Architecture of an intelligent Intrusion Detection System for Smart Home

Julian Graf
Dept. Electrical Engineering and
Information Technology
Ostbayerische Technische Hochschule
Regensburg, Germany
julian1.graf@st.oth-regensburg.de

Katrin Neubauer
Dept. Computer Science and Mathematics
Ostbayerische Technische Hochschule
Regensburg, Germany
katrin1.neubauer@oth-regensburg.de

Sebastian Fischer
Secure Systems Engineering
Fraunhofer AISEC
Berlin, Germany
sebastian.fischer@aisec.fraunhofer.de

Rudolf Hackenberg
Dept. Computer Science and Mathematics
Ostbayerische Technische Hochschule
Regensburg, Germany
rudolf.hackenberg@oth-regensburg.de

***Abstract*—Increasing cyber-attacks on Internet of Things (IoT) environments are a growing problem of digitized households worldwide. The purpose of this study is to investigate how an intelligent Intrusion Detection System (iIDS) can provide more security in IoT networks with a novel architecture, combining multiple classical and machine learning approaches. By combining classical security analysis methods and modern concepts of artificial intelligence, we increase the quality of attack detection and can therefore conduct dedicated attack suppression. The architectural image of the iIDS consists of different layers, which in parts achieve self-sufficient results. The results of the different modules are calculated by means of statement variables and evaluation techniques adapted for the individual module elements and subsequently combined by limit value considerations. The architecture image combines approaches for the analysis and processing of IoT network traffic and evaluates it to an aggregated score. From this result it can be determined whether the analyzed data indicates device misuse or attempted break-ins into the network. This study answers the questions whether a connection between classical and modern concepts for monitoring and analyzing IoT network traffic can be implemented meaningfully within a reliable architecture of an iIDS.**

Keywords*—*Internet of Things; artificial intelligence, machine learning, Smart Home, Intrusion Detection System

I. Introduction

7 billion Internet of Things (IoT) devices were used worldwide in 2018 [1]. About 14 percent are consumer devices, including digital assistants [2]. The more devices are connected, the more devices can be attacked and used by botnets or other threats. With the increasing amount of connected devices in a network, it is very difficult for a non-technical user to determine the level of security of the network [3].

Especially with Ambient Assisted Living (AAL) devices and digital assistants, security is extremely important, because highly personal data are collected by such devices. In private households and especially bathrooms are connected sensors that help older people or detect if they fell down [4]. Assistants like Google Home Mini [5] are used to make life easier and control other devices with voice input. The microphones are active all the time to receive voice commands. However, this can be also used to monitor third parties, such as visitors.

To improve the security of the networks, security software like firewalls are needed. Current firewalls are getting extended with intelligent algorithms to keep up with the increasing development of attacks. But there are still new, growing botnets, like Ares [6].

To improve the security level further, Intrusion Detection Systems (IDS) are used. Network based IDS can detect attacks without any additional software on single devices. However, these systems cannot detect every attack. With current artificial intelligence (AI) algorithms, the detection rates can be improved above eighty to ninety percent. Without AI they are just detecting below seventy to eighty-five percent [7]. This difference shows the importance of IDS with AI.

In this paper we are presenting the concept of a network-based IDS, which should be used in a research project to detect and prevent attacks. Smart Home and AAL devices like smart assistants, fall sensors, etc. are the main focus in our network. The IDS goes beyond the state of the art techniques and is equipped with experimental AI algorithms, which are presented in more detail.

The paper is structured as follows. Section II describes the related work. Section III introduces our IoT environment, while Section IV describes the network-based IDS. The architecture of the intelligent Intrusion Detection System is shown in Section V, followed by conclusion and future work in Section VI.

II. Related Work

AI and machine learning (ML) algorithms are part of many software and research projects. Therefore, a lot of approaches for IDS with different kind of AI integration can be found,

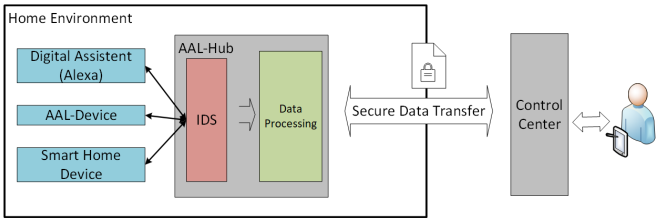


Fig. 1. SEGAL Infrastructure

too. [8] and [9] are both using ML algorithms to improve the detection rate of their IDS. Autonomous machine learning and deep learning algorithms improve the detection rate. However, we are trying to get no false positive results. To achieve this, we need to combine more approaches.

There are existing hybrid methods, like the hybrid IDS from [10]. They are using this approach, because of the high false alarm rate of the neural network. The rule-based component should reduce this rate. Our goal is quite similar, but we are using different AI algorithms. One algorithm for a low false positive rate and the other for the classification of the attack, combined with the classic components.

We found no similar combination of AI algorithms and rule-based components for our zero false positive goal, but a lot of work, evaluating single AI algorithms for IDS, e.g. [11].

III. IoT Environment - Secure Gateway for Ambient Assisted Living

IoT devices are connected to the Internet (directly or via a gateway). Our IoT setup is used as the test environment of “Secure Gateway for Ambient Assisted Living (SEGAL)”. SEGAL is a publicly funded research project. The aim of the project is the development of an AAL service, based on the Smart Meter infrastructure [12]. Data collected within an IoT environment are recorded manually and automatically by sensors and forwarded to an external control center for processing (see Figure 1). Sensors could be digital assistants (Alexa or Google Home Mini, etc.), AAL-Devices (sphygmomanometer, heart rate monitor, etc.) or Smart Home devices (smoke detector, thermostat etc.). The remote communication with the control center takes place via a Smart Meter Gateway (SMGW), which is connected to the household with an AAL-Hub through which the sensors are managed and the resulting data are aggregated.

While communication via the SMGW is considered secure by the Federal Office for Information Security [13], data exchange between the AAL-Hub and sensors offers a potential target because the connection is not necessarily secure. At this point, it must be ensured that no personal data can be tapped and no manipulated data can be infiltrated.

IV. Network-based IDS

A network-based IDS uses different evaluation techniques such as protocol stack verification, application protocol verification, create extended logs, etc. Protocol stack verification can be used to identify invalid flags and data packets. Application protocol verification is used to analyse higher-order protocols such as HTTP, FTP, TELNET etc. to examine and detect unexpected packet behavior. Creating extended logs can be important for analyzing unusual events and monitoring extended network activities [14].

A. Wireless-(network-)based IDS

A wireless IDS can detect the following events [15].

- Unauthorized WiFi and WiFi devices: Most wireless IDS sensors can detect tampered access points, unauthorized endpoints and unauthorized WiFi.
- Poorly secured WiFi devices: Access points and endpoints that do not use the required security mechanisms can be detected by wireless-based IDS sensors. This includes the detection of weak WiFi protocols or protocol implementations and all kind of misconfigurations.
- Unusual usage patterns: Anomaly and signature-based detection methods can be used to detect conspicuous WiFi usage patterns. For example, if more endpoints than usual use an access point or if there is increased network traffic between a device and an access point, this may indicate that devices are at risk or unauthorized persons have gained access.
- (Distributed) denial of service attacks and network disruptions: Denial of service (DoS) attacks can manifest themselves as flooding attacks in which a very large number of malicious messages overwhelm (”flood”) an access point so that it cannot process any further messages. Another type of DoS attack can also occur on a physical level. For example, jamming attacks emit electromagnetic energy on the frequencies of the WiFi. This disrupts the WiFi, which means that no more messages can be transmitted.

B. Detection methodologies

There are three major categories into which intrusion detection types can be divided. Signature-based Detection (SD), Anomaly-based Detection (AD) and Stateful Protocol Analysis (SPA) [15], [16], [17]. The pros and cons of each of these methodologies are shown in Figure 2.

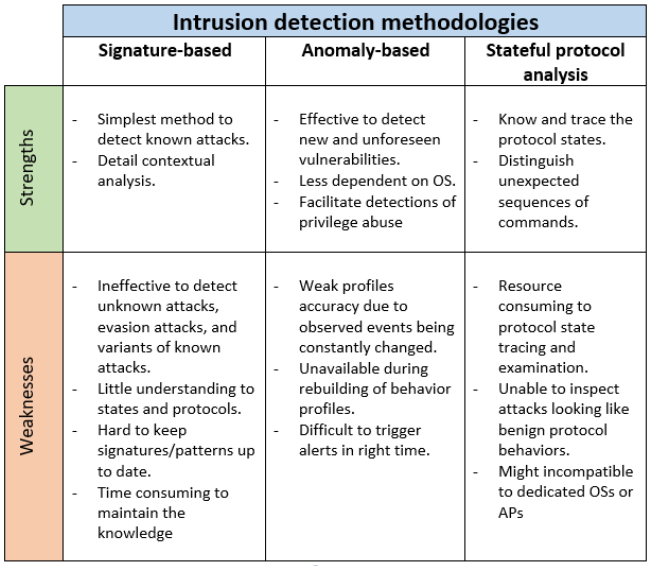
C. Fraud Detection Approaches

In the general description of intrusion detection two main distinguishing features are considered. A distinction is made between anomaly detection and misuse detection.

The separation of possible attacks into categories such as computation-dependent, AI approaches and biological concepts is a very general approach and is difficult to use in practice. Therefore a subdivision into the following subcategories makes sense: static-based, pattern-based, rule-based, state-based and heuristic-based. These subcategories are presented in Figure 3 [15].

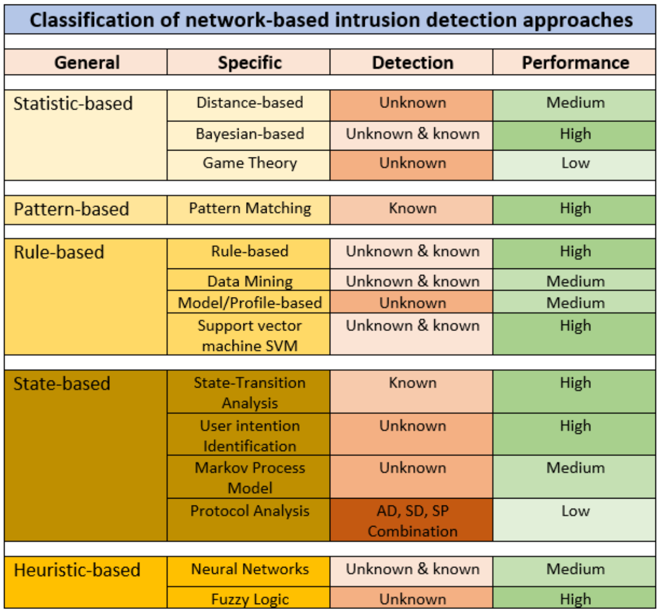
V. Architecture of the intelligent Intrusion Detection System

Our intelligent IDS (iIDS) is based on five layers, as shown in Figure 4. The first layer collects all the data, the second layer



Intrusion detection methodologies			
	Signature-based	Anomaly-based	Stateful protocol analysis
Strengths	- Simplest method to detect known attacks. - Detail contextual analysis.	- Effective to detect new and unforeseen vulnerabilities. - Less dependent on OS. - Facilitate detections of privilege abuse	- Know and trace the protocol states. - Distinguish unexpected sequences of commands.
Weaknesses	- Ineffective to detect unknown attacks, evasion attacks, and variants of known attacks. - Little understanding to states and protocols. - Hard to keep signatures/patterns up to date. - Time consuming to maintain the knowledge	- Weak profiles accuracy due to observed events being constantly changed. - Unavailable during rebuilding of behavior profiles. - Difficult to trigger alerts in right time.	- Resource consuming to protocol state tracing and examination. - Unable to inspect attacks looking like benign protocol behaviors. - Might incompatible to dedicated OSs or APs

Fig. 2. Intrusion detection methodologies [18]



Classification of network-based intrusion detection approaches			
General	Specific	Detection	Performance
Statistic-based	Distance-based	Unknown	Medium
	Bayesian-based	Unknown & known	High
	Game Theory	Unknown	Low
Pattern-based	Pattern Matching	Known	High
Rule-based	Rule-based	Unknown & known	High
	Data Mining	Unknown & known	Medium
	Model/Profile-based	Unknown	Medium
	Support vector machine SVM	Unknown & known	High
State-based	State-Transition Analysis	Known	High
	User intention Identification	Unknown	High
	Markov Process Model	Unknown	Medium
	Protocol Analysis	AD, SD, SP Combination	Low
Heuristic-based	Neural Networks	Unknown & known	Medium
	Fuzzy Logic	Unknown	High

Fig. 3. Classification of network-based intrusion detection approaches [18]

prepares the data and analyses it with different investigation modules. The third layer contains AI algorithms for advanced intrusion detection. The fourth layer analyses the results of layer two and three. The last layer performs some actions, depending on the outcome of layer four.

A. Layer 1: Data-Collection-Layer

The first and so-called base layer, consists of the Data-Collection-Layer (DCL). The DCL is responsible for collecting and storing all data transmitted over the network. In this layer all necessary components are implemented to cut fast and reliably transmitted data packets and to provide them with relevant transmission information.

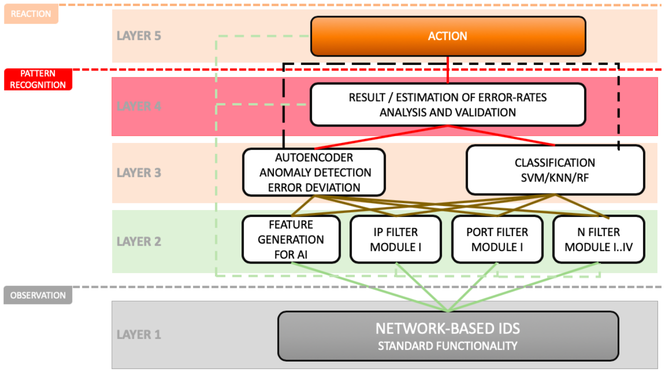


Fig. 4. Architecture of the intelligent Intrusion Detection System

B. Layer 2: Investigation and Preparation Modules

Layer 2 is divided into two main subgroups which are used for analysis and for data preparation / generation. Both parts combine different functions of data collection and processing to a common layer. The data are analyzed with signature-based-, anomaly-based- and stateful-protocol-analysis-methods and then prepared for further ML usage. As described in Figure 3, there are various methods that can be used for this purpose.

1) Investigation Modules: The Investigation Modules analyze the data transmitted over the network. To analyze the data, different approaches from statistic-based, pattern-based, rule-based and state-based methods are used. This generalization is necessary because only after the exploratory data analysis it can be determined which modules provide sufficiently good performance and therefore are suitable for being used in productive environments. Modules combining several different processes are used, too. An example of this is the snort-project [19]. Snort-rules combine rule-based and state-based methods to interpret network data in real time. In summary, it can be said that modules are used to monitor all parameters known in the network in order to guarantee the greatest possible security level.

The IDS uses the following static modules for the detection and analysis of attack / abuse data. The IP-Filter Module determines which IP-addresses are allowed to be used in the network. It analyzes the state of the dynamic host configuration protocol and checks network parameters for violations of thresholds, such as IP range limits. The Port-Filter Module determines which ports are open and available in the network. It logs all ports addressed by any member of the network and checks for port policy violations. N-Filter Module is a placeholder for all additional static modules, like modules for analyzing encrypted payload data or detection of Secure Shell logins. Which further modules are exactly included is decided in the course of the exploratory data analysis.

2) Preparation Modules: Preparation or Feature Generation Modules are used for two major reasons. First, to prepare the data for AI applications and second to extract new relevant ML features. Parameters that are not transferred in the original state of the network packets, but can be derived from them, are processed in the Preparation Modules and added to the database.

One module is used for calculating the real distance of two communicating devices. The calculation is carried out via the source- and destination-IP address. By determining the distance between the network participant and the associated cloud server, it is possible to include this value in our ML modules for classification purposes and to check whether a deviation from the standard state is a deliberate change or an indication of an attempted misuse / attack of the network.

Furthermore there are additional modules which exclusively take care of the preparation of the transmitted network data. The OneHotEnconding module is responsible for data processing / data preparation. For example, text data or data which are not suitable for ML algorithms (such as IP address, protocol names and flags) are revised and stored in a database.

In advantage, the preprocessing of the data reduces the load that arises during the operation of the ML application.

C. Layer 3: Machine-Learning and Deep-Learning Modules

Layer 3 is divided into two ML modules which fulfill different tasks. Both modules are supposed to detect attacks but use different approaches and methods. The AutoEncoder module is designed to detect anomalies regardless of attack data and to provide high accuracy in detecting whether an attack has occurred or not. On the other hand, the second ML approach should classify the different attack types.

1) AutoEncoder Module: Autoencoder are artificial neural networks with a specific architecture and processing logic. Therefore, AutoEncoders can learn efficient representations of input data, called codings, without any supervision. Codings are typically lower dimensional than the input data. AutoEncoders act as powerful feature detectors and can be used for unsupervised pretraining of deep neural networks [20]. However, they can also be used for analysis of unlabeled data. AutoEncoder work in such a way that they try to extract the most important elements from an input set, i.e. to reduce the input set dimension to a smaller dimension and then to extrapolate from this reduced dimension back to the initial state (see Figure 5).

As difference to Multilayer Perceptron, which has a very similar architecture to AutoEncoder, the number of neurons in the output layer must be equal to the number of inputs. AutoEncoder consist of two parts, the encoding or recognition network and the decoding or generative network [20]. The recognition network reduces the number of neurons until it reaches the internal representation layer. From this point the generative network tries to restore the initial input state. For restoring the initial state the decoding segment only uses the information stored in the representation layer.

For several reasons AutoEncoder are particularly suitable for the analysis of IoT data. When developing the IDS, it is easy to obtain standard transmission data from the IoT devices but very difficult to simulate attack data to the extent of the data size required for ML. With AutoEncoders, we can train the ML model to learn how the network operates in daily situations. This means that after the training phase, the procedure can reduce the transcribed data down to the representation layer and then extrapolate it back to the initial state almost immediately through the learned knowledge. The better it has been trained and the more meaningful the variables are, the lower is the error rate in the regression.

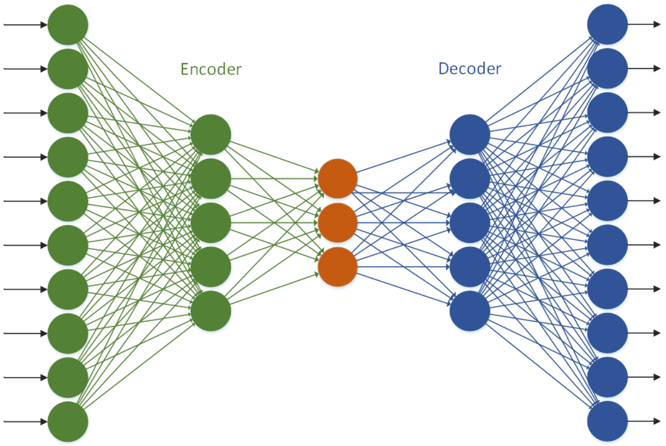


Fig. 5. AutoEncoder

If the network monitored by the IDS is attacked, the AutoEncoder will not be able to reproduce the attack data without raising a very big error rate after decoding the data. Based on this recorded error rate, it can be deduced how different the transmission data were compared to the original learned data. So we can determine whether this data belongs to the standard state of the network by analyzing the extrapolation accuracy of the decoder.

2) Attack types classification module: The attack types classification module is different to the AutoEncoder part. What we are trying to achieve in this module is not only the detection of an anomaly, but also the classification of the attack type that has occurred. The anomaly detection gives us information about an incident in the network that was not planned in this way. The classification procedure takes this data and tries to analyze the exact type of attack. For this purpose, different Machine- and Deep-Learning methods are used to make a reliable statement about the specific attack type depending on the attack data. For example, a distinction is made between data packets belonging to a distributed denial of service (DDoS) attack and a Man-in-the-middle attack.

For ML algorithms there are certain framework conditions which must be adhered to in order to enable a meaningful usage of the procedures. The most important factors are trustworthiness, meaningfulness and consistency of the data. If we look at the different types of attacks, we see that different statements must be made about these data parameters. It is therefore not possible to make a general statement about the data quality in relation to the various attacks and it can be deduced that a single uniform ML procedure cannot achieve meaningful attack classification results if it is trained across attack groups.

We subdivide the different attacks into the following categories: Exploits, Denial of Service Attacks, User to Root Attacks, Remote to User Attacks and Probes [21].

Among these methods, the detection of Denial of Service and Distributed Denial of Service attacks is particularly promising according to the state of the art. At the same time, (distributed) denial of service represents an increasingly large risk due to increasing digitalization. Therefore, the first part of the classification will focus on the detection of distributed denial of service attacks and the following will focus on the detection of the most frequently executed attacks of the other three groups.

D. *Layer 4: Estimation Module*

Since the network packages are analyzed in a differentiated way, the reasonable aggregation of the results is a significant component for the reliability of the system. In view of the many different modules there must be a meaningful evaluation possibility of the partial outcomes, which summarizes the outputs of the different modules and combines them to a final score.

The following verification factors are taken into consideration when determining the static module's validity and power of expressions. The meaningfulness of the static modules is described by the susceptibility to errors, the significance of the examined parameter, the changeability of the parameter regarding its reasonableness and the damaging effect in the event of an attack.

The evaluation and summary of the different quality of the respective ML procedures are collected in a different way. The performance of the procedures is presented using a confusion matrix, which provides information on the performance of the model using the parameters true- / false-positives (TP / FP) and true- / false-negatives (TN / FN). We use the confusion matrix in standardized form to compare models with the same target definition. It is important for the comparison that the procedures were trained, validated and evaluated with the same defined procedural objective, since this is the only way to create serious comparability.

We define the four fields of the confusion matrix as standalone variable fields. We call the field true-positive $alpha$, the field false-positive $beta$, false-negative $gamma$ and true-negative $delta$. Now we determine the absolute difference value between the variables of the same fields, i.e. $|alpha - alpha|$, $|beta - beta|$ etc., and analyze whether they exceed the specified limit value or not. If the value is within the defined limit, both models are suitable for being used in combination. In order to finally assign a meaningful value to the ML model, we use the F1 score.

The F1 score is calculated out of precision (p) and recall (r) [22], [20]:

$$p = \frac{TP}{TP + FP} \qquad r = \frac{TP}{TP + FN}$$

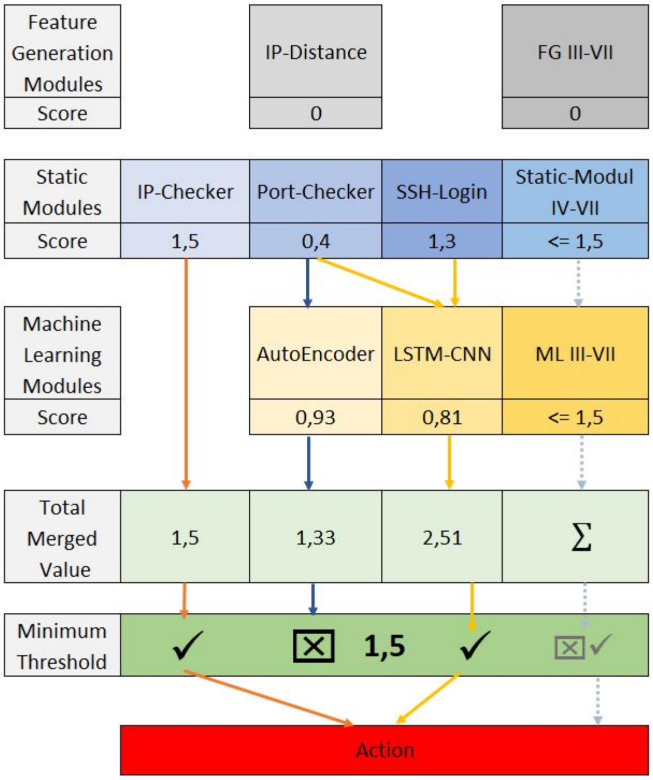


Fig. 6. Example of the scoring system

$$F1 = 2 * \frac{p * r}{p + r}$$

For the calculation of the total score all module values are added up and compared whether this new score is greater than the prescribed minimum score. Modules that were not active are initialized to the value 0. Since there are modules that can already make a hundred percent statement about attacks based on their analysis, these modules are initialized with a value greater than the minimum value. In this case, one module is sufficient to trigger an action by itself.

For the modules which indicate attacks, but whose significance is not sufficient, hints for an attack must be given in the following by additional modules and the overall result of the sum of the respective module values must exceed the minimum limit. An example with fictitious numbers is shown in Figure 6. All static modules can output data to the ML modules, as needed. But in this example, the LSTM-CNN uses only input from the port-checker and ssh-login module.

E. *Layer 5: Action*

The last layer can perform actions according to the results of layer 4. Log entries, notifications, interrupted connections, shutdown of devices or the whole internet connection are possibilities. The lower the false positive rates are, the easier it is to intervene in the operation of the network. Otherwise only false alarms are generated and the network is unnecessarily affected.

In our test environment we do not cut down connections after detecting an attack. We just want to collect data. In future applications, the IDS can hold back the data for the analysis and then decide if the data should be transferred to the internet or into the network. With this approach, data loss of private data can be avoided. This is only possible for a small kind of attacks, because the IDS needs to work really fast. Otherwise the network becomes far too slow. We assume, only the AutoEncoder and the non AI modules can be fast enough.

The classification of the attacks carried out on the network is particularly relevant in order to initiate dedicated security measures for damage limitation / damage prevention and to be able to continue to guarantee the operation of the network as far as possible. Once the nature and severity of the attack on the network is known, specific security measures can be taken to deal with it. Most of these individual safety techniques can be carried out during operation. This prevents a total network failure.

VI. Conclusion and future work

With two different AI algorithms, one for the detection (AutoEncoder) and one for the classification of the attack, the iIDS should improve the detection rates. We combine these AI results with static modules, to get the best results out of all data. In this paper we presented the architecture, which will be implemented and tested in our future work.

The final modules in layer two and layer three can change, depending on the necessary data and the testing results. We use the SEGAL project environment to evaluate the quality of this architecture. As described in layer five, the final product can use the results to perform an action based on the attack detection.

References

[1] (2019, Oct.) State of the IoT 2018: Number of IoT devices now at 7B Market accelerating. IoT Analytics. [Online]. Available: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

[2] (2019, Oct.) IoT trend watch 2018. IHS Markit. [Online]. Available: https://cdn.ihs.com/www/pdf/IoT-Trend-Watch-eBook.pdf

[3] S. Fischer, K. Neubauer, L. Hinterberger, B. Weber and R. Hackenberg, "IoTAG: An Open Standard for IoT Device IdentificAtion and RecoGnition", The Thirteenth International Conference on Emerging Security Information, Systems and Technologies, 2019, in press.

[4] W. L. Zangler, P. Panek and M. Rauhala, Ambient assisted living systems - the conflicts between technology, acceptance, ethics and privacy. Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2008.

[5] Google Ireland Limited (2019, Oct.) Google Home Mini. [Online]. Available: https://store.google.com/product/google_home_mini

[6] C. Cimpanu. (2019, Oct.) A new IOT botnet is infecting Android-based set-top boxes. ZDNet. [Online]. Available: https://www.zdnet.com/article/a-new-iot-botnet-is-infecting-android-based-set-top-boxes/

[7] N. A. Alrajeh and J. Lloret, "Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks", International Journal of Distributed Sensor Networks 9.10, p. 351047, 2013.

[8] J. Cannady, "Next generation intrusion detection: Autonomous reinforcement learning of network attacks.", 23rd national information systems security conference, pp. 1-12, 2000.

[9] A. Shenfield, D. Day and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks", ICT Express, 4(2), pp. 95-99, 2018.

[10] S. Koutsoutos, I. T. Christou and S. Efremidis, "An Intrusion Detection System for Network-Initiated Attaclcs Using a Hybrid Neural Network", Artificial Intelligence Applications and Innovations, Springer US, pp. 228-235, 2006.

[11] H. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey", Applied Sciences, vol. 9, no. 20, p. 4396, 2019.

[12] DATACOM Buchverlag GmbH. (2019, Sep.) AMI (advanced metering infrastructure), 2013. [Online]. Available: http://www.itwissen.info/AMI-advanced-metering-infrastructure-AMI-System.html

[13] Bundesamt fuer Sicherheit in der Informationstechnik. (2019, Sep.) BSI - Smart Metering Systems - Smart Metering Systems, 2019. [Online]. Available: https://www.bsi.bund.de/DE/Themen/DigitaleGesellschaft/SmartMeter/smartmeter.html

[14] S. Sapiah, Intrusion detection and prevention, Information Systems Department: Course Technology CENGAGE Learning, 2004.

[15] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)", NIST Special Publication 800, p.94, 2007.

[16] P. Stavroulakis and M. Stamp, Handbook of information and communication security, Springer, 2010.

[17] S. Axelsson, "Intrusion detection systems: a survey and taxonomy", Chalmers University of Technology, pp. 1-27, 2000.

[18] H. J. Liao, C. H. R. Lin, T. C. Lin and K. Y. Tung, "Intrusion detection system: A comprehensive review", Journal of Network and Computer Applications 36, pp. 16-24, 2013.

[19] Cisco. (2019, Nov.) Snort - Network Intrusion Detection and Prevention System, 2019. [Online]. Available: https://www.snort.org

[20] A. Geron, Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems, O'Reilly Media, Inc., 2017.

[21] K. K. R. Kendall, A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, Massachusetts Institute of Technology, 1999.

[22] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation.", Springer, pp. 345-359, 2005.