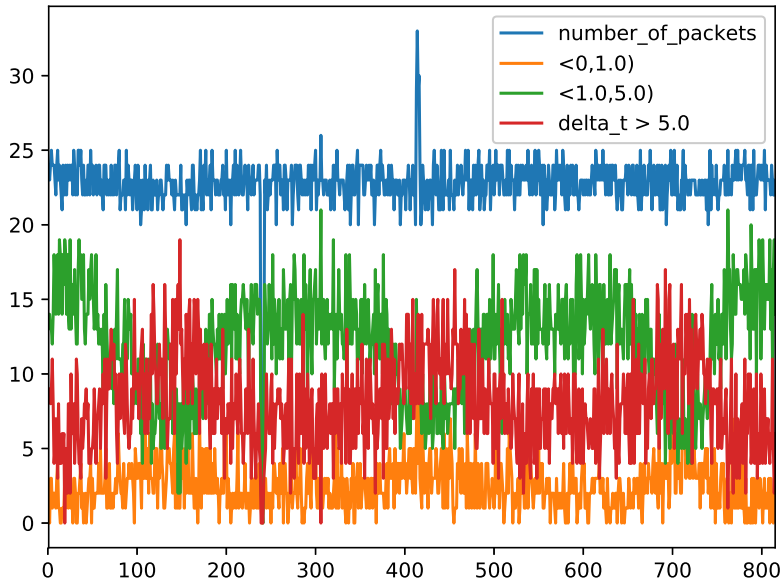


(b) Characteristics of the traffic to the master device.

Figure B.12: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with rogue devices attack (whole communication splitted by the direction, manual setting of split points).

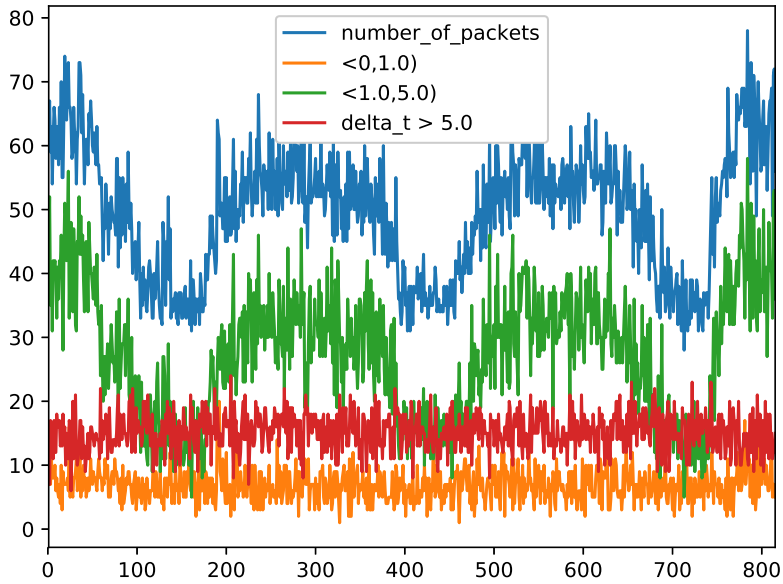


(a) Characteristics of the traffic from the master device.

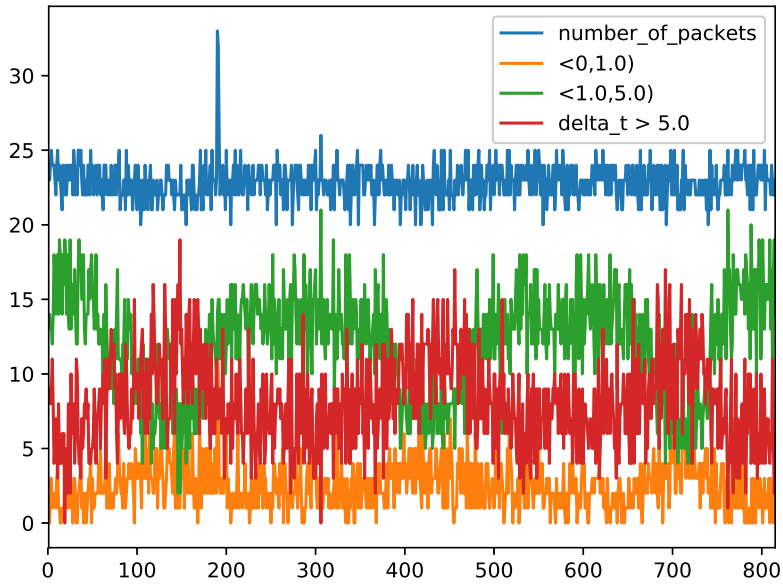


(b) Characteristics of the traffic to the master device.

Figure B.13: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with scanning attack (whole communication splitted by the direction, manual setting of split points).



(a) Characteristics of the traffic from the master device.



(b) Characteristics of the traffic to the master device.

Figure B.14: Graphs of the numbers of packets transmitted in five minute windows in mega104-17-12-18 dataset with switching attack (whole communication splitted by the direction, manual setting of split points).