Evaluating Electricity Theft Detectors in Smart Grid Networks

Daisuke Mashima[1] and Alvaro A. Cárdenas[2]

[1] Georgia Institute of Technology
mashima@cc.gatech.edu
[2] Fujitsu Laboratories of America
alvaro.cardenas-mora@us.fujitsu.com

Abstract. Electricity theft is estimated to cost billions of dollars per year in many countries. To reduce electricity theft, electric utilities are leveraging data collected by the new Advanced Metering Infrastructure (AMI) and using data analytics to identify abnormal consumption trends and possible fraud. In this paper, we propose the first threat model for the use of data analytics in detecting electricity theft, and a new metric that leverages this threat model in order to evaluate and compare anomaly detectors. We use real data from an AMI system to validate our approach.

1 Introduction

The smart grid refers to the modernization of the power grid infrastructure with new technologies, enabling a more intelligently networked automated system with the goal of improving efficiency, reliability, and security, while providing more transparency and choices to electricity consumers. One of the key technologies being deployed currently around the globe is the Advanced Metering Infrastructure (AMI).

AMI refers to the modernization of the electricity metering system by replacing old mechanical meters by *smart meters*. Smart meters are new embedded devices that provide two-way communications between utilities and consumers, thus eliminating the need to send personnel to read the meters on site, and providing a range of new capabilities, such as, the ability to monitor electricity consumption throughout the network with finer granularity, faster diagnosis of outage—with analog meters, utilities learned of outages primarily by consumer call complaints—automated power restoration, remote disconnect, and the ability to send information such as dynamic pricing or the source of electricity (renewable or not) to consumers, giving consumers more—and easier to access—information about their energy use.

Smart meters are, by necessity, billions of low-cost commodity devices, with an operational lifetime of several decades and operating in physically insecure locations [16]. Hardening these devices by adding hardware co-processors and tamper resilient memory might increase the price of smart meters by a few dollars, and because utilities have to deploy millions of devices, the reality of the

D. Balzarotti, S. Stolfo, and M. Cova (Eds.): RAID 2012, LNCS 7462, pp. 210–229, 2012.

market is that these additions are not considered cost-effective in practice, and are not even recommended as a priority [21].

Therefore, while some basic protective measures have been developed (tamper-evident seals, secure link communications), they are not enough to prevent successful attacks during the meter lifespan. In addition to vulnerabilities identified by security researchers [17,9]—some of them allowing rogue remote firmware updates [20]—hacked smart meters have been used to steal electricity, costing a single U.S. electric utility hundreds of millions of dollars annually, as reported by a cyber-intelligence bulletin issued by the FBI [14]. The FBI report warns that insiders and individuals with only a moderate level of computer knowledge are likely able to compromise and reprogram meters with low-cost tools and software readily available on the Internet. The FBI report also assesses with medium confidence that as smart grid use continues to spread throughout the country, this type of fraud will also spread because of the ease of intrusion and the economic benefit to both the hacker and the electric customer.

Detecting electricity theft has traditionally been addressed by physical checks of tamper-evident seals by field personnel and by using balance meters [10]. While valuable, these techniques alone are not enough. Tamper evident seals can be easily defeated [5] and balance meters can detect that some of the customers connected to it are misbehaving, but cannot identify exactly who they are. Despite the vulnerabilities of smart meters, the high-resolution data they collect is seen as a promising technology to improve electricity-theft detection. In general, utilities are gathering more data from many devices and they are leveraging *big data analytics* [15] to obtain better situational awareness of the health of their system. One of the key services offered by Meter Data Management (MDM) vendors for turning big data into actionable information is called *revenue assurance*, where data analytics software is used by the utility on the collected meter data to identify possible electricity theft situations and abnormal consumption trends [13]. Big data analytics is thus a new cost-effective way to complement the use of balance meters (which are still necessary to detect when electricity thieves connect directly to the power distribution lines instead of tampering with the meter) and physical personnel checking for tamper-evident seals.

In this paper we focus on the problem of data analytics in MDM systems for detecting electricity theft. While some MDM vendors are already offering this functionality, their methods and algorithms are not publicly available, so it is impossible to evaluate the effectiveness of these tests. In addition, the few papers available on the topic have limitations [18,19,11,6]: (1) They do not consider a threat model, and therefore, it is not clear how the detection algorithm will work against sophisticated attackers, (2) they have lower resolution data, and therefore they tend to focus on nonparametric statistics, instead of leveraging advanced signal processing algorithms, and (3) they assume a dataset of attack examples to test the accuracy of the classifiers, and therefore the evaluation will be biased depending on how easy it is to detect attacks available in the database, and the effectiveness of the classifier will be unknown to unseen attacks.

In this paper we make the following contributions: (1) We introduce an attacker model for anomaly detectors in MDM systems. Previous work never assumed an intelligent attacker and therefore might have easily been evaded by an advanced attacker. This threat model is particularly important in digital meters, as an attacker with access to a tampered meter can send an arbitrary fine-grained attack signal with a precision that was not previously available with mechanical attacks to meters (such as using powerful magnets to affect the metrology unit). (2) We introduce a new metric for evaluating the classification accuracy of anomaly detectors. This new metric takes into consideration some of the fundamental problems in anomaly detection when applied to security problems: (a) the fact that attack examples in a dataset might not be representative of future attacks (and thus a classifier trained with such attack data might not be able to detect new *smart* attacks), and (b) in many cases it is hard to get *attack* data for academic studies—this is particularly true for SCADA data and data from sensor and actuators in industrial or power grid systems—therefore we argue that we have to avoid training and evaluating classifiers with imbalanced and unrepresentative datasets. (3) Using real AMI data (6 months of 15 minute reading-interval for 108 consumers) provided by an utility, we evaluate the performance of electricity-theft detection algorithms, including a novel ARMA-GLR detector designed with the goal of capturing an attack invariant (reducing electricity bill) in the formal model of composite hypothesis testing.

2 Evaluation of Classifiers in Adversarial Environments

In this section we describe a new general way of evaluating classifiers in adversarial environments. Because this framework can be used for other problems, we introduce the model in a general classification setting. We focus on two topics: (1) **adversarial classification**, or *how to evaluate the effectiveness of a classifier when the attacker can create undetected attacks*, and (2) **adversarial learning**, or *how to prevent an attacker from providing false data to our learning algorithm.*

2.1 Adversarial Classification

In machine learning, classifiers are traditionally evaluated based on a testing dataset containing examples of the negative (normal) class and the positive (attack) class. However, in adversarial environments there are many practical situations where we cannot obtain examples of the attack class a priori. There are two main reasons for this: (1) by definition, we cannot obtain examples of zero-day attacks, and (2) using attack examples which are generated independently of the classifier implicitly assumes that the attacker is not adaptive and will not try to evade our detection mechanism.

In this paper we argue that instead of using a set of attack samples for evaluating classifiers, we need to find the worst possible attack for each classifier and evaluate the classifier by considering the costs of this worst-case attack.

Model and Assumptions: We model the problem of evaluating classifiers by generating worst-case attack patterns as follows:

1. A random process generates observations $x \in \mathcal{X}$. These observations are the realization of a random vector X with distribution P_0.
2. We assume x is only observed by a *sensor* (e.g., a smart meter), and the sensor sends y to a classifier. Thus while P_0 is known to the world, the specific sample x is only known to the sensor.
3. The sensor can be in one of two states (1) honest, or (2) compromised. If the sensor is honest, then $y = x$. If the sensor is dishonest, then $y = h(x)$, where $h : \mathcal{X} \to \mathcal{X}$ is a function such that the inferred probability distribution P_1 for Y satisfies a *Relation* (the attacker intent): $g(X)\ R\ g(Y)$ (e.g., $\mathbb{E}[Y] < \mathbb{E}[X]$ where $\mathbb{E}[X]$ is the expectation of the random variable X).
4. The classifier $f : \mathcal{X} \to \{n, p\}$ outputs a decision: A negative n for concluding that y is a sample of P_0 and a positive p to decide that y is a sample of P_1.

A Metric for Evaluating Classifiers in Adversarial Environments: In order to generate attacks we propose a cost function $C(x_i, y_i)$ to generate attack vectors y_i by modifying the original value x_i such that y_i is the attack that maximizes $C(x_i, y_i)$ while being undetected. In particular, we assume we are given:

1. A set $\mathcal{N} = \{x_1, \ldots, x_m\} \in \mathcal{X}^m$ where each x_i is assumed to be a sample from P_0. Note that $x_i \in \mathcal{X}$. A common example is $\mathcal{X} = \mathbb{R}^d$, i.e., each observation x_i is a vector of real values with dimension d. In a smart-metering application this can mean that x_i corresponds to the meter readings collected over a 24-hour period.
2. A value $\alpha \in [0, 1]$ representing an upper bound on the tolerable false alarm probability estimate in the set $\mathcal{N}$.
3. A cost function $C : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ representing the cost of a false negative.
4. A set of candidate classifiers $\mathcal{F} = \{f_0, \ldots, f_q\}$, where each classifier is parameterized by a threshold τ used to make a decision. If we want to make explicit the threshold used by a particular classifier we use the notation f_{i,τ_i}.

Calculating the Adversarial Classification Metric

1. $\forall f_{i,\tau_i} \in \mathcal{F}$ find the threshold that configures the classifier to allow a false alarm rate as close as possible to the upper bound α. If no such threshold exists, then discard the classifier (since it will always raise more false alarms than desired).

$$\tau_i* = \arg\max_{\tau_i \in \mathbb{R}} \Pr[f_{i,\tau_i}(X) = p | X \sim P_0] \quad (1)$$

$$\text{subject to:} \quad \Pr[f_{i,\tau_i}(X) = p | X \sim P_0] \leq \alpha \quad (2)$$

This formal definition can be empirically estimated by the following equations:

$$\tau_i* = \arg\max_{\tau_i \in \mathbb{R}} \quad \frac{|\{x \in \mathcal{N} : f_{i,\tau_i}(x) = p\}|}{|\mathcal{N}|} \quad (3)$$

$$\text{subject to:} \quad \frac{|\{x \in \mathcal{N} : f_{i,\tau_i}(x) = p\}|}{|\mathcal{N}|} \leq \alpha \quad (4)$$

2. Among all classifiers that satisfy the false alarm constraint, find the worst possible *undetected* attacks for each of them. Let $y = h(x)$ denote the attack strategy based on observation x, then the optimal attack strategy requires an optimization over the functional space h:

$$\begin{aligned} C(f_i) = \quad & \max_h \mathbb{E}[C(X, h(X))] \\ \text{subject to: } & f_{i,\tau_i*}(h(x)) = n \text{ if } f_{i,\tau_i*}(x) = n \text{ and } h(x) = x \text{ if } f_{i,\tau_i*}(x) = p \end{aligned}$$

Notice that if the negative sample raises a false alarm then the attacker just forwards this value. While in practice an attacker might want to stay undetected at all times ($\forall x, f(h(x)) = n$), this would lower the attacker's gain (e.g., the amount of electricity the attacker can steal). We are therefore, considering the most conservative case where we allow the attacker to steal more without being detected (we count alarms generated by $h(x) = x$ as false alarms, and therefore allow the attacker to remain undetected with this aggressive attack). Given a dataset of negative examples, we can empirically estimate this functional optimization problem by the following equations:

$$\begin{aligned} & C(f_i) = \max_{[y_1,\ldots,y_m] \in \mathcal{X}^m} \sum_{x_i \in \mathcal{N}} C(y_i, x_i) \\ & \text{Subject to:} \quad f_{i,\tau_i*}(y_i) = n \text{ if } f_{i,\tau_i*}(x_i) = n \quad \text{and } y_i = x_i \text{ if } f_{i,\tau_i*}(y_i) = p \end{aligned}$$

3. The best classifier f_i* is the one with the minimum cost for the most costly undetected attack:

$$f_i* = \arg\min_{f_i \in \mathcal{F}} C(f_i) \tag{5}$$

2.2 Adversarial Learning

Another fundamental evaluation criteria should be the resilience and countermeasures deployed for adversarial learning. In general, the idea of learning some basic properties of a random process and then using them to detect anomalies sounds intuitive; however, in several cases of interest the random process may be non-stationary, and therefore we might need to retrain the classifier periodically to capture this **concept drift**.

Retraining a classifier opens the vulnerability that a smart attacker might force us to learn false *normal* models by poisoning the dataset. For our smart meter example, the attacker can send fake sensor measurement readings that lower average consumption but that do not raise alarms (when classified) so they can be used as part of the new training set. Over a period of time, our new estimated probability models will be different from the real process generating this data. We refer to these attacks as *contamination attacks* because they inject malicious data used to train the classifiers.

To evaluate the susceptibility of classifiers to contamination attacks, we study how these attacks can be generated and discuss a countermeasure in Section 4.2.

3 Electricity-Theft Detectors and Attacks

AMI systems collect and send electricity consumption data to the utility company several times per day. Electricity consumption data for a consumer is a time series $Y_1, Y_2, \ldots$, where Y_i is the electricity consumption of the utility customer in Watt-hours [Wh] from time period between measurement Y_{i-1} to measurement Y_i. The time between recorded measurements can change between different AMI deployments, as there is no standard defining the granularity of these measurements; however, a common measurement frequency is to take a recording every 15 minutes.

If an attacker obtains access to the meter, or is able to impersonate it, the attacker can send any arbitrary time series $\hat{Y}_1, \hat{Y}_2, \ldots$ back to the utility. Depending of the goal of the attacker, this false time-series can have different properties. In this paper we focus on attacks that electric utilities are most interested in detecting: electricity theft.

The goal of an attacker who steals electricity is to create a time series $\hat{Y}_1, \hat{Y}_2, \ldots$ that will lower its energy bill. Assuming the billing period consists of N measurements, the false time series should satisfy the following **attack invariant** for periods of time where electricity is billed at the same rate:

$$\sum_{i=1}^{N} \hat{Y}_i < \sum_{i=1}^{N} Y_i. \tag{6}$$

While one of the goals of the smart grid is to provide more flexible tariffs, these demand-response systems are still experimental and are currently deployed in trail phases. In addition, while the electric utility we are working with has a Time Of Use (TOU) program, all the traces we received were of their flat rate program (most of their customers do not take advantage of TOU). Therefore, while in future work we might need to consider other utility functions for the attacker (e.g., $\min_{Y_t} \sum Cost_t Y_t$) for the current work we focus on an attacker who only wants to minimize $\sum Y_i$. The main goal of this paper is to establish a sound evaluation methodology that can be extended for different cost-models.

In this section we propose several electricity-theft detectors to capture this attack invariant. While these detection algorithms have been studied extensively in the statistics and machine learning literature, this is the first work that studies how to apply them for electricity-theft detection.

To use the concept of *worst possible undetected attack* as outlined in Section 2, we define the following objective for the attacker: the attacker wants to send to the utility a time-series $\hat{Y}_i$ that will minimize its electricity bill: $\min_{\hat{Y}_i} \sum \hat{Y}_i$, subject to the constraint that a detector will not raise an alarm with $\hat{Y}_i$. We assume a very powerful attacker who has complete knowledge about each detection algorithm, the parameters that a detector uses, and has a complete historical data recorded on his own smart meter. This is indeed a very strong adversary model and might not represent the average risk of a utility; however, we want to build a lower-bound on the operational performance of the classifiers. The evaluation of classifiers using machine-learning and statistics in adversarial conditions

has been historically performed under fairly optimistic assumptions [22], therefore we would like to motivate future research for evaluating attack-detection algorithms in worst-possible scenarios so their performance is not overstated.

3.1 Average Detector

One of the most straightforward ways to construct an electricity-theft detector is to use an average of the historical electricity consumption under the same conditions. This is in fact the way utilities used to detect metering abnormalities pre-AMI systems [12]: given a gross measurement (e.g., average or total power consumption for a month) Y, determine if Y is significantly lower than the historical average.

With AMI systems utilities can now obtain a fine grained view of the consumption pattern, where $\sum_{i=1}^{N} Y_i = Y$ and N is the number of measurements to compute the average. To use this electricity-theft indicator in AMI systems we can calculate $\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i$ and then raise an alarm if $\bar{Y} < \tau$, where τ is a variable threshold.

Our *average detector* implementation calculates the detection threshold τ as follows. We consider a detector to handle an average of daily record.

1. Given a training dataset, say T days in the most recent past, we can compute T daily averages, D_i $(i = 1, \ldots, T)$.
2. $\tau = \min_i(D_i)$

Determining the threshold in this way, we do not encounter any false positives within the training dataset.

An attacker equipped with knowledge of τ and our implementation can mount an optimal attack by simply sending τ as $\hat{Y}_i$ all the day. Even though this attack results in an entirely "flat" electricity usage pattern, the average detector cannot detect this anomaly.

3.2 ARMA-GLR

One of the advantages of fine-grained electricity consumption patterns produced by the smart grid is that we can leverage sophisticated signal processing algorithms to capture more properties of normal behavior. We selected Auto-Regressive Moving Average (ARMA) models to represent a normal electricity consumption probability distribution p_0 because ARMA processes can approximate a wide range of time-series behavior using only a small number of parameters. ARMA is a parametric approach, and has the potential to perform better than nonparametric statistics if we can model p_0 and the optimal attack appropriately.

We train from our dataset an ARMA probability distribution p_0 (we used the *auto.arima* function in the *forecast* library in *R* [2] to fit ARMA models of our

data by using the Yule-Walker equations and the Akaike information criteria) defined by the following equation:

$$Y_k = \sum_{i=1}^{p} A_i Y_{k-i} + \sum_{j=0}^{q} B_j V_{k-j} \tag{7}$$

where V is white noise distributed as $\mathcal{N}(0, \sigma)$.

An attacker will choose a probability distribution that changes the mean value of the sequence of observations. Therefore the attack probability distribution (p_γ) is defined by

$$Y_k = \sum_{i=1}^{p} A_i Y_{k-i} + \sum_{j=0}^{q} B_j (V_{k-j} - \gamma) \tag{8}$$

where $\gamma > 0$ is an unknown value and quantifies how small will the attacker select $\mathbb{E}_\gamma[Y]$ (the expectation of Y under probability distribution p_γ).

Given $Y_1, \ldots, Y_n$, we need to determine what is more likely: is this time series distributed according to p_0, or p_γ? To address this problem we prove the following theorem.

Theorem 1. *Among all changes that lower the mean of an ARMA stochastic processes, the optimal classification algorithm in the Neyman-Pearson sense is to raise an alarm if $\bar{\epsilon}^2$ is greater than a threshold τ: where $\bar{\epsilon} = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i$, ϵ_k is the innovation process*

$$Y_k - \mathbb{E}_0[Y_k | Y_1, \ldots, Y_{k-1}] \text{ where } \mathbb{E}_0 \text{ is the expectation under probability } p_0, \tag{9}$$

and where we assume $\bar{\epsilon}$ is smaller than zero. (If $\bar{\epsilon} \geq 0$ then we decide that there is no attack.)

Proof. An optimal classification algorithm in the Neyman-Pearson sense refers to a classifier that given a fixed false alarm rate, will maximize the probability of detection. Given two probability distributions p_0 and p_γ defining the distribution of Y_i under each class, the optimal classifier in the Neyman-Pearson sense is a likelihood ratio test:

$$\ln \frac{p_\gamma(Y_1, \ldots, Y_n)}{p_0(Y_1, \ldots, Y_n)} = -\frac{\gamma}{\sigma} \sum_{i=1}^{n} (\epsilon_i + \frac{\gamma}{2}) \tag{10}$$

However, we do not know the value of γ as an attacker can choose any arbitrary value. Therefore we need to use the Generalized Likelihood Ratio (GLR) test to find the maximum likelihood estimate of γ given the test observations $Y_1, \ldots, Y_n$.

$$\ln \frac{\sup_{\gamma>0} p_\gamma(Y_1, \ldots, Y_n)}{p_0(Y_1, \ldots, Y_n)} = \max_{\gamma>0} \sum_{i=1}^{n} (-\frac{\epsilon_i \gamma}{\sigma} - \frac{\gamma^2}{2\sigma}) \tag{11}$$

To find the maximum (assuming the constraint $\gamma > 0$ is not active):

$$\frac{\partial f}{\partial \gamma} = \sum_{i=1}^{n}(-\frac{\epsilon_i}{\sigma} - \frac{\gamma}{\sigma}) = 0 \text{ which implies } \gamma = -\sum_{i=1}^{n}\frac{\epsilon_i}{n} = -\bar{\epsilon} \tag{12}$$

as long as $\gamma > 0$ (i.e., the optimization constraint is not active).

Therefore, the final GLR test (if $\bar{\epsilon} < 0$) is:

$$\frac{\bar{\epsilon}}{\sigma}\sum_{i=1}^{n}(\epsilon_i - \frac{\bar{\epsilon}}{2}) = \frac{\bar{\epsilon}^2}{\sigma}(n - \frac{1}{2}) \tag{13}$$

and since $\frac{1}{\sigma}(n - \frac{1}{2})$ is constant for the test, we obtain our final result.

By using one-step-ahead forecast, we calculate the innovation process ϵ_i. The threshold τ is determined based on the maximum $\bar{\epsilon}^2$ observed in the training dataset. The optimal attack strategy is as follows:

1. Calculate $E = \sqrt{\tau}$
2. Send $\hat{Y}_i = \mathbb{E}_0[Y_i|\hat{Y}_1, \ldots, \hat{Y}_{i-1}] - E$

where $\mathbb{E}_0[Y_i|\hat{Y}_1, \ldots, \hat{Y}_{i-1}]$ is the predicted ith value based on the observed measurements (including crafted ones) by the ARMA model.

3.3 Nonparametric Statistics

A concern regarding the ARMA-GLR detector is that it is only guaranteed to be optimal if ARMA processes can be used to model accurately normal electricity consumption behavior and attack patterns. To address these concerns we evaluate two more algorithms: nonparametric statistics (in this section) and unsupervised learning (in the following section).

Nonparametric statistics are robust to modeling errors: they have better classification accuracy when our model assumptions for the time-series is not accurate enough. This is a particularly important property for security problems, as we generally do not have good knowledge about the probability distribution properties of attacks.

A number of nonparametric algorithms have been designed to detect changes in the mean of a random processes. In this work we consider EWMA (Exponentially-weighted Moving Average) control chart [1] and Non-parametric CUSUM [8]. Because of space constraints and the fact that nonparametric test did not perform well in our experimental results, we omit the implementation details in this section and just give a brief overview of each detector and our attack.

A detector based on EWMA chart can be defined as $EWMA_i = \lambda Y_i + (1 - \lambda)EWMA_{i-1}$ where λ is a weighting factor and $0 < \lambda \leq 1$ and Y_i is one of the time series measurements (i.e. meter readings). An alarm is raised if $EWMA_i < \tau$, where τ is a configurable parameter. An attacker with knowledge of τ can create an attack as follows: While $EWMA_{i-1} > \tau$, send $\hat{Y}_i =$

$MAX(0, \frac{\tau-(1-\lambda)EWMA_{i-1}}{\lambda})$. When $EWMA_{i-1} = \tau$, send $\hat{Y}_i = \tau$. The idea here is that, before the EWMA statistic hits the threshold, an attacker attempts to reduce the meter-reading value as much as possible, and once it reaches τ, the attacker sends τ.

On the other hand, the Non-parametric CUSUM statistic for detecting a change in the mean of a random process is defined by $S_i = MAX(0, S_{i-1} + (\mu - Y_i - b))$ $(i = 1, \ldots, N)$, where μ is the expected value of the time-series, and b is a "slack" constant defined so that $\mathbb{E}[|\mu - Y_i| - b] < 0$ under normal operation. An alarm is raised if $S_i > \tau$. Our attack against this CUSUM-based detector is as follows: Calculate $M = \frac{\tau+Nb}{N}$ and send $\hat{Y}_i = \mu - M$. Note that this attack can take advantage of the total margin calculated as $\tau + Nb$.

3.4 Unsupervised Learning

One of the most successful algorithms for anomaly detection is the Local Outlier Factor (LOF) [7]. In our experiments we used *RapidMiner* [3] to calculate LOF scores. A *LOF detector* is implemented as follows:

1. Create a vector containing all measurements of a day to be tested in order, $V_{test} = \{Y_1, \ldots, Y_N\}$ where N is the number of measurements per day.
2. For all days in a training dataset, create vectors in the same way, $V_i = \{X_{i1}, \ldots X_{iN}\}$ $(i = 1, \ldots, T)$.
3. Create a set containing V_{test} and all V_is, and apply LOF to this set.
4. If $LOF_{test} < \tau$ where LOF_{test} is a score corresponding to V_{test}, conclude V_{test} is normal and exit.
5. If $\bar{Y}(= \frac{1}{N}\sum_{i=1}^{N} Y_i) < \frac{1}{NT}\sum_{i=1}^{T}\sum_{j=1}^{N} X_{ij}$, raise an alarm.

Because a high LOF score just implies that the corresponding data point is considered an outlier, we can not immediately conclude that high LOF score is a potential energy theft. In order to focus on detecting energy theft we only consider outliers with lower than average energy consumption.

While we are not able to prove that the following attack against our LOF detector is optimal because of the complexity of LOF, in the experimental section we show how our undetected attack patterns for LOF were better than the optimal attacks against other algorithms.

1. Among daily records in the training dataset whose LOF scores are less than τ, pick the one with the minimum daily sum, which we denote $\{Y_1^*, \ldots, Y_N^*\}$.
2. Find the maximum constant B such that $\{\hat{Y}_1, \ldots, \hat{Y}_N\}$, where $\hat{Y}_i = Y_i^* - B$, does not raise an alarm.
3. Send $\hat{Y}_i$.

4 Experimental Results

We use real (anonymized) meter-reading data measured by an electric utility company during six months. The meter readings consisted of 108 customers

with a mix of residential and commercial consumers. The meter readings were recorded every 15 minutes. Because our dataset contains measurements that were sent immediately after installation, we assume the meter readings during this period are legitimate.

4.1 Adversarial Evaluation: Cost of Undetected Attacks

To complete the evaluation proposed in Section 2, we now define the cost function C as follows:

$$C(Y, \hat{Y}) = MAX(\sum_{i=1}^{N} Y_i - \hat{Y}_i, 0)$$

where $Y = \{Y_1, \ldots, Y_N\}$ is the actual electricity usage and $\hat{Y} = \{\hat{Y}_1, \ldots, \hat{Y}_N\}$ is the fake meter reading crafted by an attacker.

Note that, if the actual usage is very small, the term $\sum_{i=1}^{N} Y_i - \hat{Y}_i$ can become negative, which means that an attacker will pay more money. We assume that a sophisticated attacker can turn the attack off and let real readings go unmodified when the actual electricity consumption is expected to be lower than the crafted meter readings. Under this strategy, the cost is always positive or equal to 0.

There are a number of ways to configure an electricity-theft detector. Ideally we would like to train anomaly detections with seasonal information, but given that our data only covers half a year, experiments in this section focus on a setting where electricity theft detectors are re-trained daily based on the last T-days data.

The experiments are conducted as follows. For each customer,

1. Set $i = 0$
2. Pick records for 4 weeks starting at the ith day as a training dataset (i.e. $T = 28$).
3. By using this training data set, compute parameters, including τ.
4. Pick a record of a day just after the training dataset as testing data.
5. Test the data under the detection model trained to evaluate false positive rate. If the result is negative (i.e. normal), attacks are mounted and the cost of the undetected attack is calculated.
6. Increment i and go back to Step 2.

Given the limited set of data we had, finding the optimal training length is outside the scope of this work. We chose a 4-week sliding window because we saw on preliminary results that it gave us a good trade-off between average loss and false alarms. As we obtain more data, we plan to consider in future work year-long datasets so we can fit seasonal models into our experiments and analyze in-depth the optimal training length.

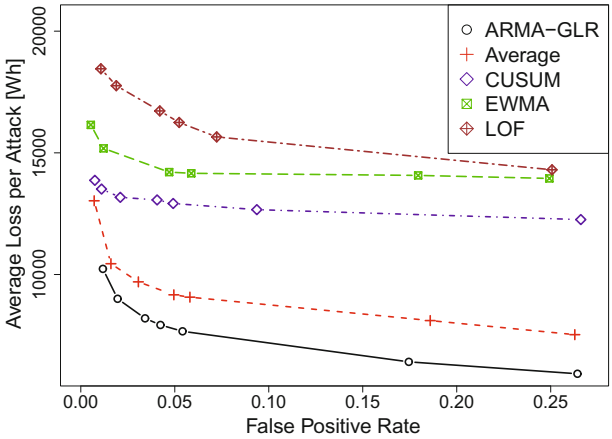


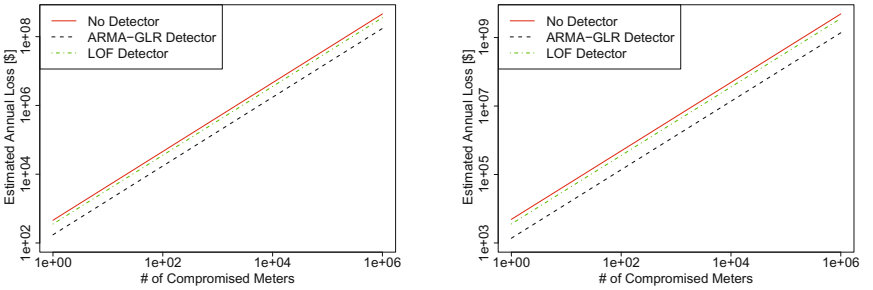
Fig. 1. Trade-off between false positive rate (the probability of false alarm) and average cost per attack

For each detector, we conducted 2,808 tests using the dataset of 108 customers, and for the cases that were claimed negative, we mounted attacks. The results are summarized in the trade-off curves in Fig. 1. The average cost per attack is calculated by dividing the total cost by the number of attacks performed.

As can be seen from the figure, the ARMA-GLR detector worked well. The average detector also is effective, but when the false positive rate is around 5%, its cost is higher than ARMA-GLR by approximately 1 KWh. It was somewhat surprising that the average detector outperformed the two online detectors: CUSUM and EWMA. One of the problems of these detectors is that they are designed to detect changes in a random process as quick as possible, and while this might have advantages for quick detection, it forced us to set very high thresholds τ to prevent false alarms in the 4-week-long training dataset. Detectors like ARMA-GLR and the average detectors on the other hand, smooth out sudden changes and are not susceptible to short-term changes of the random process. The cost of the LOF detector is the largest for all false positive rates evaluated.

Monetary Loss Caused by Undetected Electricity Theft. While assigning monetary losses to computer attacks is a difficult problem, one of the advantages of the dataset we have is that our data is directly related to the source of revenue of the utility company, and thus, it is easier to quantify the costs of potential attacks.

Using the electricity consumption rate charged by the utility company during the period of time we have available (while the utility company offers time-of-use prices, the tested customers belong to the flat rate program) we calculated that the (lower-bound) average revenue per-customer per-day is $1.256 dollars.



(a) Average customer (b) Customer with largest consumption

Fig. 2. Estimated annual loss over # of compromised meters with 5% false positive rate. Notice that the x and y axes are in log-scales.

From the experimental results, we picked one result whose false positive rate in the testing dataset is nearly 5% for each detector (we cannot achieve exact 5% in all detectors), and then, we calculated the average monetary loss for optimal attacks per-customer per-day. The result is summarized in Table 1. 5% false positive rate may seem too high, but utilities do not have investigate them equally. Instead, they could choose to focus on large-consumption customers to protect major portion of their revenue.

Table 1 shows that (at 4.2% false alarm rate), ARMA-GLR can protect 62% of the revenue of the utility against optimal attacks, while the remaining detectors fair much worse, most of them even protecting less than 50% of the revenue at higher false alarm rates.

While in practice detecting electricity theft is a much more complex problem (as mentioned in the introduction it involves the use of balance meters and personnel inspections), and the anomaly detection tests considered in this paper should only be considered as *indicators* of theft, and not complete proof of theft, we believe these numbers are helpful when utility companies create a business case for investments in security and revenue protection. For example, we can study the average losses as the number of compromised meters increases (Fig. 2(a)). In this example we notice that the losses reported in studies about electricity theft [14,4] would require about 10,000 randomly compromised meters. However, if we look at the losses caused by the top electricity consumers (commercial meters) (Fig. 2(b)), the same amount of losses can be achieved by about 100 compromised meters (or close to 10,000 compromised meters if we use ARMA-GLR detectors). While prices of electricity vary globally, we can infer that to achieve the losses previously reported, a large portion of hacked meters must correspond to large commercial consumers.

Table 1. Monetary loss caused by undetected electricity theft (5% false positive rate)

Detector	FP Rate	Average Loss	Revenue Lost
Average	0.0495	$0.55	43%
EWMA	0.0470	$0.852	68%
CUSUM	0.0491	$0.775	62%
LOF	0.0524	$0.975	77%
ARMA-GLR	0.0423	$0.475	38%

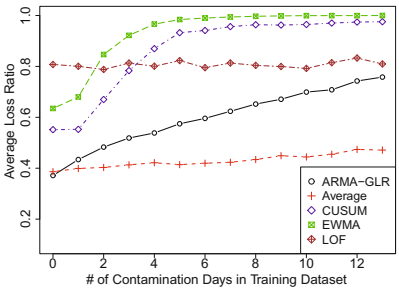


Fig. 3. Average loss ratio under contamination attack

4.2 Adversarial Learning: Detecting Contaminated Datasets

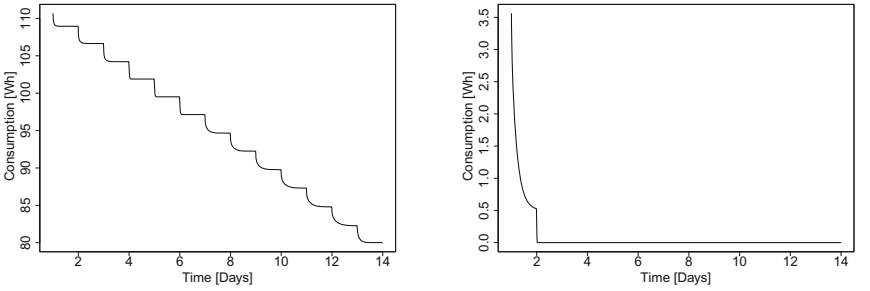
To evaluate the impact of contamination attacks discussed in Section 2.2, we show experiments using the same configuration as the ones used in the previous section. In this experiment, the optimal undetected attack is fed back into future training datasets. Namely, the training dataset of the second day includes an attack generated in the previous day, the training dataset of the third day contains two attack patterns generated for the first and the second day, and so forth.

We ran experiments for three disjoint time periods and calculated their average. The results are shown in Fig. 3. As can be seen from the plot, we can see increasing trends for all detectors except the LOF detector, which implies that LOF is more resilient to contamination attacks. In addition, the impact on the ARMA-GLR detector is much more significant than for the average detector. An intuitive explanation for this result is that ARMA models capture trends (unlike the average detector) therefore if we continue training ARMA models with a trend towards lower electricity consumption provided by an attacker, then the ARMA-GLR test will assume that future downward trends are normal.

Possible Countermeasures. A typical contamination attack pattern for the ARMA-GLR detector has the shape like the one shown in Fig. 4(a), in which we can see "roughly" a linear decreasing trend. A similar trend can be found in the case of other detectors. A straightforward way to identify such a pattern is fitting a linear model for the entire (or part of) a training dataset. We can expect that the resulting model would have negative slope significantly larger than other non-hacked customers. We applied linear regression for the contamination attack pattern of each customer. We also did the same for non-hacked meters for comparison. The results are summarized in Fig. 5(a). Though all of the attack patterns have negative slope, Fig. 5(a) shows this alone is not discriminative enough. Fortunately, we can find a clear difference in determination coefficients

(R^2) shown in Fig. 5(b)—determination coefficients are a measure of how well can a regression predict future values. High R^2, say $R^2 > 0.6$, with negative slope effectively indicates the existence of attacks. We manually investigated the attack patterns with low R^2 (those lower than 0.6) and found that all of them hit zero in the middle. For instance, the attack pattern shown in Fig. 4(b) gets to zero very quickly and remains at zero afterwards. Consecutive zeros is an indication of an anomaly and many utilities flag these events already, so the only attacks that will not be detected by the determination coefficients will be discovered by traditional rules.

The approach using linear regression also worked for other detectors since optimal attacks against them result in the similar, monotonically decreasing trends. While a motivated attacker can try to contaminate the training dataset at a slower pace so it is not detected, this will severely increase its effort and lower the effectiveness of its attacks.



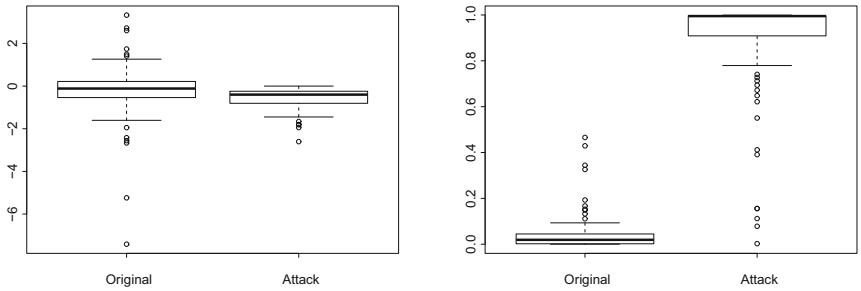
(a) Contamination attack with high R^2 (b) Contamination attack with low R^2

Fig. 4. Attack patterns under 14-day contamination attack experiment

5 Discussion

5.1 Cross-Correlation among Customers to Improve Detectors

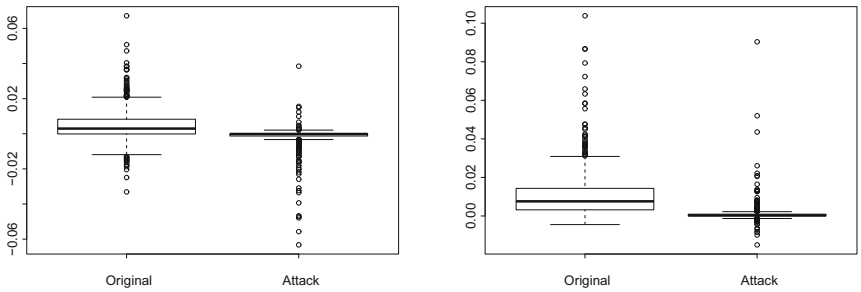
One possible way to identify attacks is to use cross-correlation among customers in nearby areas, assuming that honest customers exhibit similar trends while malicious customers have trends different from theirs. To evaluate this strategy, assuming that all 108 customers in the dataset are in the same neighborhood, we picked 7 daily consumption patterns from each of 108 customers and calculate cross covariance with the remaining 107 consumption patterns of the same day. Then, the average and quantile of these 107 cross covariances is calculated. Similarly, we calculated cross covariance between an attack pattern against the ARMA-GLR detector (Section 3.2) and original consumption patterns of the other 107 customers.



(a) Distribution of slopes of fitted linear Models

(b) Distribution of determination Coefficients of fitted linear models

Fig. 5. Distribution of slopes and determination coefficients of contamination attack patterns under linear model



(a) Average

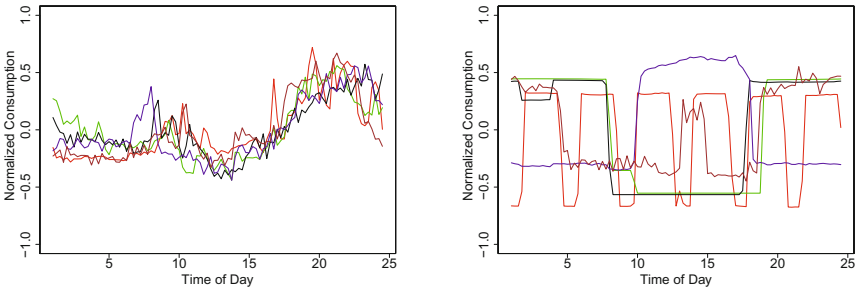
(b) 75 Percentile

Fig. 6. Distribution of average and 75 percentile of cross covariances

While we did not see significant difference in terms of 25 percentile and median of the 107 cross covariance values, their average and 75 percentile could be useful. Fig. 6 implies that a crafted attack pattern tends to exhibit a trend different from many of other customers' consumption patterns. Even though this alone can not be considered as definitive indication of attack, we could use it as an additional factor for electricity theft detection leveraging alarm fusion technologies.

In addition to cross-correlation, we can use outlier detection algorithms such as LOF [7], to identify outliers, exhibiting different trends in their electricity consumption patterns when compared to other similar consumers. In this direction, we have conducted some preliminary analyses. We smoothed daily electricity consumption patterns of a certain day in our dataset by using a low-pass filter. Then we normalized them since our focus here is anomaly in terms of shape and trends, not necessarily high or low consumption anomalies. Fig. 7 shows some samples of consumption patterns with top-5 (greater than 2.4) and low

(less than 1.0) LOF scores. While inliers with low LOF scores are categorized into a couple of "typical" usage patterns, like the one shown in Fig. 7(a), we can identify unique patterns, including "artificially-looking" ones (Fig. 7(b)). We will continue this area of research in future work.



(a) Typical patterns identified by LOF (b) Outliers identified by LOF

Fig. 7. LOF can Find Unusual Activity Patterns

5.2 Use of Auto-correlation of Residuals in ARMA-GLR Detector

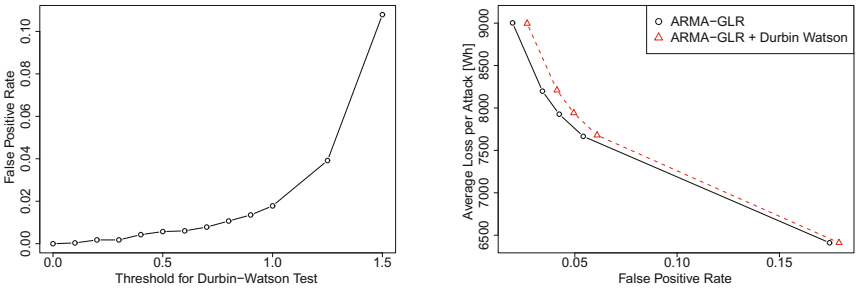
Based on the definition of attack strategy we tested in Section 3.2, we can expect that the sequence of residuals of generated attack patters have high auto-correlation, which can be a possible indication of attack. We have also explored this direction.

One of the possible metrics to quantify such auto-correlation is the Durbin-Watson statistic, defined as $d = \frac{\sum_{i=2}^{N}(e_i - e_{i-1})^2}{\sum_{i=1}^{N} e_i^2}$, where e_i denotes the ith residual and N is the number of measurements in the series. In general, we can infer that there exists auto-correlation when $d < 1.0$. Following this idea, we added the test of auto-correlation in residuals for the ARMA-GLR detector. Namely, for time-series patterns that passed the GLR test, we apply the test based on Durbin-Watson statistics. Using this approach we found that by setting the threshold for d around 1.0, it can detect all of the attacks mounted against ARMA-GLR as discussed in 3.2. The empirical relation between threshold values and the false positive rate, where false positives are counted only based on Durbin-Watson test (i.e. regardless of the result of ARMA-GLR test), is shown in Fig. 8(a).

Although we found that the use of Durbin-Watson statistics is effective to detect attacks against the ARMA-GLR detector, unfortunately it is not difficult to create attacks to defeat this other measure. For instance, a slightly modified attack strategy shown below would give attackers almost the same gain as the one he could do in case of the ARMA-GLR detector. When τ is the threshold used for the ARMA-GLR test,

1. Calculate $E = \sqrt{\tau}$
2. When i is an even number, send $\hat{Y}_i = \mathbb{E}_0[Y_i|\hat{Y}_1, \ldots, \hat{Y}_i] - 2E$. Otherwise, just send $\hat{Y}_i = \mathbb{E}_0[Y_i|\hat{Y}_1, \ldots, \hat{Y}_i]$.

This attack generates a sequence of residuals where 0 and $2E$ appear alternatively and results in having d approximately 2.0, which implies that attack can not be detected based on the threshold that is usually set around 1.0, while the total gain of an attacker is almost equal. As can be seen in Fig. 8(b), the trade-off curves are very similar. The weakness of the Durbin-Watson statistic is that it only considers first-order auto-correlation, so using higher-order correlation, such as Breusch-Godfrey Test or Ljung-Box Test, would make attacks harder. We will continue exploring ways to improve our detectors against sophisticated attackers in future work.



(a) False positive rates for Durbin-Watson statitics.

(b) Trade-off curves of ARMA-GLR and ARMA-GLR + Durbin Watson.

Fig. 8. Plots related to Durbin-Watson tests

5.3 Energy Efficiency

One of the goals of the smart grid is to give incentives for users to reduce their electricity consumption. In some cases (such as the installation of solar panels), the electric utility will know the consumer has installed these systems because the utility has to send personnel to approve the installation and allow them to sell electricity back to the grid. However, in some other cases, the incorporation of other green-energy technology might be unknown to the utility. In this case any anomaly detection algorithm will raise a false alarm. The best we can do is complement anomaly detection mechanisms with other information (e.g., balance meters) and in the case of false alarms, retrain new models with the new equipment in place. These changes are part of the non-stationarity of the random process we considered in this work.

6 Conclusions

In this paper we introduced the first rigorous study of electricity-theft detection from a computer-security perspective. While previous work has introduced other methods for electricity-theft detection, we argue that the incorporation of a new adversarial classification metric, and new mechanisms that consider adversarial learning are fundamental contributions to this growing area.

While all of the results in this paper consider pessimistic scenarios (the most-powerful attacker), we anticipate that these algorithms will perform much better under average cases where the attacker does not know the algorithms or time-intervals we use for anomaly detection and where it may not be able to compute optimal attack strategies. In addition, it is important to point out that the proposed anomaly detectors will only output *indicators* of an attack: a utility company will not only look at time-series anomalies as sources of attacks, but also at balance meters, smart meter tampering alarms, and might send personnel for periodic field monitoring reports. Combining all this information will give the utility good situational awareness of their network and accurate electricity-theft reports.

We plan to continue extending our work in multiple directions. For instance, optimal attacks are often artificial: e.g., the attacks against our average detector are constant values, therefore, adding additional mechanism that take advantage of the "shape" of the signal would be effective. We also plan to study more in-depth cross-correlation among nearby customers as an indicator of anomalies. Another approach to design classifiers resilient to attackers include the addition of randomness so the attacker cannot know at any time the state of the classifier. One example can be to leverage randomness in the use of training data, so an attacker would not know the exact configuration of the classifier. Finally, as we obtain datasets containing longer-periods of time, we plan to leverage accurate seasonal models and correlation with other factors, such as weather and temperature.

Acknowledgements. We would like to thank the reviewers and our shepherd, Guillaume Hiet, for insightful comments to improve this manuscript.

References

1. EWMA Control Charts, `http://itl.nist.gov/div898/handbook/pmc/section3/pmc324.html`
2. forecast package for R, `http://robjhyndman.com/software/forecast/`
3. RapidMiner, `http://rapid-i.com/`
4. Antmann, P.: Reducing technical and non-technical losses in the power sector. Technical report, World Bank (July 2009)
5. Appel, A.: Security seals on voting machines: A case study. ACM Transactions on Information and Systems Security 14, 1–29 (2011)
6. Bandim, C., Alves Jr., J., Pinto Jr., A., Souza, F., Loureiro, M., Magalhaes, C., Galvez-Durand, F.: Identification of energy theft and tampered meters using a central observer meter: a mathematical approach. In: 2003 IEEE PES Transmission and Distribution Conference and Exposition, vol. 1, pp. 163–168. IEEE (2003)

7. Breunig, M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104. ACM (2000)
8. Brodsky, B., Darkhovsky, B.: Non-Parametric Methods in Change-Point Problems. Kluwer Academic Publishers (1993)
9. Davis, M.: Smartgrid device security. adventures in a new medium (July 2009), http://www.blackhat.com/presentations/bh-usa-09/MDAVIS/BHUSA09-Davis-AMI-SLIDES.pdf
10. De Buda, E.: System for accurately detecting electricity theft. US Patent Application 12/351978 (January 2010)
11. Depuru, S., Wang, L., Devabhaktuni, V.: Support vector machine based data classification for detection of electricity theft. In: Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES, pp. 1–8 (March 2011)
12. ECI Telecom. Fighting Electricity Theft with Advanced Metering Infrastructure (March 2011)
13. Geschickter, C.: The Emergence of Meter Data Management (MDM): A Smart Grid Information Strategy Report. GTM Research (2010)
14. Krebs, B.: FBI: smart meter hacks likely to spread (April 2012), http://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/
15. Lesser, A.: When big IT goes after big data on the smart grid (March 2012), http://gigaom.com/cleantech/when-big-it-goes-after-big-data-on-the-smart-grid-2/
16. McLaughlin, S., Podkuiko, D., McDaniel, P.: Energy Theft in the Advanced Metering Infrastructure. In: Rome, E., Bloomfield, R. (eds.) CRITIS 2009. LNCS, vol. 6027, pp. 176–187. Springer, Heidelberg (2010)
17. McLaughlin, S., Podkuiko, D., Miadzvezhanka, S., Delozier, A., McDaniel, P.: Multi-vendor penetration testing in the advanced metering infrastructure. In: Proceedings of the Annual Computer Security Applications Conference (ACSAC) (December 2010)
18. Nagi, J., Yap, K.S., Tiong, S.K., Ahmed, S.K., Mohamad, M.: Nontechnical loss detection for metered customers in power utility using support vector machines. IEEE Transactions on Power Delivery Systems 25(2), 1162–1171 (2010)
19. Nizar, A., Dong, Z.: Identification and detection of electricity customer behaviour irregularities. In: Power Systems Conference and Exposition (PSCE), pp. 1–10 (March 2009)
20. Peterson, D.: AppSecDC in review: Real-world backdoors on industrial devices (April 2012), http://www.digitalbond.com/2012/04/11/appsecdc-in-review/
21. Smart Grid Interoperability Panel, editor. NISTIR 7628. Guidelines for Smart Grid Cyber Security. NIST (August 2010)
22. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: IEEE Symposium on Security and Privacy (2010)