Computer Networks 158 (2019) 46–60

Contents lists available at ScienceDirect

Computer Networks

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

Two-tier anomaly detection based on traffic profiling of the home automation system

Mariusz Gajewski[a,*], Jordi Mongay Batalla[a,b], Albert Levi[c], Cengiz Togay[d], Constandinos X. Mavromoustakis[e], George Mastorakis[f]

[a] National Institute of Telecommunications, Warsaw, Poland
[b] Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland
[c] Sabanci University, Faculty of Engineering and Natural Sciences, Istanbul, Turkey
[d] Computer Engineering Department, Bursa Uludag University, Bursa, Turkey
[e] Department of Computer Science, University of Nicosia, Nicosia, Cyprus
[f] Department of Informatics Engineering, Technological Educational Institute of Crete, Heraklion, Crete, Greece

ARTICLE INFO

Article history:
Received 31 October 2018
Revised 5 February 2019
Accepted 22 April 2019
Available online 24 April 2019

Keywords:
Home gateway
Wireless sensor networks
Smart home
Anomaly detection
Internet of Things

ABSTRACT

Smart building equipment and automation systems often become a target of attacks and are used for attacking other targets located out of the Home Area Network. Attacks are often related to changes in traffic volume, disturbed packet flow or excessive energy consumption. Their symptoms can be recognized and interpreted locally, using software agent at Home Gateway. Although anomalies are detected locally at the Home Gateway, they can be exploited globally. Thus, it is significantly important to detect global attack attempts through anomalies correlation.

Our proposal in this paper is the involvement of the Network Operator in Home Area Network security. Our paper describes a novel strategy for anomaly detection that consists of shared responsibilities between user and network provider. The proposed two-tier Intrusion Detection System uses a machine learning method for classifying the monitoring records and searching suspicious anomalies across the network at the service provider's data center. Result show that local anomaly detection combined with anomaly correlation at the service providers level can provide reliable information on the most frequent IoT devices misbehavior which may be caused by infection.

1. Introduction

Since telecom operators offer smart home services (in various business models), they are also able to offer security services for mitigation of attackers' activities. Moreover, the network operator: (i) has resources that enable storing data from Home Gateways (HGs) and performing computationally complex operations on them; (ii) has the ability to compare the results obtained from individual HGs.

This paper presents a mechanism for anomaly detection that splits the security tasks among users and network provider. The proposed mechanism uses computational resources of the operator and knowledge about the attack symptoms observed at users' premises. This comprehensive observation is essential in the light of attacks carried out using networked resource-limited devices, e.g., the Mirai attack (2016) [1].

The proliferation of IoT devices is broadening potential attacks. In this context, the Mirai botnet attack shows that availability of the malicious source code increases the probability of attack. In fact, more competent attackers use and modify code to expand attack capability. Therefore, mixing local and global observation techniques can be helpful in detection of coordinated attacks and, consequently, increases the chances of defending against an attack.

When detecting coordinated attacks, the knowledge about the specificity of the network activity may be helpful. Namely, the observation of the communication activity of smart devices leads to the conclusion that packet traffic generated by the constrained devices differs from traffic generated by network unconstrained hosts. Constrained devices have limited processing capabilities, memory size and small bandwidth consumption. Moreover, hardware and software components align usually strictly to the role the device plays in its environment. In this context, authors of RFC 7228 [2] distinguish between constrained devices of class 0, 1, and 2. It encompasses division into: very simple devices requiring prox-

* Corresponding author.
E-mail addresses: m.gajewski@itl.waw.pl (M. Gajewski), jordim@tele.pw.edu.pl (J. Mongay Batalla), levi@sabanciuniv.edu (A. Levi), ctogay@uludag.edu.tr (C. Togay), mavromoustakis.c@unic.ac.cy (C.X. Mavromoustakis), gmastorakis@staff.teicrete.gr (G. Mastorakis).

https://doi.org/10.1016/j.comnet.2019.04.013
1389-1286/

ies, gateways, or servers for Internet data exchange (class 0), constrained in data processing capabilities and but capable to use a protocol stack designed for constrained nodes (class 1), and less constrained and capable of supporting most of the same protocol stacks as used on user end devices i.e., smartphones (class 2).

In particular, constrained devices (of class 0 and 1 according to IETF terminology) do exchange data rarely in contrast to workstations or even smartphones operating over WLAN. In light of this, traffic profile in constrained networks should be characterized by lower throughputs, small payloads, stable packet loss and packet collision ratio (in the absence of smart device's mobility and radio interferences). Therefore, any deviation is a prerequisite for classifying such event as an attempt to attack the Home Area Network (HAN).

On the other hand, attacks on HAN networked devices are aimed to achieve various goals and therefore characterized by different attack vectors. In this case, the attack vector should be understood as a path or means by which a hacker can gain access to HAN resources in order to deliver a malicious payload. The potential attack vectors make use of the features of the HAN. Thus, any analysis of potential attack vectors should consider the characteristics of the network, such as:

- co-existence of different HAN communication technologies that are medium dependent (radio/cable, protocols used at different OSI layers) [3];
- the individual HAN network structure;
- specific smart device construction and constraints (i.e., according to classification defined in [2]).

Moreover, the problem of detecting attacks in HAN is more complicated when we consider several communication technologies and smart device deployments.

The rest of this paper is organized as follows: after a short presentation of the state of the art on Anomaly Detection Systems (ADS) in Section 2, we present our approach to intrusion detection in HAN networks based on the support of network provider in Section 3. Section 4 presents simulation results and compare the functionalities of our solution with current ADS systems. The results show the range of operation where the proposed system overcomes performance of other ADS. The paper is concluded in Section 5.

2. Related work

Home Area Network is a mix of technologies where the dominant are based on IEEE 802.15.4 (e.g., ZigBee, Thread, Z-Wave, etc.), IEEE 802.11 Wi-Fi and Bluetooth standards. Being located in an insecure wireless environment, HANs are vulnerable to cyber attacks from the immediate environment as well as from remote locations if smart devices are connected through the Internet [4,5]. Similarly, the attack can affect local resources as well as resources that are outside of the particular HAN. This gain necessitates of applying mechanisms dedicated to attack and intrusion detection in HANs. An attack can be defined as an attempt to gain unauthorized access to a service, resource or information. It could also be an attempt to compromise integrity, availability, or confidentiality of the system. In literature, we can find several methods for threat detection in HAN networks, which are ranked as a top risk similar to ransomware attacks [6]. Generally, they focus on attack detection (usually based on attack signatures) and intrusion detection (usually based on searching anomalies).

Detecting attacks based on signatures engages more protocol analysis than the ones based on traffic profile, but it is limited to known attack schemas. Since intrusion detection makes use of anomaly detection, it is open for unknown threats but does not provide the same effectiveness as the signature based ones. Many solutions described in the literature combine these two approaches in order to increase the effectiveness of attack detection (i.e., in [7,8] and [9]).

A good example of this approach is a conceptual IDS called SVELTE described in [10]. Authors implemented the monitoring part in resource-constrained WSN (Wireless Sensor Network) nodes while the resource demanding functionality was implemented in Border Router (BR) which is also responsible for connecting the 6LoWPAN capable devices with the Internet. The basic IDS functionality of SVELTE is to detect attacks targeted to routing operations (i.e., detection of spoofed or altered information, sinkholes, and selective forwarding attacks).

The authors of [11] proposed a specification based IDS for HANs in which the feature space was defined based on the network specifications extracted from the IEEE 802.15.4 standard. This solution is based on machine learning (ML) algorithm for intrusion detection and prevention system.

Some researchers pointed on selected features of the wireless network traffic as dominant in threat assessment. Following this idea, Jokar et al. presented an algorithm for detecting spoofing attacks against static IEEE 802.15.4 networks. This method is based on analysis of the received signal strength (RSS) of network packets [12]. The same parameter has also been selected for detecting Sybil attacks by Marian and Mircea [13]. Authors proposed a detection system which uses metrics derived from RSS measurements. On that basis, the proposed algorithms are able to detect an attack, provide the approximate location and then classify it as malicious or not.

Al Baalbaki et al. developed a network traffic analysis method based on packets source and destination addresses, and packet traffic statistics for anomaly detection. In ZigBee networks, the proposed mechanism collects traffic statistics and identifies abuses. After detection, the method may classify the attack type [14].

Sometimes detection of anomalies encountered in HAN requires deep inspection of data packets. In this case, searching for anomalies is carried out on the basis of exchanged data analysis. In this context, Chen et al. [15] focused on the detection of false data injection attacks in smart grids and specifically data injected by advanced metering infrastructure elements (AMI). For that reason, authors exploited spatial-temporal correlations between grid components for real-time detection of injected data.

Also, the increased node energy consumption can suggest that an attack has occurred. One of the examples found in the literature examines energy consumption to identify running attacks [16]. An essential aspect in IDS design encompasses the problem of changing conditions which can be identified as malicious behavior. In [17] authors have proposed IDS solutions that take into consideration aspects related to changes in ad hoc and wireless sensor networks. Mainly, these solutions address difficulties with proper classification of the node behavior in case of: (i) changes in the network topology, (ii) multiple behavior switching of the malicious node between normal and anomalous, and (iii) presence of the malicious node that moves out of the IDS range before being observed.

The drawback of the misuse detection approach is that these methods are limited to a set of known attacks. Instead, they usually achieve high efficiency in detection. On the other hand, security violations can also be detected by identifying abnormal system usage patterns (this intuitive observation has been formalized by Denning [18]). In consequence, most anomaly detection techniques based on that conclusion attempt to define normal activity profiles. It is described by defining various metrics typical for the observed object. Next, an intrusion is detected when the actual system behavior differs from the normal profiles. Although the efficiency of anomaly detection in security violations is lower compared to misuse detection, it is open to unknown attacks (zero-day attacks).

3. Intrusion detection system distributed in smart homes and operator's management

3.1. Proposed solution

In this paper, we focus on attacks which affect network performance of Home Gateway in a visible way. More precisely, this paper addresses those attacks in HAN wireless network that disrupt the communication flow between network nodes, including flows between nodes and HG. Moreover, the results of running attacks are observable at the HG's network interfaces, and on that basis, it is possible to infer a possible malicious behavior of network node(s). A detailed study of these attacks leads to a conclusion that they result in an increase of the network traffic due to the injection of new packets. If the number or the rate is enormously high, it can result in opposite effect – the reduction of the network traffic due to collisions and loss of packets. Both effects may occur together increasing the traffic of specific packets and decreasing others. This leads to the conclusion that the packet traffic profile observable at the HAN interfaces under the normal condition differs from the packet traffic profile under the attack.

Each constrained end device connected to the HAN, which performs its tasks according to intention of the user, does not change its behavior radically in a short period of time. In consequence, the flow-based profile remains unchanged. It gives the possibility to detect deviations from the profile that occur when HAN becomes the target (or tool) of an attack. As mentioned in Section 2, the drawback of anomaly detection is lower efficiency as compared to misuse detection, but on the other hand, it is an effective method of unknown attack detection. It allows early detection of the attack attempt when symptoms are known. Although it is difficult to determine the type of attack on that basis, source and the scale of attack can be estimated. Since the smart home market is growing rapidly, new smart devices are not flawless. Also, the human factor plays a major role in making Smart Home concept vulnerable. Table 1 maps attack symptoms to the main vulnerabilities defined by the Open Web Application Security Project [19].

Moreover, the locally identified security violations can propagate influencing the network providers' resources. For that reason, the process of anomalies identification should be extended on the service provider network to include outputs from many HGs. The motivation behind extending anomalies detection is fast growing number of end devices being susceptible to cyber attacks and thus being a threat to other devices not only in HAN but also on the Internet.

Out of the many techniques being researched in an attempt to increase the efficiency of a local IDS, alert correlation is the most promising one. Alert correlation is the process where all the alerts generated locally are sent to the centralized system. This system is responsible for analysis and comparison of the alerts. From the providers' point of view, the primary goal of correlating alerts is to detect threats that are yet unknown but may indicate a global attack caused by infected end devices.

Therefore we suggest splitting the IDS functionalities between customer premises access device (e.g., Home Gateway, Residential router, technology controller/hub) and network provider's management system. The functions assigned to the local device provides a collection of monitoring data and online detection of anomalies. It seems to be particularly important in the breakdown of Internet access that can also be caused by the hacker's attack. In turn, the network provider's role is to detect threats that are common for all of the devices managed by the operator. Potentially, detecting such attacks is necessary for further mitigation of distributed attacks whose source can be in constrained smart devices located within HAN. This problem was addressed not only by business but also by national and international agencies pressing cybersecurity problems [4,18,20,21].

Since constrained devices have limited protocol implementation, their packet communication schema is also limited and relatively easy to describe using statistical quantities (i.e., an average number of packets sent/received, average inter-arrival time, average payload size, etc.). Moreover, off-the-shelf constrained devices are not able to carry out tasks related to detecting anomalies. For that reason, the HG is the natural point of observation of the behavior of these devices. Such an approach gives the opportunity to view the packet headers as well as a source for statistical analyses. In this way, for each device in HAN, it is possible to create a traffic profile for the normal state and look for anomalies. The HG (or

Table 1
Top 10 IoT vulnerabilities defined by OWASP (source [19]) mapped on features of packet flow.

Vulnerability	Description	IP capable	Non-IP
Insecure web interface and Insufficient authentication/Authorization	"Attacker uses weak credentials, captures plain-text credentials or enumerates accounts to access the web interface"	Attacks increase **the average number of packets.** Specifically, attacks engage traffic on selected ports	EDs offer limited user access possibilities. Particularly they do not support web interface. In consequence, we assume that this vulnerability does not affect in/outgoing traffic
Insecure network services	"Attacker uses vulnerable network services to attack the device itself or bounce attacks off the device"	Attacks increase **the average number of packets,** causes **anomalies in mean packet size,**	Attacks increase **the average number of packets,** causes **anomalies in mean packet size,**
Lack of transport encryption/integrity verification	"Attacker uses the lack of transport encryption to view data being passed over the network"	Attacks increase **the average number of packets,** causes **anomalies in mean packet size,**	Transport layer is technology specific. We assume no impact on non-IP subnets.
Insecure cloud interface and mobile interface	"Attacker uses multiple vectors such as insufficient authentication, lack of transport encryption and account enumeration to access data or controls via the cloud website/mobile interface"	Attacks increase **the average number of packets,** causes **anomalies in mean packet size,**	Any application based requests are processed by the (1) backend processes then (2) the local technology controller, and depends Attacks increase **the average number of packets,** causes **anomalies in mean packet size.**
Insecure Software/Firmware	"Attacker uses multiple vectors such as capturing update files via unencrypted connections, the update file itself is not encrypted, or they are able to perform their own malicious update via DNS hijacking."	Attacks increase **the average number of packets,** causes **anomalies in mean packet size, the inter-arrival times are shortened**	Attacks increase **the average number of packets,** causes **anomalies in mean packet size, shortens the inter-arrival time**

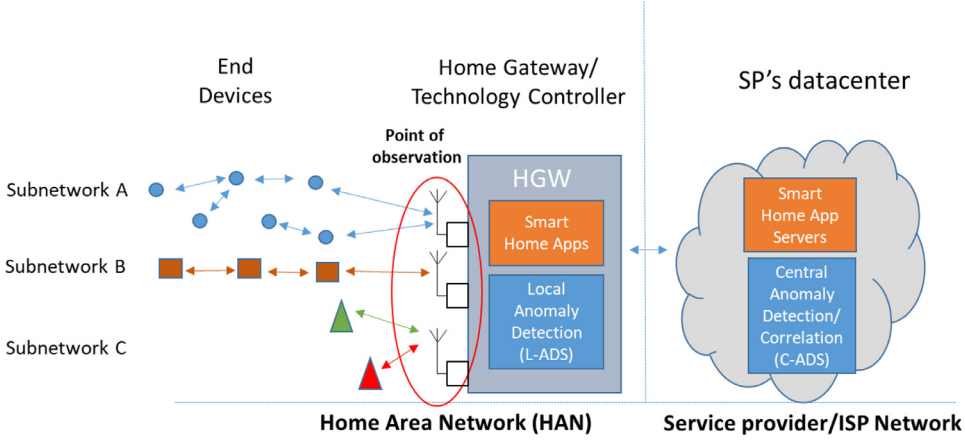


Fig. 1. The secure Smart Home concept.

technology controller) is also the earliest possible point of anomaly detection at the local level (i.e., when an anomalous behavior of an End Device is observed). It is also acquired to assess the possible existence of anomalies in HG because of the local addressing scheme used inside the HAN network – it acts as an local anomaly detection system – L-ADS. Moreover, it is also possible to isolate End Devices (ED) with an anomalous or suspicious behavior.

The proposed solution also assumes that the anomalous behavior of end devices is further analyzed to find behavioral similarities across the network at the service provider's premises – the Central ADS (C-ADS). The C-ADS executes the anomaly detection procedures that engage more resources (CPU, memory). Such resources are not available at the Home GW. For that purpose, C-ADS searches similar monitoring data which points to an attack. The above mentioned dependencies are shown in Fig. 1.

Essentially, the local detection mechanism makes use of a Machine Learning algorithm. Specifically, the located in HG process assigns monitoring records to normal or anomaly (attack) classes. For that purpose, it makes use of the Naïve Bayesian (NB) model for inference about parameter learning. We assume that the NB structure is fixed and specified on the basis of the domain knowledge about dependencies between features (e.g., the average number of packets, average packet size, inter-arrival time, etc.). In this case, an NB classifier can be learned from training data, and the learning process contains only Conditional Probability Tables (CPTs) estimation. Monitoring records are input data collected during the training phase. Further, in the validation phase, it makes a decision on the classification of new monitoring records based on CPTs.

In this context, the main issue is the distribution of anomaly (attack) detection tasks between HG and the operator's resources to increase the effectiveness of detection. The natural solution is to exploit existing high-performance operator infrastructure to implement advanced and computationally-demanding detection technologies. On the other hand, it is desirable to leave anomaly detection functionality at the user's premises to allow a quick response.

By combining both approaches, we propose to make local anomaly detection based on a full set of variables for each single monitoring record, whereas C-ADS will do detection based on larger dataset supported by data mining techniques. The C-ADS does not consider time dependencies between records and it classifies each monitoring record in near real-time (so-called "Online attack detection") using ML algorithm based on NB. This task can be carried out by the HG without a significant computing effort.

As an extension to the local detection process, we suggest centralized anomaly correlation and decision carried out by the C-ADS. It assumes searching characteristic sequence patterns within the monitored record dataset. Because of computational overhead required for that activity, we suggest performing those calculations by using the resources of the network provider.

The analysis results and the corresponding alarms are then correlated to achieve higher-level descriptions of attacks and a comprehensive view of the security issues raised during the analysis. The overall procedure is presented in Fig. 2.

The agent is installed on Home Gateways located at user's premises (see Fig. 2). The module is responsible for the collection of monitoring data that describe the behavior of the communicating local smart devices and the HG itself. The agent collects the available statistics from data gathered from various HAN interfaces (i.e., mean values of a number of packets, etc.). Records containing monitoring data are examined by the agent to determine if there are symptoms of an attack. This agent activity is not focused on detection of specific attacks but aims instead at detection of suspicious behavior that affects other smart devices in HAN and has an impact on communication with remote resources (e.g., network/service providers of cloud services as shown in [22] and [23]).

The main source of information about the state of the HAN components is the data collected by the HG including packets sniffed independently on technology interfaces. Since HG can be equipped with single or multiple LAN and WLAN interfaces, packet gathering process can be carried out by technology controllers that are responsible for monitoring the home automation system. They usually serve one or multiple HA communication technologies acting as a gateway between HA technologies (i.e., different wireless networks) and a packet network. In this case, packets statistics are calculated in HG based on data obtained from the controller's network interfaces. In consequence, the packet traffic analysis that aims at detecting anomalies should be performed independently for each of the sniffed interfaces. Finally, the detected anomalies are annotated with additional information about HAN devices engaged in the suspected packet flow. Further processing of the detected anomalies is performed by the service provider. The overall data flow and the main components of the anomaly detection in HAN are presented in Fig. 3.

This section describes the characteristics of each step of data processing and provides a theoretical foundation for processes performed by them.

3.2. Packet gathering

Our central object for classification is the packet flow generated by the HA device (network endpoint). For the work presented in this paper, we have limited our definition of a flow to being series of packets captured on interfaces. Generally, we distinguish

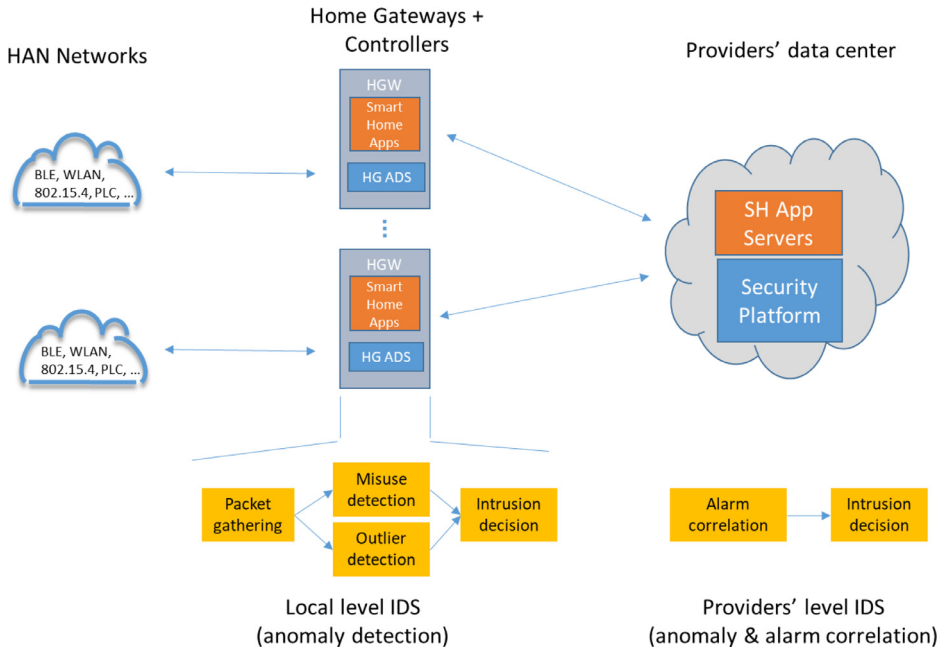


Fig. 2. Overall information flow between functional blocks.

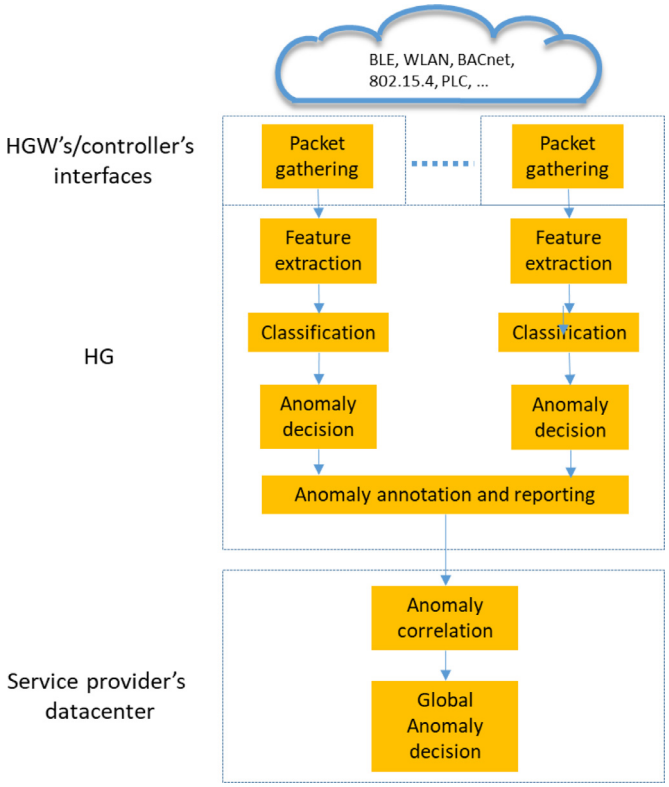


Fig. 3. Data flow of the anomaly detection process.

two types of interfaces: the first one supporting mesh subnetworks and the second one, supporting star topology subnetworks. In HA applications, mesh subnets are usually used to ensure reliability and stable conditions for data exchange between devices. Wireless technologies supporting the mesh topology are mainly based on IEEE 802.15.4 standard. Due to the nature of 802.15.4 radio communication, packet size is limited, and packet overhead is minimized. Therefore, constrained devices usually implement a limited protocol stack and simplify the addressing scheme as much as possible. For that reason, 802.15.4 devices usually do not use full IP stack and network layer addressing for communication. Moreover, address space required for addressing all devices within the entire network is not large and hence network addressing scheme can be limited.

Following this conclusion, we assume that end devices that use mesh topology networks do not use IP addressing scheme unlike devices that follow the star topology which we assume are mainly 802.11.3 WLAN devices. It causes that we consider two sets of features that will be taken into account during classification and correlation. Therefore, all the packets captured at each interface will be assigned to one of the two groups. Table 1 presents the scope of information that is retained on the basis of captured packets.

We define packet flow as a group of packets flowing from the specified source to a specified destination during a specified observation time window T_i. The essential requirement is that these packets are observable on the HG interfaces independently for each subnet and it is available at least information about the packet header according to Table 1. It implies that these packets are sent by the communicating node either as a packet source or packet forwarder. In the latter case, the communicating node plays a router role. Similarly, the HG can act as a router or it can be a packet source. Moreover, some wireless technologies (i.e. ZigBee [24]) assume that nodes are grouped in subnetworks. In this case, it is additionally necessary to distinguish the subnet identifier. The output of the process is a list of packets described by their attributes.

It should also be noted that a deep analysis of the higher (then network) layer payloads is often infeasible because almost all technologies provide security mechanisms at the network layer, and the entire payload of the network level packet is encrypted.

3.3. Feature extraction

Once we have captured packets, the next step is to perform feature extraction process. The gathered packets are the basis for traffic statistics that describe each end device conversation within the HAN. Since there are different communication technologies used in HAN and classifiers used for packet traffic within those technologies, there are also different sets of features used in classification

Table 2
Features retrieved from gathered packets broken down by type of subnetwork (IP based or non-IP).

Feature	(Sub)network type
Timestamp (denotes the time the packet was received)	IP, non-IP
Source MAC address	non-IP
Destination MAC address	non-IP
L3 source address	IP, non-IP
L3 destination address	IP, non-IP
Higher layer protocol	IP
Higher layer protocol type, one of {routing, management, data}	non-IP

process. Given the assumption about IP and non-IP traffic observed on HG network interfaces, we can obtain various statistics for each subnetwork. Statistics are computed based on the observation time window T_i and include parameters as presented in Table 2.

The features mentioned in Table 3 are computed for each packet flow within the time window. Given the above list of features, the flow-based profile of each HAN connected device can be constructed. It is computed during the training phase when statistical parameters are computed (according to Table 3) for flows identified with parameters from Table 2. This model is to illustrate the behavior of an end device, which is seen from the position of the HG/controller.

3.4. Feature analysis and classification (Machine learning)

In this step, we use data classification algorithm to prepare the model in the training phase for anomalies detection in packet traffic and Bayes for classification unobserved traffic profile assignment. The Bayesian methods are popular for dealing with uncertainty in real decision making due to its high capacity of adapting to a large number of situations [25]. For this reason, it has been applied successfully to various research domains. Generally, the Bayes Network structure provides explicit inter-relationships among the features. The literature provides examples of the three basic types of NB classifiers, namely, Naive Bayesian Classifier (NBC), Learned NB and Expert-elicited NB. The NBC is the simplest and computationally effective model. The NBC considers child nodes where they all share the same and single parent node (decision node). The NBC also assumes conditional independence for the child nodes. Learned NB is an NB whose structure (Directed Acyclic Graph, DAG) and parameters (tables of conditional probabilities) are determined on the basis of training data. In the first step, the DAG structure has to be constructed. Next, the Conditional Probability Table (CPT) parameters defined by the DAG have to be estimated. In the Expert-elicited, the structure of NB is constructed manually on the basis of knowledge of a domain expert (in terms of determining conditional dependencies between features).

In our case, we assume that the dependencies between features are negligible. This approach simplifies the structure of the graph and makes calculations much easier, which is important when dealing with a large number of variables (features). On the other hand, different features could be dependent (i.e., inter-arrival time and number of packets). It could lead to the misclassification if the method were used to assign multiple classes. Although the independence assumption can be hardly met, the NB classifier has outperformed many sophisticated classifiers, especially where the features are not strongly correlated [26].

The second assumption is regarding the values of features represented by variables. If they are not discrete, the values of each numerical feature should be normally distributed. This assumption is strong when we consider Internet traffic, but in the case of constrained end devices, we know that the traffic profile results from performed tasks and these tasks are minimized. Therefore, for further considerations, we assume that the packet traffic can be normally distributed. Similarly, in the case of WiFi devices, we apply the same principle, because we consider only smart devices that perform tasks strictly defined by the vendor.

Let us consider an observed data set $X = (X_1, \ldots, X_n)$ with n data samples, where each sample contains the measurements of m features observed during a given time interval (e.g., during one minute we observe the following features of packet traffic behavior: number of packets, average interarrival time, average payload size, etc.), so that each sample $X_i = (A_1^{(i)}, \ldots, A_m^{(i)})^T$ is described by m features $\{A_1, \ldots, A_m\}$ with their numeric values (in time interval). The list of features is described in Tables 2 and 3.

Let us assume that there are k known classes of interest $(C_1, \ldots, C_k)$ and each sample may be classified in one of the previous classes (e.g., the sample i is an anomaly of class k), i.e., $C :: X_i \rightarrow \{C_1, \ldots, C_k\}$ that defines the assignment of instance X_i to a particular class in C.

The Bayesian inference about the class assignment of an unobserved sample y_i follows the formula:

$$\mathrm{p}(c_j|\mathrm{y}) = \frac{p(c_j)p(y|c_j)}{\sum_{c_j} p(c_j)p(y|c_j)} \tag{1}$$

where $p(c_j)$ denotes the probability that the sample y belongs to class c_j, $p(y|c_j)$ is the probability of y given c_j and denominator is the prior probability of training data – the total probability $p(y)$, where $p(y) = \sum_{c_j} p(c_j)p(y|c_j)$ for all disjoint events.

For classification purposes, we can eliminate the denominator from the original Bayes formula because it takes a constant value for particular sample y_i. Taking into account the independence of random variables y_i leads to the following formula:

$$\mathrm{p}(c_j|\mathrm{y}) \propto \left[\prod_{i=1}^{n} p(y_i|c_j)\right] p(c_j) \tag{2}$$

Table 3
The set of features calculated from flow statistics and used to train the ML algorithm.

No.	Feature	Variants	(Sub)network type
1	An average number of packets forwarded by the end device to the GW	Routing, data	non-IP
2	An average number of packets sent directly by the end device the to the GW	Routing, data	IP, non-IP
3	An average number of packets sent directly by the end device outside of the HAN	Data	IP
4	An average number of packets forwarded (indirectly) by GW to the end device (ED)	Routing, data	non-IP
5	An average number of packets sent directly by the GW to the end device	Routing, data	IP, non-IP
6	A mean payload size forwarded by the ED to the GW (destination)	Routing, data	Non-IP
7	A mean payload size sent directly by the ED to the GW (destination)	Routing, data	IP, non-IP
8	A mean payload size forwarded by the GW directly to the ED	Routing, data	Non-IP
9	A mean payload packet size forwarded by the GW indirectly to the ED	Routing, data	Non-IP
10	An average of the inter-arrival time of all received packets from the end node	Routing, data packets	IP, non-IP

Table 4
Summary of annotation used in anomaly reports.

Anomaly report data for non-IP subnetwork	Anomaly report data for non-IP subnetwork
• Identifier of HG, (annotated information based on information saved in the HG) • Identifier of subnetwork supported by the HG (annotated information maintained by the HG) • Source device vendor, (converted from MAC address), • Routed flow (Yes/No) • Packet flow direction (from the HG point of view) – this parameter carries information about flow direction: (From/to endpoint, ED) and replaces subnetwork addresses • Anomaly cause*	• Identifier of HG, (annotated information based on information saved in the HG) • Identifier of subnetwork supported by the HG (annotated information maintained by the HG) • Source device vendor, (converted from MAC address); • Packet flow direction (from the HG point of view) – this parameter carries information about flow direction (From/to endpoint, ED) and replaces local IP addresses; • Anomaly cause*

* Anomaly cause points to the dominant cause of anomaly occurrence. It is obtained by comparison between values from anomaly sample and reference value in one of the three categories: a number of packets, payload size, and inter-arrival time. In result, this parameter carries binary information about the possible cause of anomaly as the pure numerical values are not comparable.

where y_i is a sample of training dataset C_j is a class and the probabilities are estimated using the training examples. When only the samples belonging to one class (e.g., samples without anomaly) are available, the calculation of $p(C_i)$ is incomputable. Therefore, Eq. (1) requires a modification to learn anomalies modeling if training set consists of normal traffic samples uniquely. The literature shows few examples of enhancements in the Bayes classifier design [26,27]. Most of them try to make the classification process *balanced* as if all classes were represented by a non-zero number of samples in training set *X*. An intuitive approach assumes that the conditional probability of the membership of an observed sample y_i to the *normal* class should exceed a minimum threshold. The value of this threshold t should take into consideration the trained conditional probabilities of the *normal* class and is given as [28]:

$$t = min_{x \in X}\left[\prod_{j=1}^{n} p(A_j)\right] \tag{3}$$

where A_j is the attribute value for the sample X_i and $p(A_j)$ is the probability of the attribute's *i-th* value.

The above classification procedure is carried out independently for flows observed at the different interfaces of HG/controller. During both training and testing phase, the one separate naïve Bayesian classifier is available for each interface of the HG/controller.

The output of this step is a flow classified as anomalous. Since each flow is identified with source and destination addresses (MAC, L3) and available higher layer information, further processing requires this data should be annotated with additional information. The locally detected anomalies are annotated and reported to the central process in the next step.

3.5. Annotation and anomalies reporting

This step encompasses two actions: annotation and reporting of identified anomalies to service provider's datacenter. The first step focuses on replacing information that is only useful in a local area network with information of global importance. Specifically, local network addresses are replaced by identifiers and MAC addresses are replaced with the vendor name. Moreover, anomaly variables are also reported.

Finally, anomaly report includes the following identification features that are derived from features listed in Table 4. Moreover, the report includes information that allows identifying the place where the anomaly occurred. The table summarizes anomaly report data.

Our approach is close to the multi-dimensional cluster proposed in [29] except we do not consider the local addresses as a significant feature for the correlated pattern because being specific to a HAN, it will not appear across multiple subnetworks. Instead of that, we use HG's and subnets identifiers which are together unique in the network.

3.6. Anomalies correlation and central intrusion decision

C-ADS carries out analysis focused on dependencies between anomalies that have been identified at the local level. Anomalies are correlated independently for each subnetwork – technology. In consequence, anomalies reported by HGs are aggregated based on technology and statistical features. The purpose of aggregating data is to detect changes in the behavior of devices that are common to a specific group (i.e., identified with vendor name and technology).

Before describing our correlation algorithm, we first define the context in which anomaly correlation occurs. We propose a central ADS correlating anomalies that are reported by a set of connected HGs. Assuming that they are m HGs served by the central ADS and each HG encompasses j subnetworks denoted as D, i.e.,

$$D = \{d_i | i = 1,\ 2,\ \ldots,\ m\}$$

where each HG d_{ij} is responsible for monitoring traffic to its own HAN interfaces (from subnetworks) and generating anomaly alerts.

We consider a centralized ADS approach, which contains a central node that runs the correlation algorithm. Periodically, each HG reports a set of raw anomaly alerts that are collected from its monitored subnetworks. In particular, the set of reported anomalies can be empty. At the end of a given interval of length T, let n_i denote the number of subnetworks of the i-th HG and l_j anomaly alerts reported by the subnetwork j of the HG d_i. Then the total set of anomaly reports received by the central server in that period is denoted as R, i.e.,

$$R = \left\{r_{ijk} | i = 1,\ 2,\ \ldots,\ m; j = 1,\ 2,\ \ldots,\ n_i,\ k = 1,\ \ldots, l_j\right\}$$

where i – the number of monitored HGs, j – number of subnetworks and k – number of anomaly reports from these subnetworks. Each anomaly report $R_i = (R_1^{'(i)},\ \ldots,\ R_m^{'(i)})^T$ is then a vector describing annotated anomaly and consisting of two variable groups:

- variables that uniquely identify anomaly report (according to Section 3.4), let us denote them as G
- variables that describe anomaly derived from A,

Such that, anomaly report anomaly report $R = \langle G,\ A \rangle$. This approach assumes that centrally correlated anomalies are to be used to identify security breaches. From the service provider point of view, it is important to detect if locally identified anomalies are repeated across the network. Moreover, it is also important to determine how many similar anomalies can be observed across HGs connected to the provider's network. For this reason, the process

of correlating anomalies should aim at finding repeated pattern of similar anomalies that are generated by different HGs. When considering the set of all observed anomalies R, the correlation algorithm should aggregate anomalies that have the same attributes for different devices. As a result, we obtain a structure that maintains counters of occurrences of anomalies in the set R.

To facilitate the update on occurrence counters, for any item r_{ijk} we adopt the tree structure to organize the occurrence counters in R. This intuitive approach is used commonly to perform measurements and has many applications including accounting of traffic engineering, service provision and anomaly detection. Advances on this topic are provided in [30]. Authors proposed a scalable counter architecture that reduces memory requirement and extends estimation range. To classify and record a packet, the counter tree requires minimal memory accesses on average. Adopting this structure to our needs we call the ACTree (Anomaly Occurrence Tree) of r_{ijk}. Each item r_{ijk} can be mapped into an integer. So, a leaf node in the ACTree of r_{ijk} is in the form of $\langle r_{ijk}, \text{Count}(r_{ijk})\rangle$, where $\text{Count}(r_{ijk},)$ maintains the occurrence of item r_{ijk}. over variables G. This means that all occurrences of the r_{ijk} are summed up over the variables in G.

Considering the above we define the correlation algorithm as follows (Algorithm 1):

In the first step, the algorithm creates empty buckets indexed by identifiers of HGs and subnetworks served by these HGs. Next, for each element from anomaly set R: first, it decides whether it is generated by the IP capable subnetwork or not. If true, it is classified using the counter tree for IP capable subnetworks (see Fig. 4), otherwise is classified using decision tree for counting anomalies encountered in non-IP subnetworks (see Fig. 5). Next, anomaly reports are aggregated using an adequate counter tree. It is meant as the counter tree leaf update. Anomalies are assigned to leaves according to values of attributes typical for subnetwork type. Each operation will include incrementing the appropriate leaf and updating all nodes on the path from this leaf to the root.

This method assumes that sequences of monitoring records are different for normal behavior and for attacks, as well as each attack type may be characterized by different monitoring record sequence pattern. In order to find suspicious patterns, we developed a searching algorithm (to be executed in C-ADS) that is able to find all patterns (repeated sequences of monitoring records) from the monitoring dataset.

When correlating anomalies coming from different HAN locations, we focus on anomalies that are caused by external causes (i.e., selected smart devices attacked remotely). Similarly, we also consider anomaly events that impact both locally and externally (i.e., attacks resulting in generating traffic to other HAN smart devices or services on Internet). It should also be remembered that the Home Area Network can use many technologies that connect smart devices. Some of them do not support IP protocol directly and require border gateway functionality to translate protocols between networks (i.e., the ZigBee networking protocol, nor the ZigBee cluster library were designed to support IPv4, and TCP/IP/UDP – see ZigBee specification [31]). Therefore, the correlation process should take into account the following features:

The input data to the alert correlation process are the anomalies reported by the participating HGs. Each anomaly report corresponds to a suspicious flow that has been observed, annotated and reported by a local ADS. We consider that each report contains main features as listed in Section 3.4. The aim of correlation is to aggregate alerts that have common feature values. We assume that there is predefined sets of features across HGs. The correlation algorithm searches for frequent instances of these patterns among the input alerts.

4. Performance evaluation

We evaluate the performance of the proposed system via simulations. The simulations aim at validating the anomaly detection

Algorithm 1
Correlation algorithm: finding occurrences of anomaly reports.

```
#input: set R of anomaly reports
#output: set of aggregated anomalies
Begin
1: // initialize the set of counters that are assigned to leaves of ACTree
2: for each r_ijk in R do
        a. if get_vendor(r_ijk) = nonempty
                i. then get_vendor_from(r_ijk) //non-IP endpoint
        b. else
                i. then get_src_IP_form(r_ijk) // IP capable endpoint
3: if bucket(vendor) not □ buckets then
a. create buckets
4: end if
5: assign_graph_vertex(r_ijk) // Assign the particular bucket to the anomaly graph vertices
6: increment number of anomaly occurrences in the total statistics of anomaly
End
```

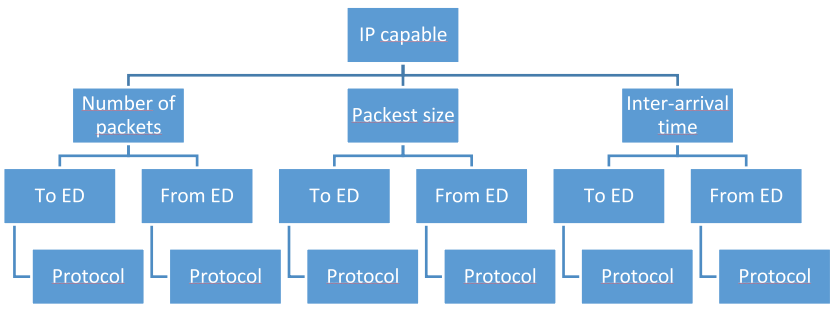


Fig. 4. The counter tree for counting anomalies encountered in IPcapable subnetworks.

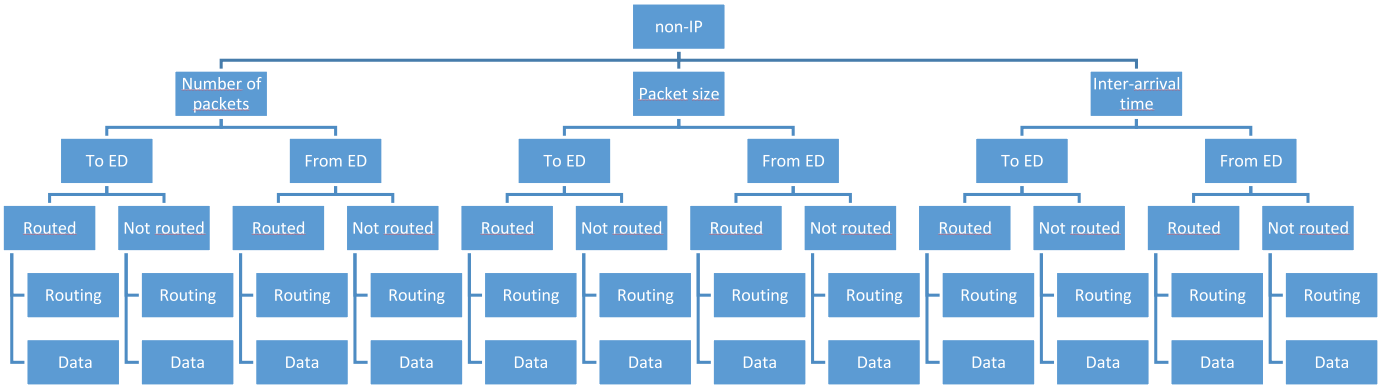


Fig. 5. The counter tree for counting anomalies encountered in non-IP subnetworks.

and correlation mechanism. For this, we simulate the traffic sent by end nodes and test the local anomaly detection based on those data, and in the next step we use the annotated anomalies to correlate suspicious events.

4.1. Simulation scenarios

It is difficult to simulate all the possible scenarios of Home Area Network due to the variety of technologies and protocols used in HAN. Therefore, we have selected two topologies of radio network as representative for most popular deployments nowadays: mesh topology and star topology as shown in Fig. 6. The first one is represented by a network based on the IEEE 802.15 standard (Wireless Sensor Network), and the second one is represented by the WiFi network (IEEE 802.11 g):

- The Wireless Sensor Network (WSN) encompasses 4 nodes and the border router (BR acting also as the concentrator or HG). It was based on ContikiOS protocol stack implementation. Sensor nodes were the source of transmitted data (end points) and the BR (border router) was a sink. The simulation was carried out in the Cooja Simulator because it offers the ready-to-use IEEE 802.15.4 implementations including the set of routing protocol as well as 6LoWPAN based protocol suite. The input data was captured on the BR's interface as raw data.
- The WiFi network consists of 4 nodes operating in client mode and the central node acting in a dual role as an access point and a controller. This part of the simulation was carried out in the Omnet++ simulator. The input data was captured at the access point wireless interface as packet exchange trace.

For the purpose of this paper, we have limited the set of possible HAN technologies to WLAN and 802.15.4 based as representative. Nevertheless, also other technologies can be incorporated. The only requirement is that HAN network interfaces should make able to collect frames that are the subject to further analysis including L2-L3 frame/packet headers.

During simulation, all the nodes numbered from 2 to 5 (see Fig. 6) send messages to one selected node. The same approach was applied in both variants (non-IP and IP subnetworks), as shown in Fig. 6(a) and (b), where the source end points are circles/triangles and the sink endpoint is a square. This actually corresponds to the scenario of smart metering and environmental sensors (i.e., temperature/humidity, light sensors, etc.) where end devices send the data to the concentrator [32]. Each sent message is acknowledged by the data gathering process which provokes concentrator to send the acknowledgment to the sender.

4.2. Simulation dataset

The monitoring records collected at the BR are based on captured packet statistics taken every 1 min. feeding the monitoring record database. Each monitoring record includes statistical information summarizing the 1-minute intervals of observed conditions at BR and the measurements correspond to features A_j, listed in Table 3.

The testing WSN network emulates data exchange typical for sensors and actuators based on sky motes [33]. The payload of the data packets does not exceed 100 bytes. Each data packet sent by a sensor node was acknowledged by the BR. In this context: RPL, 6LowPAN, and UDP are standardized protocols at network (and higher) level. The implemented protocol suite uses 802.15.4 MAC compliant data and acknowledgment packets.

The WiFi network statistics that correspond to features listed in Table 3 are collected on the basis of 1-minute intervals. We assumed that IP communication encompasses protocols ICMP, DNS, TCP, and HTTP. We assumed that the payload size of exchanged messages depends on the protocol and is approximately equal to: 20B, 80B, 800B, respectively for ICMP, DNS, TCP, and HTTP packets.

4.3. Simulation run out

The tests run out as follows: at the beginning, the system is trained by providing one-hour traffic without anomalies (normal traffic) and one-hour traffic with anomalies. For this, the nodes send traffic to a destination (source and destination define the traffic profile in our simulations). The system obtains the measurements from the input interfaces of the destination node (each 1 min) and learns that such datasets are for normal or anomalous conditions. In this step, the detected anomalies are annotated according to rules described in Section 3.4 and used for correlation at the central level.

In the second phase of the tests, the nodes send the traffic and the system receives measurements (from input interfaces of destination node) of traffic profiles and decides whether there are anomalies or not. Once again we perform one-hour test with normal traffic and one-hour test with anomalous traffic in the testing phase. The results of the tests are related to the efficiency of the ADS in finding anomalies in the traffic profiles.

In normal conditions, all nodes are authorized and have assigned network addresses. Normal conditions within the IEEE 802.15.4 based subnetwork indicate that the all nodes have recognized its neighborhood, so only functions for routing and maintenance are performed. The packet traffic related to these

Table 5
Summary of simulation campaign.

Topology	Mesh		Star	
Standard	IEEE802.15.4, 6LowPAN		IEEE 802.11 g	
Protocols	RPL, UDP		TCP, UDP, HTTP, DNS	
No. of nodes	4 end nodes + 1 concentrator (border router)		4 end nodes + 1 concentrator (access point)	
Simulation environment	Cooja Simulator		Omnet++	
Test duration	Normal cond.: approx. 1 h	Anomaly cond.: approx. 1 h	Normal cond.: approx. 1 h	Anomaly cond.: approx. 1 h
Total number of collected samples	641	212 (only selected flows are treated as anomalous)	3500	529 (only selected flows are treated as anomalous)
Traffic rate	3/min	10/min	TCP, HTTP – 20pcks/min DNS – 5 pcks/min ICMP – 2 pcks/min	TCP, HTTP – 40 pcks/min DNS – 15 pcks/min ICMP – 10 pcks/min

functions results from applying the standard algorithms and timers. Moreover, the sending rate of the data packet was set to 3 packets per minute. The normal conditions within the simulated WiFi network assume that end points exchange only administrative and data messages. Moreover, each smart device was able to send TCP, UDP, HTTP, and DNS data packets, where HTTP packets were generated 20 packets/min.

In the anomalous variant, one selected node (triangle node 2 in Fig. 5) acts as infected and increases the traffic 10 times. It results in increased traffic volume that was generated to the sink node as well as increased traffic in the whole network. Although this approach does not assume simulating a specific attack, many security breaches result in increased traffic volume directed to the sink node as well (see Table 1 for details). This example situation was arranged in both simulation subnetworks, i.e. IEEE 802.15.4 network and WiFi network.

Table 5 summarizes the simulation settings.

4.4. Simulation results

In this section we present simulation results encompassing two phases of anomaly detection: the local anomaly detection and the central anomaly correlation. Essentially, the results obtained during the first phase aim to assess effectiveness of the used classification method, comparing the two threshold definitions used for anomaly samples recognition (see Section 4.4.1 for further details). Next, we present results obtained for centralized correlation of detected and reported anomalies. Since this process is performed independently from online detection procedures, it was carried out on the basis of reported anomalies that were gathered during the first phase.

4.4.1. Local anomaly detection

Although the Naive Bayes model does not take into account real dependencies between variables features, it offers fast computation and gives the opportunity to evaluate predictions quickly. This is the main reason for our selection of NB model.

The Naive Bayes model is created on the basis of a summary of the data in the training set. This summary is then used for making predictions. The summary of the training data collected comprises the mean and the standard deviation values for each attribute, by class value. In the simulation, we assumed 8 features for non-IP network and 6 features for IP network as well as two class values. This gives us 16 attributes for each data flow within the non-IP network and 12 attributes for IP network. Fig. 7 presents the mean values of packets sent and received (directly and being forwarded) by the BR to/from sensors in training datasets obtained under normal conditions.

Towards this end, different cases of anomalies were simulated. In the general case, one of the subnet devices (triangle node 2 in Fig. 6) is a flooding device. It causes both increase of routing-related traffic volume and data traffic volume observable in the flow, whose source is node 2 (presented in Fig. 6) and the destination is node 1 (presented in Fig. 6). This flow is defined as "2→1" in the next results. Fig. 7 shows the average number of packets in each flow (defined as source→destination in the figures) for normal and anomaly conditions, whereas Fig. 8 shows the payload size.

Fig. 7 shows that histogram clearly points to disturbances in packet number. Consequently, we may suspect the misbehavior of the (triangle) node even only observing traffic statistics at the HAN interface of the controller/BR. Analyzing Fig. 8, we notice that increased throughput due to the flooding does not imply the payload size, and what is intuitively understood. It also indicates the need for independent analysis of changes in features statistics for effective anomaly detection.

When applying the ML method to search for anomalies in packet traffic, we usually search for sparse events that can be interpreted as a potential attack. Chosen classification method should be sensitive and precise. Therefore, we performed ROC analysis to

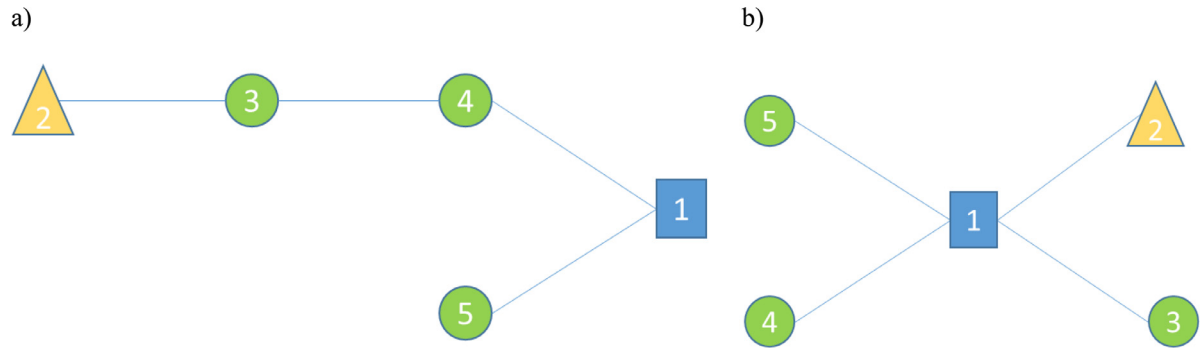


Fig. 6. Testing network topology for simulation of (a) IEEE 802.15.4-based network, (b) WiFi network.

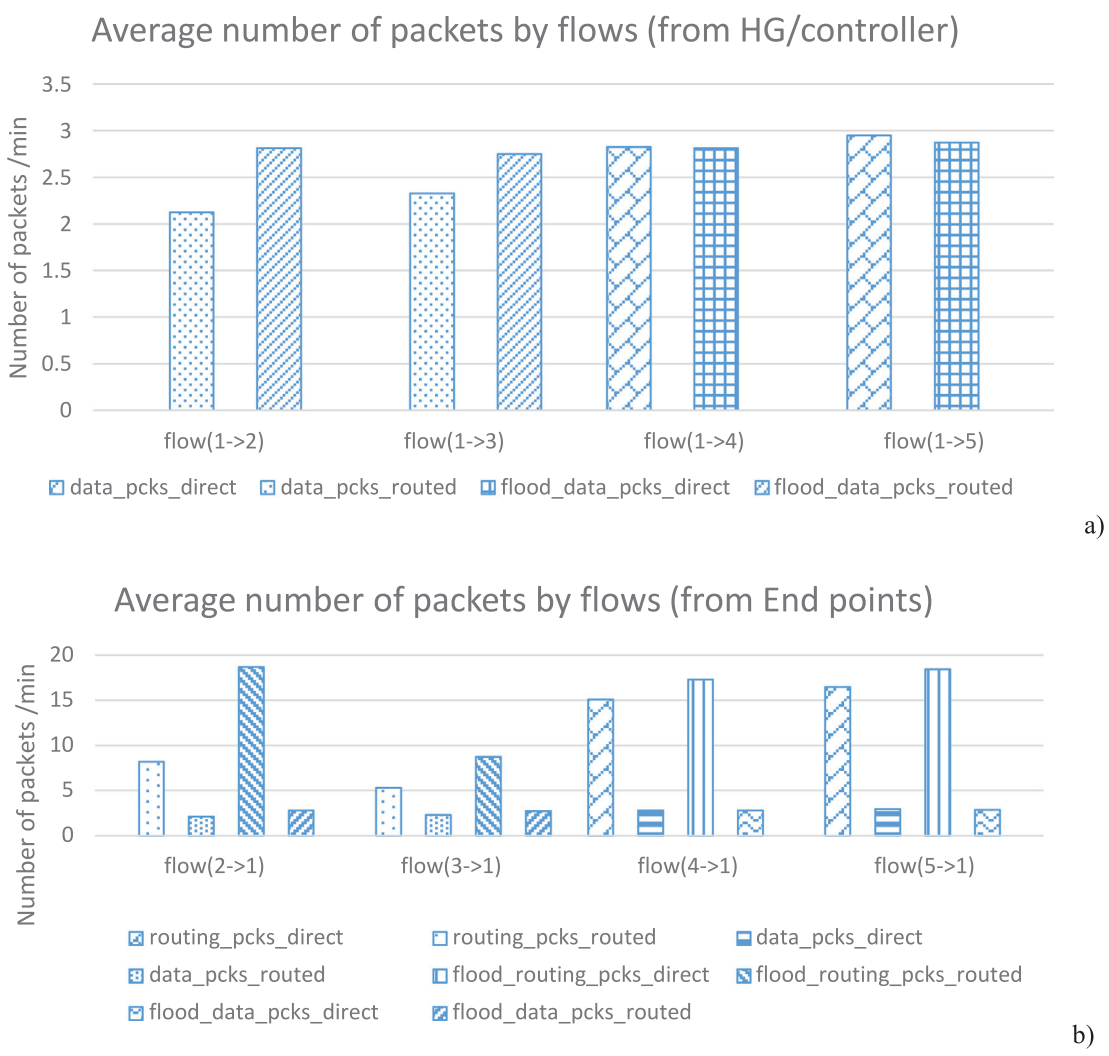


Fig. 7. Histogram of number of packets for normal/anomaly conditions per flow across the non-IP subnet – (a) packet flows directed to the End Device, (b) packet flows directed to the HG/controller.

indicate the accuracy of the classification method proceeded by L-ADS.

In order to provide ROC analysis, the Bayesian algorithm should make a decision on anomaly or not anomaly for each sample. Such a decision is taken on the basis of a given threshold that can be provided in two manners:

- based on the interquartile range (IQR) and assuming that all samples beyond 1,5*IQR are treated as anomalous;
- based on the probabilities of features, which has been defined in formula (3) in Section 3.3. In this case only if $\prod_{j=1}^{number\ of\ features} A_j \geq t$ the test sample is predicted as a member of the positive (anomalous sample) class.

For the scope of verification of the ADS system, the conditions taken during the testing phase were known a priori, and we took measurements under normal conditions. Moreover, we fixed threshold level that cut off the anomaly samples. The testing phase becomes then a verification (of the ADS presented) phase. During the verification phase, we assumed that anomaly packet traffic was generated by the flooding node. The verification phase lasted 1 h. This allowed gathering approx. 60 samples of each flow type.

When the algorithm decides for each (verification) sample whether it is an anomaly or not, we may decide if the algorithm made a True Positive (the probe was taken under positive condition, whatever "positive" indicates, and the algorithm predicted positive condition. For example, if the algorithm classified the probe as "under normal condition", i.e., normal condition in this case was the positive condition, and the probe was really from the normal condition dataset) or False Negative (the probe was taken under Positive Condition, but the algorithm classified the probe as Anomaly Condition).

The test must also be conducted with Negative Condition (if Positive Condition was "under normal condition," we must provide experiments for verification dataset under attack x. In this case the Negative condition is "under attack x"). Repeating the operations with the verification dataset of attack x, we may measure how many False positives (the probe was taken under attack x, and the algorithm classified the probe as normal conditions) and True negatives (the probe was taken under attack x, and the algorithm classified the probe as attack x) the algorithm made.

By normalizing the True Positives and False Positives by 60 (number of samples) for each flow, we obtain the True Positive Rate (TPR), see (8), and the False Positive Rate (FPR), see (9). Then, we may build the Receiver Operating Characteristic (ROC) curves for visualizing the quality of the ML classification. ROC curves

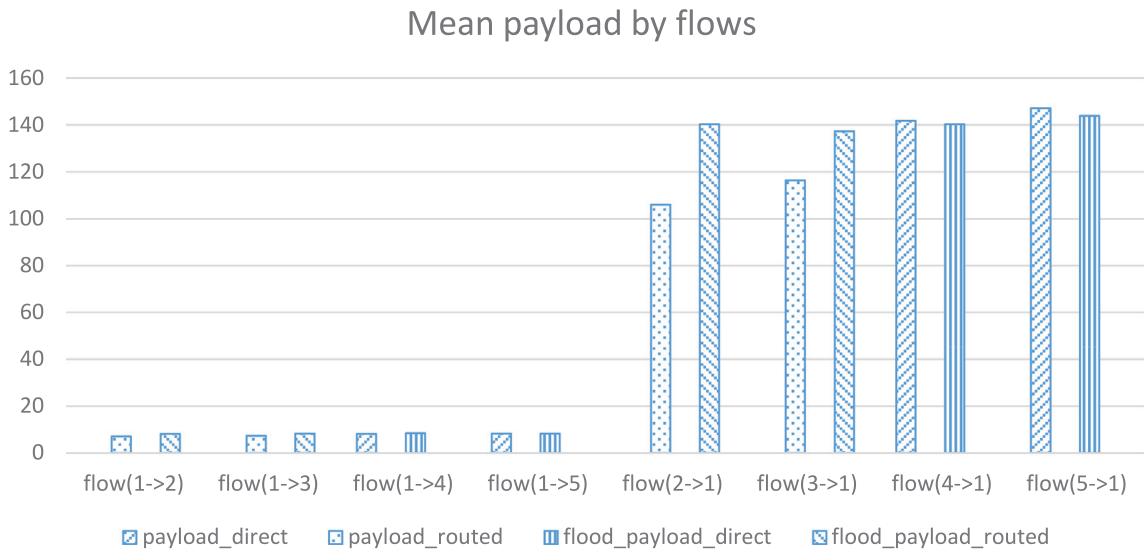


Fig. 8. Histogram of average payload size for normal/anomaly conditions per flow across the non-IP subnet.

present the relation among TPR and FPR for different values of decision criterion ("cut-offs").

$$TPR = \frac{\#\ of\ times\ the\ classifier\ labelled\ an\ attack\ monitoring\ record\ as\ an\ attack}{\#\ of\ attack\ monitoring\ records} \tag{8}$$

$$FPR = \frac{\#\ of\ times\ the\ classifier\ labelled\ normal\ monitoring\ record\ as\ an\ attack}{\#\ of\ normal\ monitoring\ records} \tag{9}$$

Each value of the indicator pair (TPR, FPR) was calculated for various "cut-offs" (10%, 25%, 50%, 75% and 100%). In other words, we calculated TPR and FPR when the threshold required for classifying monitoring record from the validation set as an anomaly is 10, 25, 50, 75 or 100% respectively. Fig. 8 illustrates the ROC curve for the anomaly decision algorithm for selected flows in non-IP subnetwork, so the algorithm analyses the samples and decides whether it is a probe of attack or not. Note that the ROC curve for anomaly detection is made on the basis of considering the anomaly as Positive Condition whereas the Negative Condition is the "under normal conditions." The figure shows the points of the ROC curves for different cut-offs and the trend of the ROC curves. It describes the performance of the classification algorithm, which applies only to selected flows, i.e., 2->1, 3->1, 4->1. For clarity purposes, we did not label the cut-off values in the figure. Respectively, Fig. 10 shows the ROC curve for anomaly decision algorithm for selected flows in IP subnetwork.

The ROC analysis indicates the accuracy of the classification method proceeded by L-ADS and applied for anomaly detection. This information is revealed in shape and in the position of the trend of the curve. The best position of the curve (the most effective decision algorithm) is as close to point (0,1) as possible. In Fig. 9, we may observe that the curves are very near to the diagonal providing from left bottom corner to the right top corner. This diagonal is called no-discrimination and indicates that the selection is binary (two potential solutions) random. Therefore, we may conclude that the applied algorithm fails if any attack has to be detected on the basis of individual monitoring record classification. As we suspected, a successful attack detection requires a deep packet inspection and more computationally demanding traffic analysis which can be proceeded only in powerful machines, which are rare in HANs.

The ROC curves presented in Fig. 10 are located far from the diagonal in comparison to data presented in Fig. 9. It suggests the higher accuracy of applied ML method when larger sets of samples are used in learning and testing phases. Also, the chosen thresholds implied the location of ROC curves in this case. Fig. 9(b) shows the small advantage of the probabilistic approach over IQR analysis.

In summary, using probability-based threshold for anomaly discrimination improves slightly performance of the classification process. It is visible as shifting in the position of ROC curves in comparison to ROC curves for classifiers based on IQR as visible on Figs. 9 and 10.

4.4.2. Anomaly correlation

The centralized anomaly correlation process is performed in C-ADS module independently from online detection procedures and relies on searching similarities among anomaly reports. The difference with the online detection is that this is based on single samples classification while anomaly correlation process uses reported anomalies and tries to find similar records. It expands the anomaly detection process covering more HGs and operates on aggregated information coming from HGs.

During the simulation stage, we use existing anomaly samples obtained from HG. In total, we made 5 sessions during which we obtained 150 samples classified as anomalous. During each session, suspicious samples were annotated to be recognized as anomaly reports coming from 5 different HGs. All generated reports were stored in the repository. In the next step, the stored reports were read in random order. Each anomaly report was analyzed and classified based on Algorithm 1 (see Section 3.5 for details). Information about the classification result was stored in the counter tree. Results are summarized in Fig. 11.

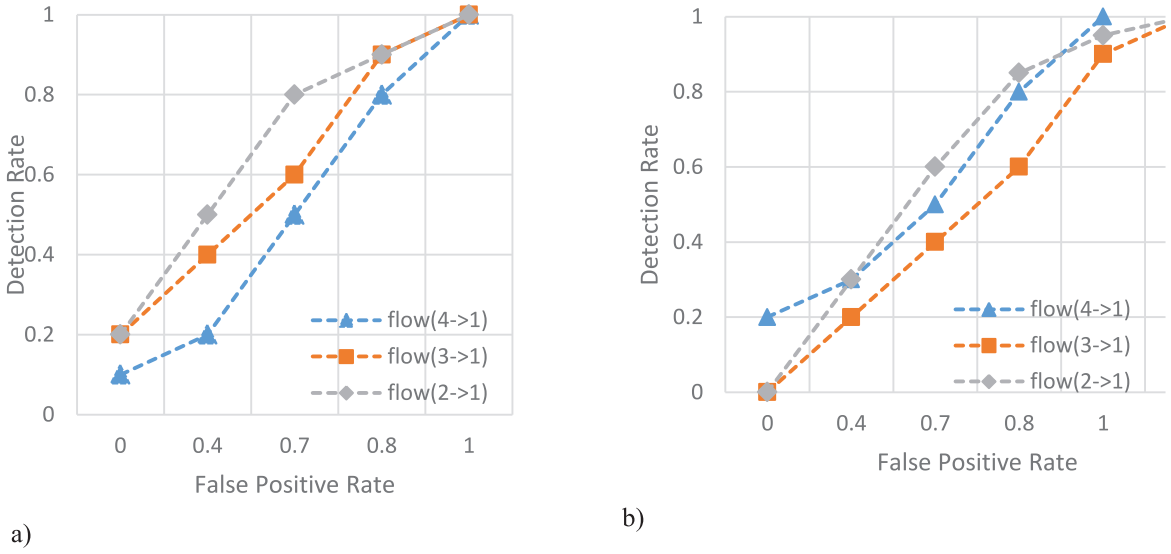
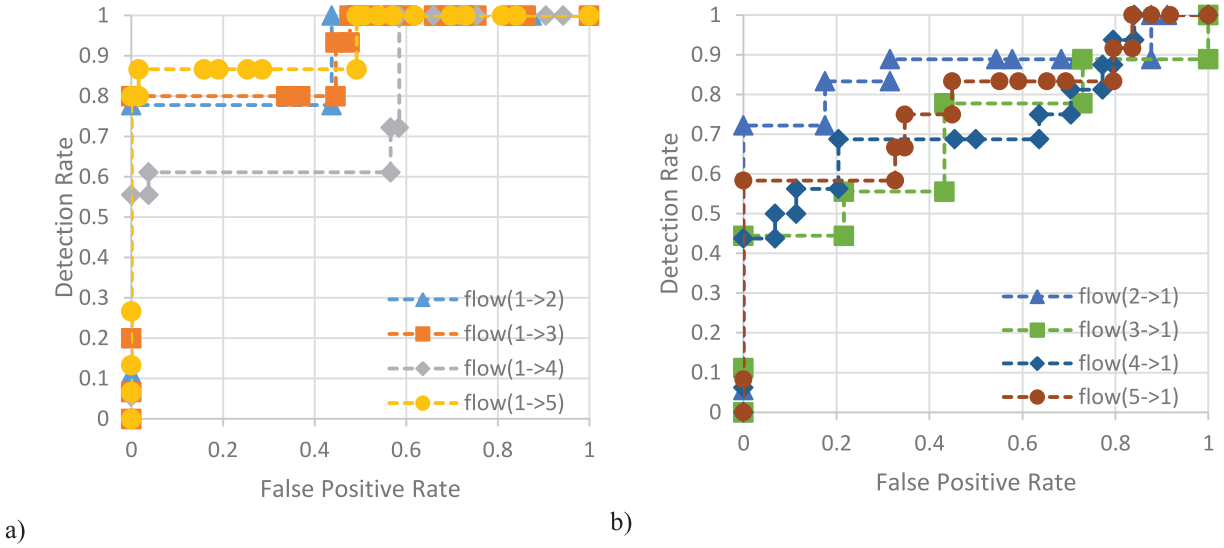


Fig. 9. ROC curve for anomaly detection using NB with a threshold based on: (a) IQR, (b) feature probabilities in non-IP subnetwork.



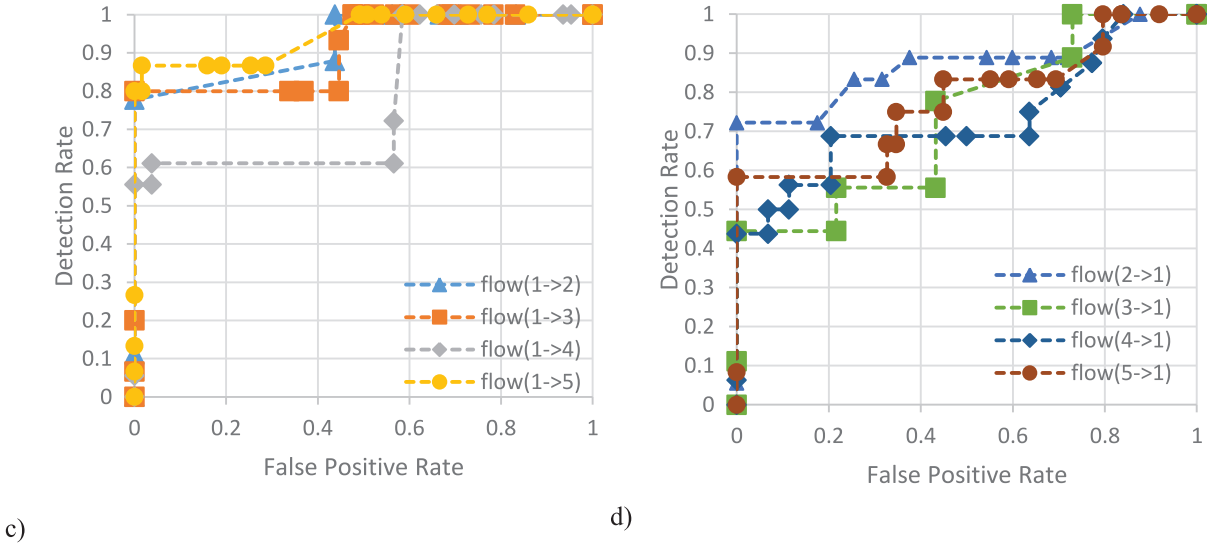


Fig. 10. ROC curve for anomaly detection using NB with a threshold based on: (a and b) IQR, (c and d) feature probabilities in IP-based subnetwork.

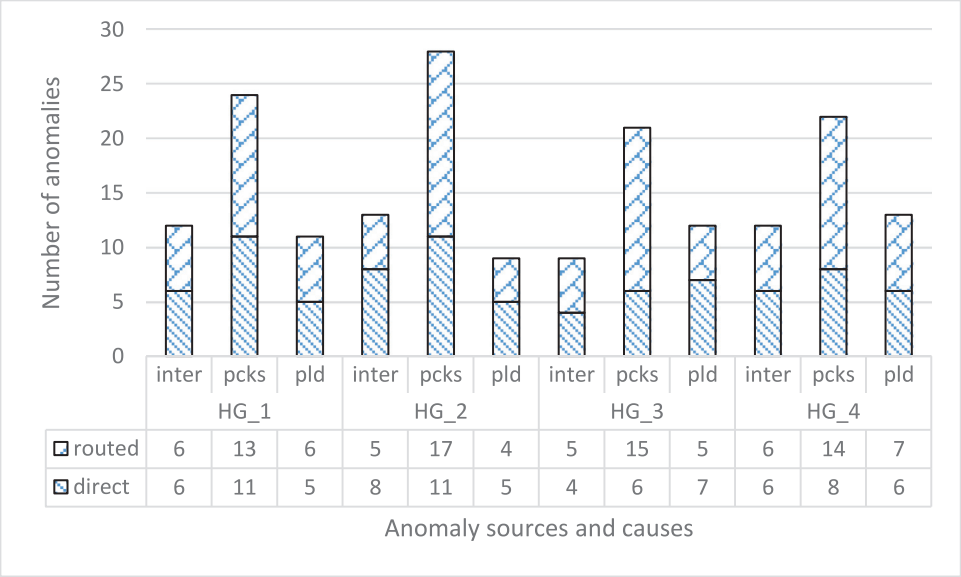


Fig. 11. Correlation results for non-IP network simulation.

The results show the counter values from the ACTree leaves broken down by HGs and anomaly causes (i.e., inter-arrival time – marked as "inter", number of packet – marked as "pcks", and payload – marked as "pld" in the Fig. 11). We aggregated values for routed and direct packet flow statistics for presentation purposes. Because we assumed that simulated anomaly is repeated in each HG, we can also observe similar counter values across the ACTree leaves. The dominant counter values point to increased packet traffic. It can be concluded that the source of anomalies related to increased packet traffic is the end device connected to the HAN. If the anomaly is spread out over the HGs, the service provider should interpret whether it might be the hint of a possible threat or not.

5. Conclusion

The presented results show that locally detected anomalies that can be symptoms of a potential attack can be a rich source of information about the suspicious behavior of endpoints. Particularly, there is a need to increase the effectiveness of detecting attacks that can affect seriously not only the HGs' resources but also service providers' network resources. Effective detection of such attacks requires more computational power for anomaly correlation and endpoint node activity pattern analysis. Some part of the computational power can be moved to the network provider's datacenters. The presented concept of two-tier ADS follows this idea and overcomes operations performed only at the Smart Homes level. However, the presented approach does not exclude intrusion detection performed in other ways – e.g., on the basis of deep packet inspection as matching patterns based on protocol type, addresses, ports, etc.

Acknowledgments

This research work was undertaken under the PolTur FUSE project supported by the National Centre for Research and Development (NCBiR) in Poland. This work is partially supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 117E017.

Conflict of interest

None.

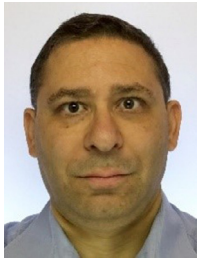
References

[1] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J.A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, Y. Zhou, Understanding the Mirai Botnet, USENIX Security Symposium, 2017.
[2] C. Bormann, M. Ersue, A. Keranen, Terminology for constrained node networks, IETF, RFC, 7228, May 2014.
[3] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, S. Guizani, Internet-of-Things based smart cities: recent advances and challenges, IEEE Commun. Mag. 55 (Sept (9)) (2017) 16–24.
[4] VA. Memos, KE. Psannis, Y Ishibashi, B.-G. Kim, B.B. Gupta, An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework, Fut. Gener. Comput. Syst. 83 (June 2018) 619–628. https://doi.org/10.1016/j.future.2017.04.039.
[5] C. Stergiou, K.E. Psannis, Efficient and secure big data delivery in cloud computing, in: Multimedia Tools and Applications, 76, Springer, November 2017, pp. 22803–22822.
[6] I. Yaqoob, E. Ahmed, M.H. Rehman, A.I.A. Ahmed, M.A. Al-Garadi, M. Imran, M. Guizani, The rise of ransomware and emerging security challenges and solutions in the Internet of Things, Comput. Netw. 129 (2) (Dec , 2017) 444–458 Part.
[7] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, M. Guizani, Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges, IEEE Wireless Commun. Mag. 23 (5) (Nov, 2016) 10–16.
[8] R.A.A. Habeeb, F. Nasaruddin, A. Gani, I.A.T. Hashem, E. Ahmed, M. Imran, Real-time big data processing for anomaly detection: a survey, Int. J. Inf. Manage. (Sep, 2018). https://doi.org/10.1016/j.ijinfomgt.2018.08.006.
[9] M.M. Rathore, A. Paul, A. Ahmad, S. Rho, M. Imran, M. Guizani, Hadoop based real-time intrusion detection for high-speed networks, in: the Proceedings of the 2016 IEEE Global Communications Conference (Globecom), Washington DC, USA, 4-8 Dec, 2016, pp. 1–6.
[10] S. Raza, L. Wallgren, T. Voigt, SVELTE: real-time intrusion detection in the Internet of Things, Ad Hoc Netw. 11 (8) (November 2013) 2661–2674.
[11] P. Jokar, N. Arianpoo, V.C.M. Leung, Spoofing detection in IEEE 802.15.4 networks based on received signal strength, Elsevier Ad Hoc Netw. 11 (8) (2013) 2648–2660.
[12] P. Jokar, H. Nicanfar, V.C.M. Leung, Intrusion detection system for home area networks in smart grids, Second IEEE International Conference on Smart Grid Communications, 2011.
[13] S. Marian, P. Mircea, Sybil attack type detection in Wireless Sensor networks based on received signal strength indicator detection scheme, in: Proceedings of the Tenth Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2015, May 2015, pp. 121–124.
[14] B. Al Baalbaki, J. Pacheco, C. Tunc, S. Hariri, Y. Al-Nashif, Anomaly behavior analysis system for ZigBee in smart buildings, in: Proceedings of the Twelfth IEEE/ACS International Conference of Computer Systems and Applications, AICCSA 2015, November 2015.

[15] P.Y. Chen, S. Yang, J.A. McCann, J. Lin, X. Yang, Detection of false data injection attacks in smart-grid systems, IEEE Commun. Mag. 53 (2) (2015) 206–213.
[16] L. Caviglione, M. Gaggero, J.-F. Lalande, W. Mazurczyk, M. Urbański, Seeing the unseen: revealing mobile malware hidden communications via energy consumption and artificial intelligence, IEEE Trans. Inf. Forensics Secur. 11 (4) (2016) 799–810.
[17] A. Derhab, A. Bouras, M.R. Senouci, M. Imran, Fortifying Intrusion detection systems in dynamic ad-hoc and wireless sensor networks, Int. J. Distrib. Sens. Netw. (2014).
[18] D.E. Denning, An Intrusion detection model, IEEE Trans. Softw. Eng. (2) (1987) 222–232.
[19] https://www.owasp.org/index.php/Main_Page, last visited on the 20 October 2018.
[20] William Fisher (NIST), Sudhi Umarji (MITRE), Identity and Access Management For Smart Home Devices, National Institute of Standards and Technology (NIST), 2016 accessible at: https://www.nccoe.nist.gov/sites/default/files/library/concept-papers/idam-smart-home-concept-draft.pdf .
[21] C. Lévy-Bencheton, E. Darra, G. Tétu, G. Dufay, M. Alattar, Security and resilience of smart home environments good practices and recommendations, December 2015, European Union Agency For Network And Information Security (ENISA), 2015.
[22] C. Stergiou, K.E. Psannis, B.B. Gupta, Advanced media-based smart big data on intelligent cloud systems, IEEE Transaction on Sustainable Computing, 2018 in Press.
[23] C. Stergiou, K.E. Psannis, B. Gupta, Y. Ishibashi, Security, privacy & efficiency of sustainable cloud computing for big data & IoT, Sustainable Computing, Informatics and Systems, Elsevier, June 2018 In Press.
[24] Shahin Farahani, Chapter 1 - ZigBee Basics, in: S. Farahani (Ed.), ZigBee Wireless Networks and Transceivers, Newnes, 2008, pp. 1–24. ISBN 9780750683937.
[25] M.W. Berry, M. Castellanos (Eds.), Survey of Text Mining: Clustering, Classification, and Retrieval, Springer, September 30, 2007.
[26] K. Wang, S.J. Stolfo, One class training for masquerade detection, ICDM Workshop on Data Mining for Computer Security, 2003.
[27] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss, Mach. Learn. 29 (1997) 103–130.
[28] P. Datta, Characteristic Concept Representations Ph.D. thesis, University of California Irvine, 1997.
[29] C.V. Zhou, C. Leckie, S. Karunasekera, Decentralized multi-dimensional alert correlation for collaborative intrusion detection, J. Netw. Comput. Appl. 32 (5) (2009) 1106–1123 ISSN 1084-8045.
[30] M. Chen, S. Chen, Z. Cai, C. Tree, A scalable counter architecture for per-flow traffic measurement, IEEE/ACM Trans. Netw. 25 (2) (April 2017).
[31] https://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeeip/, last visited on the 20 October 2018.
[32] A.P. Plageras, K.E. Psannis, C. Stergiou, H. Wang, B.B. Gupta, Efficient IoT-based sensor BIG Data collection-processing and analysis in Smart Buildings, Fut. Gener. Comput. Syst. 82 (May 2018) 349–357.
[33] http://www.contiki-os.org/index.html, last visited on the 20 October 2018.

Mariusz Gajewski is a R&D specialist at the National Institute of Telecommunications (NIT). He received the M.Sc. degree in telecommunication in 2000 and the M.Sc. degree in business management in 2003 from Warsaw University of Technology. In 2010, he joined the Internet Architectures and Applications Department in NIT. He specializes in technical aspects of network architecture, IPv6 protocol stack testing, Future Internet architectures as well as Internet of Things.

Jordi Mongay Batalla (Ph.D. hab.) received M.Sc. degree from Universitat Politecnica de Valencia (Spain) in 2000 and Ph.D. and hab. degrees from Warsaw University of Technology in 2010 and 2017, respectively, where, nowadays, he has an Assistant Professor position. In 2010–2011 he worked in Telcordia Poland (Ericsson R&D Co.) and later started working in National Institute of Telecommunications (NIT), one of the largest scientific research institutions in Poland. Jordi Mongay Batalla is still with NIT and, from 2010, he is Head of Internet Architectures and Applications Department in NIT. Jordi Mongay Batalla coordinates/coordinated 3 international projects: Chist-ERA DISEDAN, PolLux IDSECOM and "Open standards for mobile services" (a Collaborative project with Mobile Operators: T-Mobile, Orange, Plus and P4) and is/was Technical Coordinator in other international projects (e.g., Eurostars Delta). He took part in several international and national projects, among these: FP6 and FP7 projects: EuQoS, EFIPSANS, COMET, ALICANTE and several COST actions. To ALICANTE, he added through the call Objective ICT-2011.11.3: Supplements to Strengthen Cooperation, where he presented a proposal for extending the Service layer of ALICANTE system (ALICANTE – EU enlarged). He is author or co-author of about 100 papers published in books, international and national journals and conference proceedings.



Albert Levi is a professor of Computer Science and Engineering in Sabanci University, Faculty of Engineering and Natural Sciences, Istanbul, Turkey. He received Ph.D. degree in Computer Engineering from Bogazici University, Istanbul, Turkey, in 1999. Previously, he served as a visiting faculty member in the Department of Electrical and Computer Engineering, Oregon State University and as a visiting professor in the Faculty of Computer Science, Dalhousie University. His research interests include computer and network security with emphasis on mobile and wireless system security, public key infrastructures (PKI), privacy, and application layer security protocols. Dr. Levi has served in the program committees of various international conferences. He also served as general and program co-chair of ISCIS 2006, general chair of SecureComm 2008, technical program co-chair of NTMS 2009, publicity chair of GameSec 2010 and program co-chair of ISCIS 2011. He is editorial board member of The Computer Journal published by Oxford University Press and Computer Networks published by Elsevier.

Cengiz Togay is an assistant professor at the Computer Engineering Department at Bursa Uludag University. He received Ph.D. degree in Computer Engineering from the Middle East Technical University, Ankara, Turkey, in 2008. His research interests include component-oriented software engineering, WebRTC based secure communications, smart cards, privacy, and the Internet of Things (IoT). He has served in the program committees of various international conferences. He is an IEEE member since 2014.

George Mastorakis received his B.Eng. degree from the University of Manchester, his M.Sc. from University College London, and his Ph.D. from the University of the Aegean. He is Associate Professor at the Technological Educational Institute of Crete and Research Associate at the Centre for Technological Research of Crete. His research interests include cognitive radio networks, network traffic analysis, and radio resource management.

Constandinos X. Mavromoustakis received his dipl. Eng. degree from the Technical University of Crete, his MSc from University College London, and his Ph.D. from the Aristotle University of Thessaloniki. He is leading the Mobile Systems Lab of the University of Nicosia. He is vice-chair of IEEE/ regional Cyprus section, and serves as the Chair of Computer Society Chapter of the Cyprus IEEE section.