# Context-aware local Intrusion Detection in SCADA systems: a testbed and two showcases

Justyna J. Chromik[1], Carina Pilch[2], Pascal Brackmann[2], Christof Duhme[2], Franziska Everinghoff[2], Artur Giberlein[2], Thomas Teodorowicz[2], Julian Wieland[2], Boudewijn R. Haverkort[1], and Anne Remke[2]

***Abstract*—This paper illustrates the use of a testbed that we have developed for context-aware local intrusion detection. This testbed is based on the co-simulation framework Mosaik and allows for the validation of local intrusion detection mechanisms at field stations in power distribution networks. For two cases, we show how this testbed assists with studying the effectiveness of two local IDS mechanisms under different kinds of attacks.**

***Index Terms*—SCADA, intrusion detection, distributed control**

## I. Introduction

The shift to smart grids and the inclusion of more locally generated renewable resources is turning traditional one-way power grids into so-called more and more complex *cyber-physical systems*, in which the physical world and the cyber world heavily interact. The process of energy generation and transportation requires distributed control mechanisms, which are often implemented in *Supervisory Control and Data Acquisition* (SCADA) systems, which are not necessarily secure [1], [2]. In this paper, we address how we can evaluate the security of such distributed SCADA-based control systems by using only (or primarily) locally available information. In line with previous works, we take into account not only the state of the network itself [3], but also the state of the physical controlled system [4], [5], [6]. This form of context-awareness promises to be more effective. Distributed control can be especially dangerous to the power grid's stability, e.g., [7] shows that bad demand response management of the future smart grid might cause a blackout in a neighborhood. Especially, coordinated attacks on many neighborhoods could damage the whole system. Recently, [8] and [9] proposed an approach for intrusion detection that takes this into account; [8] detects malicious *photovoltaic panel* (PV panel) production using weather context information, whereas [9] detects incorrect measurements of voltage based on the knowledge of the topology of the distribution system. Moreover, [5] proposed an approach for insider threat detection by assigning the trust values to the smart grid devices, based on the values describing the state of the physical system, and [4] computes the state of the entire power system within a trusted *Intrusion Detection System* (IDS) and informs the IDSes in the remote field stations about the undesired commands.

It is often difficult to test or validate approaches as sketched above on relevant scenarios in real-life. Hence, most of the available security mechanisms are tested in simulation environments where mainly network traffic properties of the control network are being analyzed. True testbeds with expensive hardware in place are often difficult to access.

Next to the above, we have recently presented a *local* monitoring approach [6], [10], which can be used as additional security measure. This paper proposes a testbed for validating such a local approach, based on co-simulating the SCADA control network and the power system and monitor both, the SCADA network and the physical process. While we described the general capabilities of the proposed framework in [11], this paper demonstrates its capabilities in two specific scenarios: (i) the detection of tampered *Remote Terminal Unit* (RTU) readings, (ii) the detection of tampered PV production readings. In these scenarios we investigate two protocols, Modbus/TCP [12], a popular protocol in industrial control systems, and MQTT [13], which gains increasing popularity in, e.g., the Internet of Things. This paper shows that local measurements taken at an RTU or at a smart home can be used to identify intrusions that may have passed other high-level security mechanisms like firewalls or centralized IDS.

Relevant scenarios from literature vary from voltage and current control to monitoring of smart houses. Ideally, the detection of malicious readings/commands must be done at the field location, based on measurements taken as close to the source as possible [14]. Existing testbeds often simulate the entire power system together with the entire communication network [15] and use licensed software [16]. In contrast, we use the well-maintained and open source co-simulation framework Mosaik [17], [18], which is easily extensible to incorporate various existing simulators, and we co-simulate only the necessary smart grid elements: a power distribution system with the investigated controllers within the relevant (sub-)system.

The rest of the paper is organized as follows. Section II explains the general concept of the testbed. Section III and Section IV present the two scenarios under investigation. The paper concludes in Section V.

## II. Local Intrusion Detection with Mosaik

Section II-A describes our testbed, II-B the types of attacks investigated and II-C the IDS approach.

[1]Design and Analysis of Communication Systems, University of Twente, The Netherlands {j.j.chromik, a.k.i.remke, b.r.h.m.haverkort} at utwente.nl

[2]Safety-critical systems group, University of Münster, Germany {carina.pilch, p_brac01, c_duhm02, f_ever02, a_gibe01, t_teod01, j_wiel04, anne.remke} at uni-muenster.de

978-1-5386-4055-5/17/$31.00 ©2017 IEEE

### A. Mosaik

Mosaik is an open source co-simulation framework written in Python (under GNU LGPL) [17], using the discrete-event simulation library based on SimPy. With the provided API, different simulators can be connected, while Mosaik interfaces their data transfer and tracks the execution order.

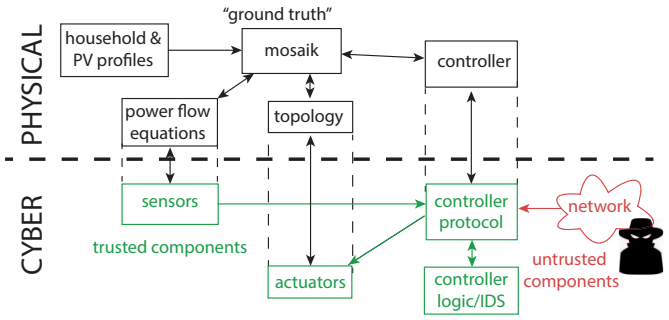



Fig. 1: Scheme of the testbed: simulated "ground truth" above dashed line, and trusted (green) and hacker tools (red) below.

We extend the available Mosaik simulators to allow for changes in the topology and by including a communication network for control. We use the basic power system as provided in the demo, which consists of houses, PV panels and a distribution network (buses and branches). The physical properties of the power distribution system are described in a JSON file. We expand the description of branches to include their state: online or offline. Furthermore, we add a topology simulator, which determines which buses are isolated, based on information from controller, and is able to adjust the state of buses and branches. This adjusted model is then fed into the power flow equation simulator.

The control network consists of a controller using Modbus/TCP [12] or *Message Queue Telemetry Transport* (MQTT) [13]. We enable the integration of the controller into the physical system by making the following connections. The RTU determines the values measured by sensors, based on the data obtained from the Mosaik power flow equation simulator. Then, based on the internal logic of the RTU, local control actions may be taken, e.g., the RTU might open or close a power line. Figure 1 illustrates the general scheme of the testbed. The upper, black elements indicate the most significant simulators co-simulated in Mosaik. The topology and controller as well as the elements below the dashed line are part of the proposed extension to the framework.

### B. Threat model and attack type

In the following, we assume that the controller (shown in Figure 1) is connected through some networks to a central SCADA server. We focus on threats that access the system via the network, either as insider attack from the central SCADA server or as third party (hacker) attack, that can perform different types of attacks, as follows:

**Random attacks** are launched by performing random actions on a random element of the power system, e.g., opening or closing a random power line or changing a measurement of a random sensor.

**Deterministic/targeted attacks**, in which the attacker launches an attack to a pre-determined target, for example, opening an especially critical power line.

**Semantic attacks** are engineered to bypass known defense mechanisms, such as stealthy attacks bypass bad data detection mechanisms [19]. These attacks require in-depth knowledge of the system and could be performed, e.g., by an insider who understands well how the system works.

### C. Intrusion detection approaches

Our aim is to illustrate the benefits of a context-aware IDS mechanism that uses information on (the state of) the controlled physical process and is not just based on (the state of) the SCADA network traffic. Moreover, with the energy transition leading to automation and remote control of field stations, a distributed and local IDS becomes increasingly important to prevent incidents like the one in Ukraine in December 2015 [2]. Hacking into a central control room allowed hackers to control field stations, leaving more than 230 000 citizens without power. Therefore, in future smart grids the control needs to work in a distributed way and we aim to perform this detection based only on local measurements.

As indicated in Figure 1, we propose to position a local IDS within the logic of the controller, which locally determines whether a control command or measurement reading is malicious or faulty. The decision is based on comparing local measurements to the physical restrictions and safety requirements, as presented in [6], [10]. We address two specific scenarios in what follows, adapted from [5] and [8], whose details will be presented in the following two sections.

## III. RTUs attack scenario

The first scenario is based on the topology shown in Figure 2, inspired by a real topology of a cooperating partner, whereas some elements have been added for the purpose of our analysis. RTUs (indicated by a red circle) are placed at those buses that are connecting more than two other buses. The controller uses the Modbus/TCP protocol based on PyModbus[1]. Attackers use: (i) a self-developed simple command line shell, allowing them to read and overwrite sensor values and states of switches, and (ii) a script interpreter, enabling more complex and automated attacks via *for*- and *if-then-else*-statements. This allows them, e.g., to slowly increase values of sensor readings until a critical threshold is reached. In what follows, we explain the local detection mechanisms and the performed attacks.

### A. Intrusion Detection

The IDS placed on an RTU checks whether (i) the sensor readings are not compromised and (ii) the outcome of a command would not bring the system into an undesired state.

[1] http://pymodbus.readthedocs.io/en/latest/index.html

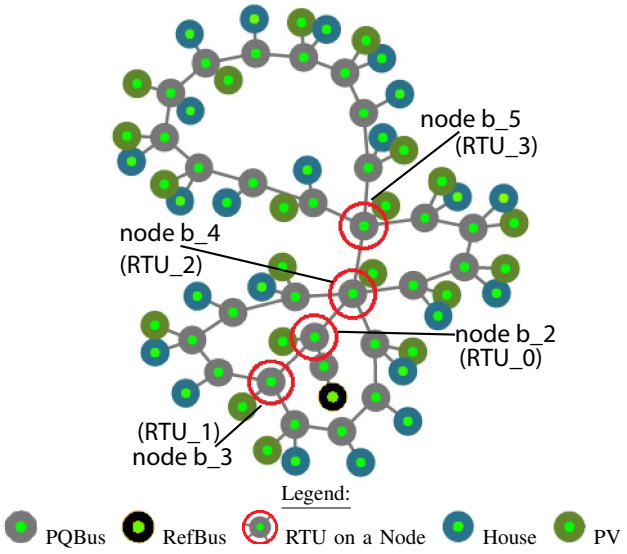



Fig. 2: Power distribution topology.

*Trusting sensors*

Every RTU has a set of connected sensors, e.g., in Figure 2, the RTU on node $b_4$ is connected to five other nodes, hence, five current sensors are attached. Every RTU keeps track of an abstract *internal trust value* (ITV) assigned to each of the sensors. If the ITV exceeds the threshold value $\text{ITV}_{th}$, the sensor is marked as untrusted. The RTU broadcasts to other RTUs if a sensor changes to *not trustworthy*. Such untrusted sensors will then be ignored until an operator restores the trust.

Similarly to [5], the voltage angle is checked between adjacent RTUs in one step and the change of voltage magnitude and current is checked at each power line between consecutive steps. If the checks described above are satisfying, the sensors are considered trustworthy. Otherwise, the ITV is increased and warnings might be issued. As proposed in [10], Kirchoff's current law is checked at each discretized time point for each node. However, as the ITVs of all current sensors connected to same node are increased, a sensor that is working correctly is potentially marked as untrusted, because other sensors at the same node have been attacked.

*Assessing commands*

The RTUs monitor all connected buses and use safety requirements as described in [10] to ensure that also within a decentralized energy management scheme a safe grid state is maintained in all branches. An incoming command is not executed if either of the following two requirements is violated. The first requirement states that the current of all power lines has to stay below a predefined maximum cut-off current value. The second requirement states that voltage stays within 10000V $\pm$10%, as in [9].

### B. Detection of attacks

The script interpreter changes sensor readings to create attacks for different levels of system knowledge.

**Random attack**: A random RTU and a random sensor connected to that RTU are chosen. Then, the value read by the sensor is changed by a random percentage. Figure 3 shows the resulting plots for the RTUs in Figure 2. Note that each line in a plot corresponds to a connected current sensor. The graphs representing this attack take the form of step functions. The simulation reveals that before large differences in successive values can mark sensors as untrusted, the application of a check based on Kirchhoff's law detects the intrusion.

**Deterministic attack**: This attack targets sensors located at the RTU of node $b_2$ in Figure 2. It increases the measurement of $sensor_2$, measuring the current on the branch connecting nodes $b_2$ and $b_4$, and of $sensor_4$, measuring the current on the connection between $b_2$ and $b_3$, by $1\%$ in each simulation step. In Figure 4, this continuous increase can be seen as a steady increase of the black and red graph. The attack starts at simulation time $47700$ and the IDS senses the violation of Kirchhoff's law immediately; after three simulation steps, the ITV of the attacked sensors reach the threshold value. As soon as the attack is detected, all sensors on the attacked node are marked as untrusted, which yields two sensors incorrectly labeled as untrusted.

**Semantic attack**: Using the hacker tools, a semantic attack can be performed, e.g., such that the Kirchhoff's law is too inadequate to identify the attack. This can be done by tampering measurements, e.g., with multiple sensors, such that the sum of the changed measurements remains the same. This ensures that Kirchhoff's law still holds for all sensors connected to one bus. To trick the proposed IDS, we could also run a first attack that will definitely be detected, so that the IDS will mark some nodes as untrusted. Later, these nodes will be ignored by the IDS and by other RTUs. This impairs the IDS's control functions and challenges grid stability. These types of attacks are complex and require a high level of insider information and are out of the scope of this paper.

*Output of the proposed IDS*

Figure 5 shows the IDS on-screen output corresponding to the deterministic attack plotted in Figure 4. We highlighted three warnings given by the RTU to the operator. The first (marked in red) indicates that Kirchoff's law was violated at bus $b_2$. It is followed by warnings about the sensors connected to that bus. Since Kirchoff's law was violated on the bus, any of the sensors connected to it could be malicious, hence, their ITV are increased. The second one (blue) indicates that Sensor 2 is marked as untrusted, because its ITV exceeds the corresponding threshold $\text{ITV}_{th}$. Finally, the third one (green) shows that warnings about the malicious sensors are sent to the neighboring RTUs: RTU_2 and RTU_1.

## IV. Smart House Scenario

We describe a possible attack scenario on a smart house, assuming that the controller presented in Figure 1 uses the MQTT protocol to control the (dis)charging process of a local battery storage. The topologies used in this scenario are shown in Figure 6. The first one presents the power grid for the IDS

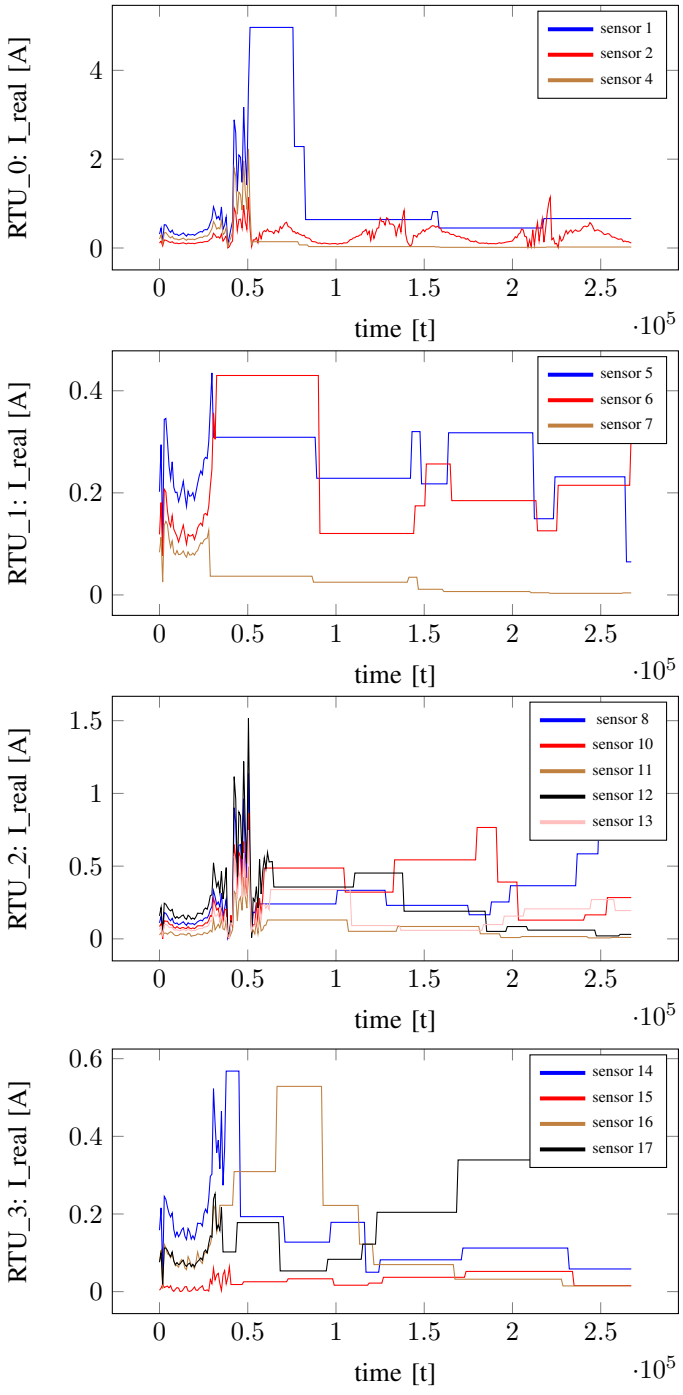



Fig. 3: Graphs of the current readings ($I_{real}$) against simulation time; from top: $RTU_0$ - $RTU_3$.

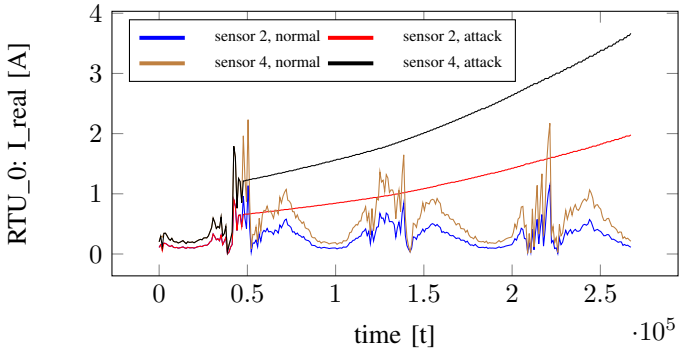



Fig. 4: Current readings ($I_{real}$) for sensors 2 and 4 in a simulation without attack and with a deterministic attack.

prototype. We connect an additional PV panel and household directly to the PQ bus. The smart house, indicated in dark red, has a PV panel and a (consuming) household connected to it. The scenario in Figure 6b is used to illustrate a self-learning IDS. The smart house consists of two PQ buses and a RefBus, which is directly connected to the power grid. Again, the smart house is a prosumer with a PV panel and local battery storage. In the following sections, we explain the local detection mechanisms and the performed attacks.

### A. Intrusion Detection

In [8], a mechanism embedded in the power network is presented, which shows how a faulty or malicious power supply reading can be detected using a neural network. We examine the production of a smart house and alert the owner of that smart house. This allows the detection of attacks at the earliest possible time, without extending the existing power grid with new safety features.

We present two approaches. The first one is a *prototype IDS*, which has 100% accurate predictions and a predefined deviation $\varepsilon$ for solar production. In this IDS, the valid range for production is then $[\text{prediction} - \varepsilon; \text{prediction} + \varepsilon]$. All values in the interval are accepted by this IDS without issuing a warning.

The second one, a *self-learning IDS*, adjusts predictions on-the-fly using values of the previous hour. Assuming that the production follows a Normal distribution, the estimator was trained with the power production data of the previous hour. For learning purposes, each day is separated into 3 phases: (i) a *pre-phase* without power generation, from 00:00 a.m. up to the time of the first generation; (ii) a *prod-phase*, when PV panels generate power; and (iii) a *post-phase*, which starts 10 minutes after the last generation and ends at 11:59 p.m.

Training the estimator with data from the production phase determines the expected mean value of the Normal distribution. The variance and standard deviation are computed as the sample variance $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$, whereas $n$ is the number of values, $x_i$ the value of the $i^{th}$ minute and $\overline{x}$ the expected mean. Next, we used a hypothesis test with a significance level of 5% to analyze the current power production of the PV panel. The validity range of the production is then given by $[\mu - 2.5 \cdot \sigma; \mu + 2.5 \cdot \sigma]$. In case that a value does not lie inside the range of the current hour, a warning is sent to the user via the MQTT protocol.

### B. Attack scenarios

Depending on the knowledge of the attacker, deterministic and semantic attacks are performed by manipulating the measurements of the PV panel power production.

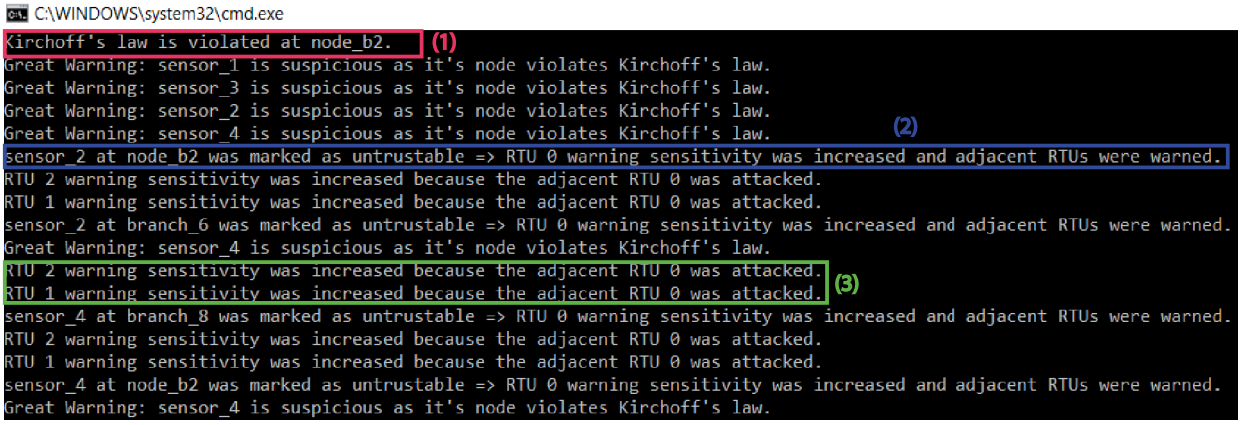
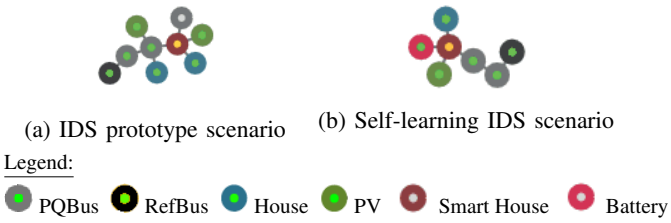



Fig. 5: IDS output for the deterministic attack.




(a) IDS prototype scenario (b) Self-learning IDS scenario

Fig. 6: Scenarios for the IDS.

**Deterministic attacks** - We simulate the power grid with the PV panel data provided in the Mosaik demo scenario[2]. As attack, we alter the production readings of the PV panel. While in the original data the production slowly starts at 08:59 a.m. (after 539 minutes), we set the values between 08:59 a.m. and 09:15 a.m. (555 minutes) to a high value (1000W). In the consecutive 15 minutes, we keep the original production data and then add another high production range between 09:31 a.m. (571 minutes) and 09:45 a.m. (585 minutes), as shown in Figure 7. In the following, we show how the proposed IDSes are handling such attacks in the scenarios shown in Figure 6.

*The prototype IDS*: To show the functionality of the approach, we use the original data without measurement errors for training the prediction of the production. Within the same data set, values are falsified, as described above. Due to the relatively large changes, we expect the prototype to issue warnings only for the falsified values inserted by us. Note that the $\epsilon$ for the validity range has been set to of 150W. Results are shown in Figure 7, in which warnings are marked as red crosses. As expected, only the inserted values are marked as malicious. Clearly, such an accurate detection is only possible for perfect predictions. However, such high-quality predictions are not available, so the expected production needs to be learned from previous values.

*The self-learning IDS*: As mentioned in Section IV-A, the predicted mean and the variance of a Normal distribution are learned from the data of the previous hour. As shown in Figure 8, a variety of original values is now also classified as malicious. For example, the values in between our inserted data caused warnings (between 555 and 571 minutes) are marked because these realizations differ substantially from the predicted Normal distribution. However, even after our attack warnings are issued for many original measurements, for example for all high values after 700 minutes. This is due to a sudden large variance in measurements that occurs around 700 minutes. The higher values then lie outside the validity range, whereas the lower values are close to the previous ones, which leads to a great number of *false positives*.

The above shows that both the prototype and the self-learning IDSes are not ideal. However, combining weather information and data from previous hours and possibly days could improve the IDS's accuracy.

**Semantic attacks** - Nevertheless, it is possible to attack the power grid without a warning by the IDS. If an attacker knows how the IDS operates, it is possible to manipulate the data such that the values still lie within the validity range, but that the variance constantly rises. If this is done over several hours, the IDS accepts large jumps that can heavily burden the power grid. Such a semantic attack, which is accepted by our proposed self-learning IDS, is presented in Figure 9. Slightly after the beginning of the production phase, measurements are

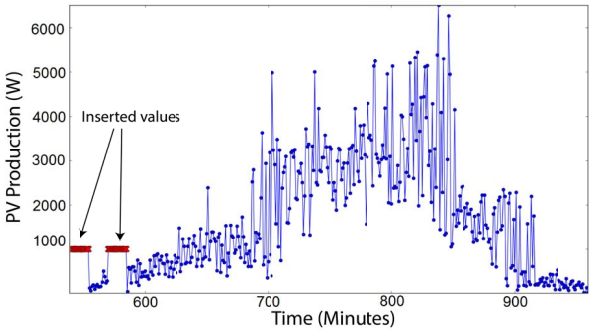



Fig. 7: Red crosses indicate warnings of the prototype IDS.

[2]https://bitbucket.org/mosaik/mosaik-demo

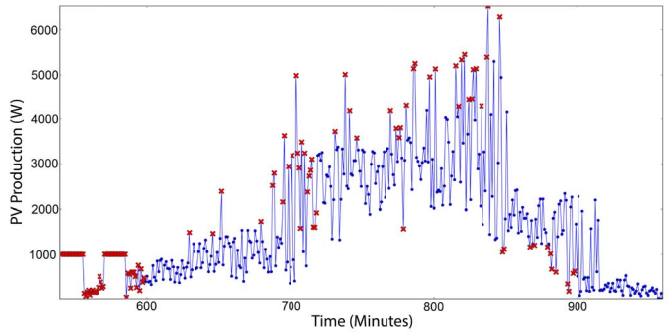



Fig. 8: Red crosses indicate warnings of the self-learning IDS.

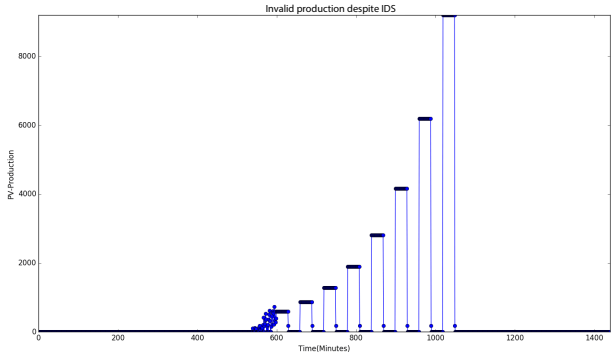



Fig. 9: Manipulated values.

altered to both the highest and the lowest production value that are still within the range of accepted values. Regularly repeating this influences the computed estimator for the next hour by allowing a higher variance. So, after 1000 minutes values greater than 9000W are still considered as valid by our IDS. This kind of attack could be prevented by comparing measurements to current weather prediction or even to measured temperatures at the smart house, as suggested in [8].

## V. Conclusions

Smart grids require secure and dependable control networks, that are able to detect cyber attacks and intrusions also locally in the field stations. Hence, intrusion detection mechanisms should be incorporated in the field stations and not only in the central control room. This paper illustrates a testbed, based on co-simulation with Mosaik, that can be used to validate local intrusion detection mechanisms at field stations. We use it to evaluate the impact of several local IDS mechanisms on different kinds of attacks using two scenarios: (i) an RTU of buses connecting power lines and (ii) a smart house.

Future work will improve the IDS for the smart home by incorporating weather predictions into the self-learning IDS. Furthermore, more complex semantic attacks will be investigated next to a broader study with more simulation runs to quantify detection rates on the distributed approach. The ultimate goal of this line of research is the development of a dedicated local monitoring tool for field stations in power distribution that is able to deal with different protocols.

## Acknowledgment

This research is funded through the NWO project (”MOre secure scada through SElf-awarenesS”) grant nr. 628.001.012.

## References

[1] B. Zhu, A. Joseph, and S. Sastry, “A taxonomy of cyber attacks on SCADA systems,” in *Int. Conf. on Internet of things and on cyber, physical and social computing*, pp. 380–388, IEEE, 2011.

[2] ICS-CERT, “Alert (IR-ALERT-H-16-056-01) Cyber-Attack Against Ukrainian Critical Infrastructure.” https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01 , released February 25, 2016.

[3] M. A. H. Sadi, M. H. Ali, D. Dasgupta, R. K. Abercrombie, and S. Kher, “Co-simulation platform for characterizing cyber attacks in cyber physical systems,” in *Symposium Series on Computational Intelligence*, pp. 1244–1251, IEEE, 2015.

[4] H. Lin, A. Slagell, C. D. Martino, Z. Kalbarczyk, and R. K. Iyer, “Adapting Bro into SCADA : Building a Specification-based Intrusion Detection System for the DNP3 Protocol,” pp. 2–5, 2012.

[5] H. Bao, R. Lu, and R. Deng, “BLITHE: Behavior rule-based insider threat detection for smart grid,” in *Vol. 3, No. 2*, IEEE internet of things journal, IEEE, April 2016.

[6] J. J. Chromik, A. Remke, and B. R. Haverkort, “What’s under the hood? Improving SCADA security with process awareness,” in *Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids*, pp. 1–6, IEEE, 2016.

[7] G. Hoogsteen, A. Molderink, J. L. Hurink, G. J. Smit, F. Schuring, and B. K. Liandon, “Impact of peak electricity demand in distribution grids: a stress test,” in *PowerTech, 2015 IEEE Eindhoven*, pp. 1–6, IEEE, 2015.

[8] A. M. Kosek, “Contextual anomaly detection for cyber-physical security in smart grids based on an artificial neural network model,” in *Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids*, pp. 1–6, IEEE, 2016.

[9] Y. Isozaki, S. Yoshizawa, Y. Fujimoto, H. Ishii, I. Ono, T. Onoda, and Y. Hayashi, “On detection of cyber attacks against voltage control in distribution power grids,” in *IEEE International Conference on Smart Grid Communications*, pp. 842–847, IEEE, 2014.

[10] J. J. Chromik, A. Remke, and B. Haverkort, “Improving SCADA security of a local process with a power grid model,” in *4th Int. Symp. for ICS&SCADA Cyber Security Research*, BCS Learning & Development Ltd., 2016.

[11] J. J. Chromik, A. Remke, and B. R. Haverkort, “A Testbed for locally Monitoring SCADA Networks in Smart Grids,” in *International workshop Energy-Open*, IEEE, 2017. To appear.

[12] “The Modbus Organization, Modbus application protocol specification, ver. 1.1b3,” 2012.

[13] “OASIS Standard, MQTT Version 3.1.1,” 2014. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf.

[14] A. Wain, S. Reiff-Marganiec, H. Janicke, and K. Jones, “Towards a distributed runtime monitor for ICS/SCADA systems,” 2016.

[15] M. Lévesque, D. Q. Xu, G. Joós, and M. Maier, “Communications and power distribution network co-simulation for multidisciplinary smart grid experimentations,” in *45th Annual Simulation Symposium*, p. 2, Society for Computer Simulation International, 2012.

[16] P. Gunathilaka, D. Mashima, and B. Chen, “Softgrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions,” in *2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, pp. 113–124, ACM, 2016.

[17] OFFIS, “Mosaik Documentation,” April 2017. http://mosaik.readthedocs.io/en/latest/overview.html.

[18] F. Schloegl, S. Rohjans, S. Lehnhoff, J. Velasquez, C. Steinbrink, and P. Palensky, “Towards a classification scheme for co-simulation approaches in energy systems,” in *Smart Electric Distribution Systems and Technologies (EDST), 2015 International Symposium on*, pp. 516–521, IEEE, 2015.

[19] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.