Diagnosing False Data Injection Attacks in the Smart Grid: a Practical Framework for Home-area Networks

Adrian Padin[1,3], Yeabsera Kebede[1,3], Maxwell Morgan[1], Davis Vorva[1],
Atman Fozdar[2], Richard Kalvaitis[2], Nikolas Remley[2],
Samir Tout[2], and Michael G. Kallitsis[3]

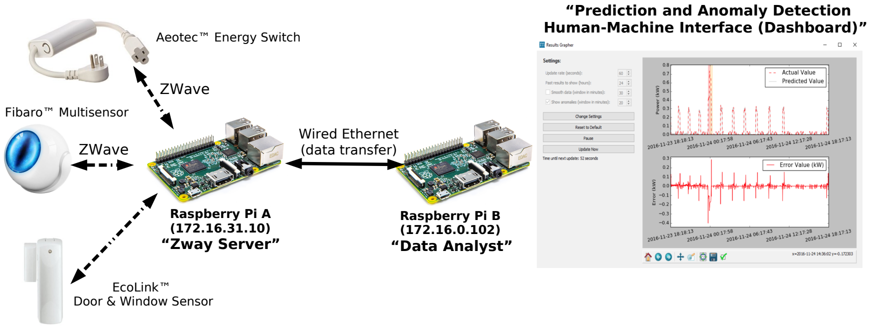
[1] Computer Science and Engineering, University of Michigan, Ann Arbor, USA,
[2] Information Assurance, Eastern Michigan University, Ypsilanti, USA,
[3] Merit Network, Inc., University of Michigan, Ann Arbor, USA
`apadin,yabskbd,mjmor,dvorva@umich.edu`,
`afozdar,rkalvait,nremley,stout@emich.edu`,
`mgkallit@merit.edu`

Abstract. Advances in the metering infrastructure of the electric grid allow two-way communication capabilities between the utility center and a vast array of smart meters installed in the grid's distribution and transmission components. Nefarious users that manage to compromise insecure smart meters can alter the payload transmitted from these meters, and abruptly increase or reduce electricity demand in a coordinated manner. This malicious practice, known as *false data injection attack*, can destabilize the grid. This paper describes a practical framework for diagnosing false data injection attacks in the smart grid. We propose a behavioral-based monitoring system that can be installed at home-area networks for detecting the aforementioned anomalies. We demonstrate a real-world prototype of our system engineered with inexpensive devices such as Raspberry Pi's and Z-Wave wireless sensors, and evaluate its performance with real data.

Key words: Smart grid, anomaly detection, false data injection attacks, statistics, algorithms, software, monitoring, real-world measurements.

1 Introduction

The modernized electric grid, or smart grid, is a "system of systems" that integrates two-way communication capabilities in order to enable the grid's efficient, reliable, secure, and resilient operation [1, 2]. The power grid leverages functionality introduced by *Advanced Metering Infrastructure* (AMI) nodes that are installed to provide real-time pricing estimates, accurate information for power demand, and network diagnostics (such as voltage frequencies) to the utility. Demand response schemes, the introduction of renewable energy sources (e.g. solar and wind), and the deployment of micro-grids underline the requirement for anomaly-free and robust operation of the smart metering infrastructure.



Settings:
Update rate (seconds): 60
Past results to show (hours): 24
□ Smooth data (window in minutes): 30
☑ Show anomalies (window in minutes): 20
Time until next update: 52 seconds
x=2016-11-24 14:36:02 y=-0.172303

Fig. 1. System Architecture.

At the same time, these data communication capabilities end the grid's isolation with "external" communication networks such as the Internet, and instill an array of new security threats. A plethora of vulnerable industrial control or smart grid devices can easily be enlisted with simple scanning tools [3, 4]. Cases where adversaries were capable of inflicting *physical* damage onto critical smart grid infrastructure have already been documented; the list includes the Stuxnet worm and the attacks against Iranian nuclear facilities [5], the compromise of a steel mill in Germany [6], and the cyber attacks on the Ukrainian power grid [7]. Nefarious users that manage to infect AMI meters in an orchestrated manner (e.g., via self-propagating malware) can manipulate their energy readings, and abruptly increase or reduce the energy demand reported to the utility. These *false data injection attacks* can compromise demand response schemes and endanger the grid's stability and state estimation process.

Our proposed solution is a a monitoring system for the detection of false data injection attacks in residential smart meters. The engineered solution is based on *inexpensive hardware* that operate in a federated manner in home-area networks, and apply statistical-based, correlative monitoring techniques to detect the onset of abnormal AMI activities. The proposed system consists of off-the-shelf "Internet-of-Things" (IoT) devices, such as Raspberry Pi's and Z-Wave wireless sensors, which collect measurements from a home-area network (e.g., motion, temperature, circuit power load—see Figure 1). Our main contributions are twofold: 1) we illustrate the design and implementation of a situational-awareness system in home-area networks using *broadly accessible*, inexpensive IoT devices—our original prototype was engineered with devices that cost roughly 400 USD; and 2) we leverage the sensor measurements obtained from our situational-awareness system to *transition to practice* an anomaly detection methodology for diagnosing bad data injection attacks in smart grids. The monitoring algorithm, recently proposed by our team in [8], utilizes the sensor readings to learn, in a correlative manner, their association with the home's power consumption. This is achieved by fitting a linear regression model that is hence employed to forecast the energy usage using the independent observations (i.e., the wireless sensor readings). Large deviations between the actual meter value and the predicted ones are flagged as anomalies by our sequential hypothesis testing module (see [8] for more details). Our prototype is currently being utilized in a real-world setting at a model house (see [9]) at NextEnergy, Inc.[1] Code can be downloaded from [10].

This paper is organized as follows: Section 2.1 presents the measurement framework that is based on Z-Wave sensors; in Section 2.2 we give a brief overview of our statistical-based diagnosis system, and in Section 3 we demonstrate our platform using real-world data collected at the NextEnergy facility.

2 System Description

2.1 Real-time Monitoring and Measurements

The architecture of our framework is illustrated in Figure 1, and resembles our deployment at NextEnergy, Inc. The model house at NextEnergy [9] is relatively small (about 400 sq. ft.) with two main rooms; a living room and a bedroom. It also includes a small bathroom and a small kitchen, and is equipped with several home appliances such as smart TV, stove, microwave, laundry machine and dryer, etc. The monitoring setting includes two Raspberry Pi's, several Z-Wave wireless sensors, and a USB microphone utilized as a "sound sensor". In particular, we used three Fibaro-branded [11] Z-Wave multi-sensors (i.e., motion, temperature, and luminosity), one Ecolink door sensor [12], one Aeotec energy switch [13] and an Everspring water/flood sensor [14].

All Z-Wave sensors are paired with the "ZWay server" Pi that is equipped with a RaZberry daughter board (mounted on GPIO pins and communicating with the sensors at 908.42 MHz) and runs a Z-Way server. The Z-Way server is a control program used to manage and monitor Z-Wave home automation networks and associated IoT devices. The Z-Wave network can be managed via a web interface hosted on this server to change various settings and collect data from the sensors. The Z-way server was configured with a static private network IP, with the web interface located at port 8083. The above-mentioned sensors were included into the network by using the "sensor inclusion" procedure [15].

A second Raspberry Pi node is utilized to *poll* data from the "Z-Way server" Pi and to run the anomaly detection algorithm in real-time. Although our system can be deployed on a single Pi, we elected the use of a pair in order to avoid computational bottlenecks that could arise on a single-node design. The two nodes communicate over an Ethernet point-to-point network. The "Data analyst Pi" retrieves sensor data from the "Z-Way server" every 15 seconds by issuing one HTTP GET request per sensor data-point. A typical URL for retrieving data via the Z-Wave JSON API is: `http://172.16.31.10:8083/ZWaveAPI/Run/devices[2].instances[0].commandClasses[49].data[1].val.value`. In this command `devices[2]` refers to the Z-Wave device with unique ID 2, `instances[0]` to the first instance of the device's function (i.e. first socket on a smart power strip), `commandClasses[49]` to a group of functions and variables capable of retrieving raw sensor data and `data[1]` to the type of data being retrieved [16].

[1] NextEnergy, a Detroit-based organization, provides experimentation facilities and laboratories for developing and testing advanced energy-related technologies.

Along with the data retrieved from the Z-Wave sensors we implemented a separate sound sensor using an off-the-shelf USB microphone. The microphone is installed on the "Data Analyst Pi" which runs an audio analyzer tool, Sound eXchange (SoX) [17]. Our code calls the `arecord` and `sox` command-line tools every 15 seconds in order to obtain the maximum sound amplitude recorded by the microphone during the sample period. For further details of the installation and tools used to implement the sound sensor refer to [10].

Energy Harvesting Sensors. We also examined the option of using *energy harvesting sensors* for our home-area measurements. These are inexpensive sensors that get powered by ambient energy sources such as solar, and thus offer the advantage of being battery-free. In particular, we tested EnOcean's door/window magnetic contact sensor (STM 320U) [18], wireless switch (PTM 210U) [19], and wireless temperature sensor (STM 332U) [20]. All the EnOcean sensors are paired with "EnOcean Pi" [21], that can be mounted on GPIO pins of a Raspberry Pi, and operates on 902 MHz acting as a bridge controller for the EnOcean sensors. "EnOcean Pi" runs an "FHEM" server (analogous to the Z-Way server, albeit with less functionality) which is a control program used to configure, monitor and control a variety of IoT devices including EnOcean and Z-wave sensors. The "EnOcean network" is very similar to Z-Wave network in terms of sensor management and data collection. All the sensors can be included in the network by pressing the "LRN" button on sensors after turning on inclusion mode on web interface [22], and data can be polled by launching an HTTP request similar to `http://IP_OF_RPI:8083/fhem?room=EnOcean`. Despite their battery-free capabilities we decided not to include them in our NextEnergy deployment. In a residential setting, these sensors cannot always receive the energy necessary for their seamless operation, and this resulted to unreliable measurements when we were evaluating them. Thus, we opted for Z-Wave sensors only.

2.2 Anomaly Detection Module

The sensor data retrieved from the "Data Analyst Pi", along with the total electricity consumption retrieved by our system directly from the home's smart AMI meter, are employed for detecting false data injection attacks. We posit the following *linear regression* model:

$$t = w_1 x_1 + \ldots + w_M x_M + \epsilon,$$

where t is the target/response variable (the total power consumption in Watts), x_i, $i \in \{1, 2, \ldots, M\}$ are the independent variables (regressors) and ϵ is a noise term that is normally distributed with zero mean and variance $1/\beta$. The independent variables are the sensor observations that provide ambient information about the home-area network. The regression coefficients w_i, $i \in \{1, \ldots, M\}$ and the other model parameters are obtained through the training phase, outlined below. For completeness, we next present an overview of our statistical anomaly detection module. Further details can be found in our previous work [8].

We denote the sensor observations by the feature vector $\mathbf{x} = (x_1, \ldots, x_M)^\top$. To train our system and learn the model parameters, we obtain training data for a period of size N; $\mathbf{t} := (t_1, \ldots, t_N)^\top$ represents the target values in the training set, and $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are the corresponding covariate values. We construct the $N \times M$ *measurement matrix* $\mathbf{X}$ by stacking the input variables of each data point. The imposed model implies that given the value of $\mathbf{x}$, the corresponding value of t has a Gaussian distribution with mean equal to $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$ and variance β^{-1}. Thus,

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}). \tag{1}$$

Assuming the data is drawn independently from (1), the *likelihood* is $p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(\mathbf{x}_n, \mathbf{w}), \beta^{-1})$. We follow a Bayesian approach; this allows us to perform a training phase and select the "best" model (i.e., one that avoids overfitting and has low variance) without the explicit need for cross-validation runs. A prior of the model parameters $\mathbf{w}$ is introduced, and we consider a *conjugate prior* that is a zero-mean, isotropic Gaussian governed by a single parameter α, i.e., $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$, where $\mathbf{I}$ is the identity matrix of appropriate dimension. The *posterior distribution* takes the form of another Gaussian

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, S_N), \tag{2}$$

with $\mathbf{m}_N = \beta S_N \mathbf{X}^\top \mathbf{t}$ and $S_N^{-1} = \alpha \mathbf{I} + \beta \mathbf{X}^\top \mathbf{X}$. The optimal parameter vector $\mathbf{w}^*$ in $y(\mathbf{x}, \mathbf{w})$ is obtained by maximizing the posterior distribution, and equals $\mathbf{w}^* = \mathbf{m}_N$. The model hyper-parameters α and β are learned through the iterative process we describe in [8]. Implementation details and our code are in [10].

After the completion of the algorithm's training period, each set of observations $\mathbf{x}_n$ acquired every 15 seconds is utilized by the "Data Analyst Pi" to obtain an estimate/prediction, $\hat{t}_n := y(\mathbf{x}_n, \mathbf{w}^*)$, of the house power consumption for that time point. The "Data Analyst Pi" compares this prediction with the actual power consumption reported by the smart meter at the same time slot, and *significant differences* between the two are flagged as anomalies by the *sequential hypothesis testing* methodology we proposed in [8].

Our hypothesis testing module utilizes the *predictive distribution* of the model. This takes the form

$$p(t_n|\mathbf{x}_n, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t_n|\mathbf{m}_N^\top \mathbf{x}_n, \sigma_N^2(\mathbf{x}_n)), \tag{3}$$

where the variance of the predictive distribution is given by $\sigma_N^2(\mathbf{x}_n) = \beta^{-1} + \mathbf{x}_n^\top S_N \mathbf{x}_n$. The first term represents the noise in the data, and the second term reflects the uncertainty of predictions associated with the parameter vector $\mathbf{w}^*$.

The predictive distribution plays the role of *Null Hypothesis* or *reference* distribution, denoted as F_n, for the differences (referred as *errors* henceforth) between the actual and the predicted power consumption. Following [23], for each new observation $(t_n, \mathbf{x}_n)$ we calculate the error $e_n := t_n - \mathbf{m}_N^\top \mathbf{x}_n$, and then find the p-value corresponding to that error using the reference distribution F_n. We are interested in employing a hypothesis testing criterion for detecting

Table 1. Predictive power and analysis of variance for the independent variables (regressors/covariates) obtained using our infrastructure during the seven-day measurement study starting on Wednesday, 22 Feb. 2017 00:00 EST.

Regressor Variable	**Corr. Coeff. with power**	**Var. Explained** R^2 **(%)**
Motion sensor 1 (living room)	0.20	4.11
Motion sensor 2 (bedroom)	0.11	1.13
Motion sensor 3 (outside)	0.06	0.35
Luminosity 1 (living room)	0.24	5.95
Luminosity 2 (bedroom)	0.18	3.38
Luminosity 3 (outside)	0.13	1.65
Temperature 1 (living room)	0.06	0.32
Temperature 2 (bedroom)	0.10	0.92
Temperature 3 (outside)	0.05	0.23
Sound sensor (living room)	0.20	3.85

sequences of “abnormally” small p-values. We monitor for anomalies by utilizing an Exponentially Weighted Moving Average (EWMA) control scheme [24, 23], known as *Q-charting* in quality control. EWMA allows us to tame the false alert rate and obtain higher *test power* (i.e., correctly rejecting the Null hypothesis when it is indeed false). We refer the reader to [8, 10] for further details.

3 Numerical Experiments and Evaluation

This section evaluates our system using real-world measurements collected at NextEnergy. To test our implementation of the proposed system, a measurement study was conducted during the week of February 22nd–28th, 2017. During this study, the model house was inhabited from approximately 9am–9pm (12 hours) by 1-2 members of our research team. The intent of the study was to recreate the realistic living conditions that could influence a home’s power usage. The participants took care to act as if they were inhabiting the home, going so far as to cook meals and use the appliances such as the microwave and laundry machines in order to simulate real-world living conditions.

An analysis of variance (ANOVA) was conducted on the collected data to determine which input features explain the most amount of variability in the data. In particular, by performing a regression analysis on each covariate, one can calculate the *coefficient of determination* R^2, defined as $R^2 = 1 - \frac{SS_E}{SS_T}$, where SS_T is the total sum of squares, and SS_E is the error sum of squares (see [25]). Table 1 describes the results of this analysis, and shows a clear correlation betwen several key covariates (such as motion and luminosity) with the house’s power consumption[2]. The most informative variables appear to be the motion, luminosity, and sound sensors in the living room. This is likely due to the fact that the living room was the room which was most often occupied by the researchers throughout the experiment. We also examined the inclusion of *temporal* information in our regression model; the high correlation coefficient

[2] Some covariates were not included in this analysis due to their invariant zero values.

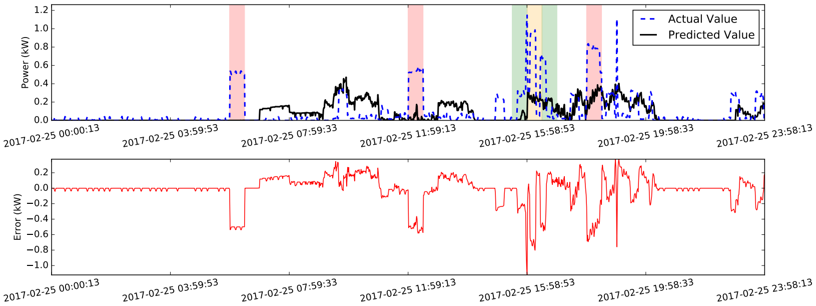


Fig. 2. NextEnergy data: prediction and detection performance. We observe the predicted values as reported by our system (solid black line) and the actual power consumption (dotted blue line) for a single day. Lower panel shows differences between predicted and actual electricity consumption.

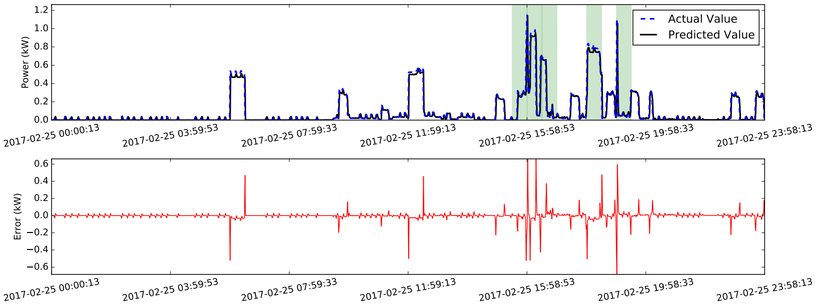


Fig. 3. NextEnergy data: performance of the auto-regressive model.

(0.99) between the power consumption at time n and $n-1$ suggested that an *auto-regressive* model could improve prediction performance.

Figures 2 and 3 illustrate the performance of our algorithm. The colored bars indicate the number of alerts thrown by the system; red regions indicate between 41 and 60 alerts were raised in a one-hour period, yellow regions indicate 21–40 alerts were raised, and green regions signify 1–20 alerts. We study two models: one where only the features of Table 1 are used (Figure 2), and one where a temporal feature was also added (Figure 3). The prediction and detection performance for February 25th, 2017 are shown for demonstration. The system is initially trained for 24 hours before the first forecasting/detection period can begin; it is then re-trained at the mark of every hour on the previous day's data. To evaluate the detection system in the presence of "bad data" attacks, three attacks were manually injected at 6am, 12pm, and 6pm (red shaded regions in Figure 2). All attacks had a duration of 30 minutes and magnitude 500 Watts. As expected, the error charts of Figures 2 and 3 show that the auto-regressive model demonstrates better prediction accuracy. However, the first model demonstrates a better anomaly-detection accuracy; it is able to track all three injected attacks with only a few false positives (yellow and green region near 4pm, Figure 2). Many more false positives and false negatives are observed in Figure 3 (green regions), likely because the auto-regressive model adjusts too well to the attacks and can only detect the beginning and end of each anomaly.

4 Related Work

Our proposed methodology falls under the category of *anomaly-based detection*, and complements signature-based and specification-based detection methods [26, 27, 28]. Signature-based methods are suitable for identifying malware that have already appeared in a smart grid environment and whose activities have been documented in malware databases. Such systems examine packets as they arrive to the utility's control center and look for *known* binary patterns (e.g., *Snort* [29] is a representative example of such systems). Specification-based detection is accomplished by measuring deviations from a normal operational profile that is predefined. Examples include finite state machine monitors, data validation with range checks, authentication monitor and physical health inquiries for catching unresponsive nodes, and verification of system state [30, 31].

One shortcoming of relying on prior knowledge recorded in black-lists is that new malware activities will not be uncovered. Similarly, specification-based methods can be cumbersome to fine-tune; finding a valid range for the AMI power demand and supply is not easily determined. Further, subtle attacks might exist that involve modifying control parameters in a way that appears to be within a normal range, but still being capable of inflicting system damage. Instead, an anomaly-based method identifies attacks by checking for significant deviations from normal traffic patterns; it monitors the signal of interest to find its normal behavior, and detects outliers when a statistic exceeds a predefined threshold.

Existing anomaly-based defenses against adversaries that inject spurious data measurements into the power grid follow a "network-view" perspective. Such countermeasures for detecting false data injection appear in [32, 33, 34, 35]. [32] proposes an adaptive cumulative sum test combined with a multivariate hypothesis testing problem to prevent an erroneous grid-state estimate. [33] studies a graph theoretic method for securing an optimal set of meter measurements so that state estimation is not compromised. [34] couples anomaly-based methods with a data integrity check to combat stealth attacks, while [35] looks for inconsistent grid behavior using clustering techniques. [36] sheds light into situations of *multiple adversaries* performing injection attacks, and discusses optimal defense strategies from game theory. Instead, we tackle the problem from a different vantage point. The "home-area view" we suggest aims to detect arbitrary data injection attempts at their origin, i.e., compromised residential smart meters. Our framework complements the above-mentioned work since the alert output signal generated by our method could serve as an additional input that can be communicated to the utility (via a secure, alternate channel).

5 Conclusions

We have presented a practical framework for the detection of false data injection attacks in smart electric networks. The proposed system employs correlative monitoring to detect the onset of "spoofed-data" incidents in smart meters. The collected data are analyzed in a Bayesian linear regression model that helps us learn the normal operating regime of a meter's power consumption. Significant and persistent deviations from this regime are treated as outliers and reported by our system as anomalies. Our system is deployed with broadly accessible, inexpensive devices such as Raspberry Pi's and wireless sensors for home-area networks. We tested our proposed system in a real-world environment at a model house in Detroit, Michigan, and have presented results which show that the system can accurately detect injected attacks.

Future work includes the examination of new prediction models, especially models that can better capture the non-linear relationship between the response variable (power consumption) and the independent input features measured within a home-area network. Special attention should be paid towards the design of a system that can be trained efficiently on off-the-shelf computer nodes such as the Raspberry Pi so that the system remains low-cost and accessible.

References

1. Hamid Gharavi and Reza Ghafurian, "Smart Grid: The Electric Energy System of the Future," *Proceedings of the IEEE*, June 2011.
2. Massachusetts Institute of Technology, *The Future of the Electric Grid: An Interdisciplinary MIT study*, MIT Energy Initiative, 2001.
3. Zakir Durumeric, Eric Wustrow, and J. Alex Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *USENIX SEC'13*, 2013, pp. 605–620.
4. Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman, "A search engine backed by internet-wide scanning," in *Proceedings of the ACM CCS '15*, 2015, pp. 542–553.
5. N. Falliere, L. Murch, and E. Chien, "W32.stuxnet dossier," 2011.
6. R. Lee, M. Assante, and T. Conway, "German steel mill cyber attack," 2014.
7. R. Lee, M. Assante, and T. Conway, "Analysis of the cyber attack on the ukrainian power grid," 2016.
8. M. G. Kallitsis, G. Michailidis, and S. Tout, "Correlative monitoring for detection of false data injection attacks in smart grids," in *IEEE SmartGridComm*, 2015.
9. Next Energy, "NextHome," `http://www.bit.ly/12OOr1j`.
10. Adrian Padin, Davis Vorva, Maxwell Morgan, Yeabsera Kebede, and Michael Kallitsis, "A practical framework for detecting false data injection attacks in home-area networks," `https://github.com/Merit-Research/Smart-Grid-Analytics`.
11. Fibaro, "Fibaro multisensor," `http://www.fibaro.com/us/the-fibaro-system/motion-sensor`.
12. Ecolink, "Z-wave door/window sensor," `http://www.discoverecolink.com/product/dwzwave2-eco/`.
13. Aeon Labs, "Smart energy switch," `http://aeotec.com/z-wave-plug-in-switch/942-smart-energy-switch-manual-instructions.html`.
14. Everspring, "Everspring z-wave water/flood sensor," `http://www.everspring.com/ST812.aspx`.

15. Z-Wave.Me Team, "Z-way users documentation," http://razberry.z-wave.me/docs/zwayUse.pdf.
16. Z-Wave.Me Team, "Z-way developers documentation," http://razberry.z-wave.me/docs/zwayDev.pdf.
17. Chris Bagwell, Rob Sykes, and Pascal Giard, "Sound Exchange (SoX)," http://sox.sourceforge.net/.
18. EnOcean Team, "Stm 320u (door/window contact sensor)," https://www.enocean.com/en/enocean_modules_902mhz/stm-320u/.
19. EnOcean Team, "Ptm 210u (wireless switch)," https://www.enocean.com/en/enocean_modules_902mhz/ptm-210u/.
20. EnOcean Team, "Stm 332u (wireless temperature sensor)," https://www.enocean.com/en/enocean_modules_902mhz/stm-332u/.
21. Element 14, "Enocean pi 902," https://www.element14.com/community/docs/DOC-55169?ICID=enoceanpi-space-learn.
22. EnOcean Team, "Raspberry pi talks enocean," https://www.enocean.com/fileadmin/redaktion/pdf/white_paper/wp_Raspberry_talks_EnOcean.pdf.
23. Diane Lambert and Chuanhai Liu, "Adaptive thresholds: Monitoring streams of network counts," *online, J. Am. Stat. Assoc*, pp. 78–89, 2006.
24. James M. Lucas and Michael S. Saccucci, "Exponentially weighted moving average control schemes: Properties and enhancements," *Technometrics*, vol. 32, no. 1, pp. 1–29, Jan. 1990.
25. Douglas C. Montgomery, *Design and Analysis of Experiments, 8th Ed.*, Wiley, 2013.
26. R. Berthier, W.H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in *IEEE SmartGridComm*, 2010, pp. 350–355.
27. F.M. Cleveland, "Cyber security issues for advanced metering infrastructure," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 2008, pp. 1–5.
28. Wenye Wang and Zhuo Lu, "Survey cyber security in the smart grid: Survey and challenges," *Comput. Netw.*, Apr. 2013.
29. Sourcefire, "Snort," http://www.snort.org/snort.
30. Hoda M. Hassan, Mohy Mahmoud, and Sherif El-Kassas, "Securing the aodv protocol using specification-based intrusion detection," in *Proceedings of Q2SWinet '06*, New York, NY, USA, 2006, pp. 33–36, ACM.
31. Andrea Carcano, Igor Nai Fovino, Marcelo Masera, and Alberto Trombetta, "State-based network intrusion detection systems for SCADA protocols: A proof of concept," in *Proceedings of CRITIS'09*, 2010, pp. 138–150.
32. Yi Huang et al., "Bad data injection in smart grid: attack and defense mechanisms," *Communications Magazine, IEEE*, Jan. 2013.
33. Suzhi Bi and Ying Jun Zhang, "Graphical methods for defense against false-data injection attacks on power system state estimation," *Smart Grid, IEEE Transactions on*, vol. 5, no. 3, pp. 1216–1227, May 2014.
34. Wei et al. Yu, "An integrated detection system against false data injection attacks in the smart grid," *Security and Communication Networks*, vol. 8, no. 2, 2015.
35. Abdulmohsen Almalawi, Xinghuo Yu, Zahir Tari, Adil Fahad, and Ibrahim Khalil, "An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems," *Computers and Security*, vol. 46, no. 0, pp. 94 – 110, 2014.
36. A. Sanjab and W. Saad, "Smart grid data injection attacks: To defend or not?," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2015, pp. 380–385.