Author copy. Do not redistribute.

Disclaimer: This document is not the final version of the paper.
The final version can be found in the proceedings of the 12th International Conference on Quantitative Evaluation of Systems (QEST 2015)

PCA-Based Method for Detecting Integrity Attacks on Advanced Metering Infrastructure

Varun Badrinath Krishna, Gabriel A. Weaver, William H. Sanders

Information Trust Institute, Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign,
1308 West Main Street, Urbana, IL 61801, USA
{varunbk,gweaver,whs}@illinois.edu
http://iti.illinois.edu

Abstract. Electric utilities are in the process of installing millions of smart meters around the world, to help improve their power delivery service. Although many of these meters come equipped with encrypted communications, they may potentially be vulnerable to cyber intrusion attempts. These attempts may be aimed at stealing electricity, or destabilizing the electricity market system. Therefore, there is a need for an additional layer of verification to detect these intrusion attempts. In this paper, we propose an anomaly detection method that uniquely combines Principal Component Analysis (PCA) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to verify the integrity of the smart meter measurements. Anomalies are deviations from the normal electricity consumption behavior. This behavior is modeled using a large, open database of smart meter readings obtained from a real deployment. We provide quantitative arguments that describe design choices for this method and use false-data injections to quantitatively compare this method with another method described in related work.

Keywords: smart, meter, grid, anomaly, detection, principal, component, analysis, data, cyber-physical, AMI, PCA, SVD, DBSCAN, electricity, theft, energy, computer, communication, network, security

1 Introduction

The *Advanced Metering Infrastructure (AMI)* provides a means for communication between electric utilities and consumers. Smart meters are increasingly replacing traditional analog meters to enable the automated reading of electricity consumption and the detection of voltage variations that may lead to outages. For example, by 2018, the Illinois-based Commonwealth Edison Company will have installed 4 million smart meters in all homes and businesses in Northern Illinois [5].

AMI is perceived to provide other benefits, beyond describing the state of the electric distribution grid. For example, smart meters have been rolled out by electric utilities such as BC Hydro to detect electricity theft [2]. In 2010,

however, the Cyber Intelligence Section of the FBI reported that smart meters were hijacked in Puerto Rico, causing electricity theft amounting to annual losses for the utility estimated at $400 million [6].

In [20], we show that an attacker may be able to destabilize a real-time electricity market system by compromising the electricity price relayed to the Automated Demand Response (ADR) interfaces. Equivalently, it may be possible to destabilize the system by compromising smart meter consumption readings, causing suppliers to modify the electricity price accordingly. Electricity theft and destabilization of electricity markets are just two of several attacker goals that illustrate the need for effective intrusion detection systems. Other attacker models are discussed in [3].

It must be noted that smart meters, such as those manufactured by GE, are equipped with encrypted communication capabilities and tamper-detection features. However, reliance on those mechanisms is not a sufficient defense against cyber intrusions that exploit software vulnerabilities. In their *Cyber Risk Report 2015*, HP Security Research states that the enterprises most successful in securing their environments employ complementary protection technologies [8]. Such technologies work best in the context of the assumption that breaches will occur. By using all tools available and not relying on a single product or service, defenders place themselves in a better position to prevent, detect, and recover from attacks.

The anomaly detection methods presented in this paper assume that an attacker has compromised the integrity of smart meter consumption readings, and aim to mitigate the impact of such an intrusion. How the attacker can get into a position where he is capable of modifying communication signals is not a focus of this paper and is discussed in [9], [13], and [14]. Our aim is to verify the data reported to the utility by modeling the normal consumption patterns of consumers and looking for deviations from this model.

Our proposed method leverages Principal Component Analysis (PCA) [15]; anomaly detection methods that leverage PCA have been proposed in [4, 11, 18, 19]. However, these papers focus on classifying anomalies, such as network volume anomalies, that manifest themselves as spikes in the data. Electricity consumption behavior, however, tends to be naturally spiky. Therefore, these methods fail to detect actual anomalies in consumption, such as extended changes in consumption patterns.

We propose a method that leverages the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm and show that this algorithm [1, 7], when combined with PCA, effectively detects anomalies in electricity consumption data. There are three advantages to using this method. First, by extracting the principal components that retain the maximum amount of variance in the data, we extract underlying consumption trends that repeat on a daily or weekly basis. Principal components that account for lower variance essentially represent noise in the consumption behavior, and this noise is filtered out. Second, the first two principal components allow us to visualize a massive dataset in a 2-dimensional space. Anomaly detection can then be performed in a way that

can be visually verified. Third, the anomaly detection is performed in a space that spans the consumptions of all consumers. Therefore, it becomes significantly harder for an attacker to reverse-engineer and circumvent this detector, as he would need full information of all the consumers' smart meters in the network.

Online anomaly detection is an important feature that would enable better adoption of our method. For this feature, we leverage the technique in [17]. The results of our method could feed into recent cyber physical vulnerability assessment techniques such as [21], or be incorporated into a stand-alone tool.

Our model of consumption patterns is based on a large, open dataset that is described in Section 2. We propose and delineate our own anomaly detection method in Section 3, and evaluate this method against other well-known methods in Section 4. We conclude in Section 5.

2 Description of the Dataset

The dataset we use was collected by Ireland's Commission for Energy Regulation (CER) as part of a trial that aimed at studying smart meter communication technologies. It is the largest, publicly available dataset that we know of, and access details are provided in the Acknowledgments section of this paper. The fact that the dataset is public makes it possible for researchers to replicate and extend this paper's results.

The dataset is an anonymized collection of readings from 6,408 consumers, collected at a half-hour time resolution, for a period of up to 74 weeks. Of the 6,408 consumers, we restrict our analysis to the largest subset that contains the same 74 weeks, by calendar date. This restriction results in a set of 2,982 consumers, of which 2,374 were residential, 253 were small and medium enterprises (SMEs), and 355 were unclassified by CER.

3 Data-Driven Detection Strategies

In this section, we analyze the CER smart meter dataset and model electricity consumption patterns to aid in the detection of integrity attacks. We discuss two distinct detection strategies. The first is based on the average detector proposed in [12]. Given the limitations of that technique, we devised an alternative method, which is based on Principal Component Analysis (PCA). We discuss it in detail and quantify its effectiveness in Section 4.

The electricity consumption patterns in the CER dataset guide our anomaly detection methods. The authors of [12] admit that they arbitrarily evaluate detection strategies and use consumption models (such as the Auto-Regressive Moving-Average model) that do not capture actual electricity consumption patterns. In contrast, our detection methods stem from our analysis of consumption patterns in a dataset obtained from a real, large-scale deployment.

3.1 Assumptions and Notations Used in Detection

The detection strategies presented in this section look for anomalies in the smart meter readings that are reported to the utility. We assume that the meters are correctly measuring current, but the readings being communicated may have been compromised.

In the context of this paper, a detection strategy is a centralized online algorithm that would typically run at the utility control center and is defined as follows. The input is a set of new smart meter consumption readings that are reported to the utility. We refer to it as the *input set*, and it may contain one or more readings for each consumer. We refer to the output as the *classification* of the set, which is binary: normal or suspected attack. Note that the classification is based on an input set, and not on a single reported reading. So it is possible for the detection algorithm to classify individual readings as normal, but classify a combined set of such readings as anomalous. This may happen because of a deviation of the combined readings from the expected combined pattern.

We divided the 74 weeks of consumption data obtained from the CER dataset into two sets: a *training set* of the first 60 weeks and a *test set* of the remaining 14 weeks. Note that anomalies in the training set are not labeled, so we do not have ground truth on which readings are anomalous. As such, our algorithm is unsupervised, and our training set serves to build a model of the consumption patterns while accounting for the possibility of anomalies in it.

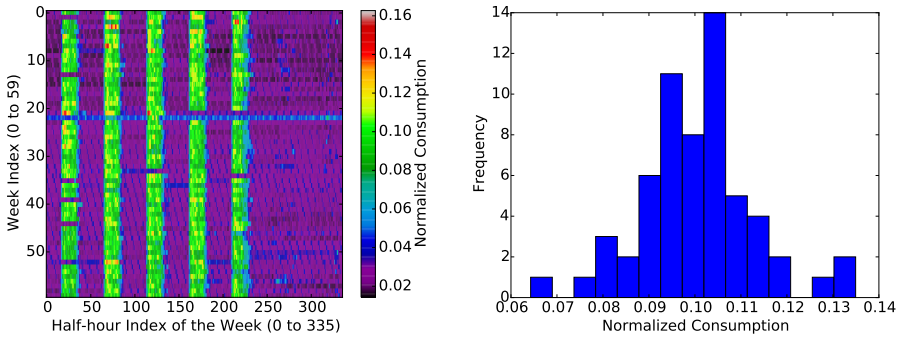
It is reasonable to assume that the training set obtained from CER is free from integrity attacks. However, there are anomalous consumption behaviors in the dataset. These anomalies might reflect periods when consumers were, for example, traveling, leading to abnormally low consumption, or hosting parties, leading to abnormally high consumption. Such events lead to false positives if the detection strategy classifies them as suspected attacks. The test set is used in Section 4 to evaluate false positives and false negatives reported by the detection algorithms using models built from the training set.

We use the following matrix notations in this section. $A_{(i,j)}$ refers to the element in matrix A at the intersection of row i and column j. $A_{(i,:)}$ refers to the row vector at row i, and $A_{(:,j)}$ refers to the column vector at column j.

3.2 Use of Averages to Detect Anomalies

The Average Detector was shown to be effective relative to the other methods proposed in [12]. This detector is formulated as follows. Let $D_c(t)$ represent the total consumption of consumer c during time period t. Given that our dataset contains smart meter measurements at a half-hour granularity, t refers to a particular half-hour. The consumption reported to the utility is denoted by $D'_c(t)$.

$D_c(t)$ is non-deterministic, and the true value at a certain time t is unknown to us, since our only knowledge of the value is through the reported reading $D'_c(t)$. As $D'_c(t)$ may be manipulated by integrity attacks on smart meter communications, we need to devise methods to validate $D'_c(t)$ for each t.



(a) Normalized consumption for all weeks for SME 1028

(b) Distribution of normalized consumption at half-hour index 71

Fig. 1: Normalized consumption of SME Consumer 1028. The five green/blue vertical bands in (a) represent time periods of higher electricity consumption that correspond to weekdays.

For each consumer c, we define an $M \times N$ matrix H_c in which each row represents one week, and there are M weeks. Each column represents the time of the week, and there are N times. In addition, we carefully align the weeks such that the first column of each matrix H_c refers to the first time period of a Monday, and the last column refers to the last time period of a Sunday. In our training set, H_c is a 60×336 matrix, as there are 60 weeks in the training data and 336 half-hours in a week.

Note that while consumption patterns typically repeat every week, the absolute consumption depends on the week. If the week is during winter in a country whose climate is like Ireland's, chances are that an occupant of the house will turn on the heating system. In contrast, the heating system will most likely be turned off during a week in the summer. To ensure that weeks can be compared on even grounds, we normalize each row in the matrix H_c, corresponding to the week, by dividing the row by its $l2$-norm. Fig. 1 illustrates the normalized H_{1028} matrix (consumer identity $c = 1028$). The fact that this consumer has nearly zero consumption on weekends indicates that it is likely not a residential consumer. Indeed, the CER dataset has labeled 1028 as an SME.

We define the half-hour index (HHI) of the time t as a mapping $HHI : t \rightarrow \{0, 1, ..., 335\}$. For example, if we want to know whether $D'_c(t)$ is anomalous when t is December 2, 2014 at 12 p.m., we determine that this date is a Tuesday and that the time of the week corresponds to an HHI of 71. Fig. 1(b) represents $P(D_{1028}|HHI(t) = 71)$. We do not make any assumptions on the underlying distribution, as we do not have the data necessary to construct a valid distribution. In previous work, we showed that a normal distribution is observed when D_c is conditioned on multiple parameters, such as $HHI(t)$, solar irradiation, external temperature, and building occupancy [10].

The detection algorithm for an input set is performed on a per-user basis as follows. For each consumer c_k, we calculate the average of the *input set* $IS_k =$

$\{D'_{c_k}(t_1), D'_{c_k}(t_2)...\}$, where the times t_j may index a single time point ($j = 1$), a day ($j \in [1, 48]$) or a week ($j \in [1, 336]$), etc. This produces a single average value $avg(IS_k)$ for the new data. We then compare this average to the averages taken over the same time points in all previous weeks for consumer c_k. For example, if IS_k contains the set of all consumption points on a Tuesday, we compare the average $avg(IS_k)$, with averages of every Tuesday in the history of the dataset. If $avg(IS_k)$ is less than the lowest (or greater than the highest) average seen in past Tuesdays, we say that the input set is anomalous. If IS_k is a singleton set containing, say, the consumption at 12 p.m. on a Tuesday, we compare it against a set of consumptions at 12 p.m. on previous Tuesdays. In this case, $avg(IS_k)$ is the same as the single value in IS_k, so the notation remains valid.

3.3 Detecting Anomalies with Principal Component Analysis

The drawback of the average detector is that an attacker can circumvent it by ensuring that the average of the input set for a consumer $avg(IS_k)$ does not change significantly. Specifically, the elements of the input set can vary in a manner that is not consistent with the typical consumption patterns, but this change of pattern will not be quickly detected if the average is kept within reasonable bounds. Therefore, there is a need for a method that analyzes the variation in the consumption pattern as a collection of meter readings, as opposed to individual meter readings. For this purpose, we propose using Principal Component Analysis (PCA), and to detect deviations from the pattern we propose the use of a clustering technique.

PCA reveals the underlying trends in the smart meter data, across thousands of consumers, by reducing the dimensionality of the data, while retaining most of the data's variance. As such, it provides us with a way we can collapse a vector of electricity consumption readings in a high-dimensional space into one in a lower-dimensional space. This greatly aids anomaly detection methods, which can be intuitively executed in the lower-dimensional space, without loss of significant information. PCA not only immediately reveals clusters in data, but also is sensitive to changes in consumption patterns that may indicate integrity attacks.

The PCA Mechanism We demonstrate the mechanism of PCA by constructing two different matrices (A & B) from our entire training set. A has $M_A = 20,160$ rows, one for each half-hour of the 60-week period of study, and $N_A = 2,982$ columns, one for each consumer. In this example, we can think of the consumption of each consumer across all 60 weeks as a column vector in a 20,160-dimensional space. There are 2,982 such column vectors. Using PCA, we will collapse these column vectors from $M_A = 20,160$ dimensions into two dimensions. Due to high correlation, two data points are sufficient to capture the patterns of each consumer, relative to the patterns of other consumers. Let P_A be the matrix of dimension $2 \times M_A$ that transforms A of dimension $M_A \times N_A$ to Y_A of dimension $2 \times N_A$. Then,

$$P_A A = Y_A \tag{1}$$

For calculation and notation convenience, we pre-process A by subtracting each row by the mean for that row and dividing the entire matrix by $N_A = 2,982$. We are interested in the covariance between the M_A rows (or readings per consumer) in A. For the corresponding AA^T covariance matrix, $P_A = U^T_{(0:1,:)}$, where U is obtained from the Singular Value Decomposition (SVD) of $A = U\Sigma V^T$. Here, the columns of U are the eigenvectors of the covariance matrix AA^T, and Σ^2 (the eigenvalues) represent the amount of variance retained by the principal components. This is illustrated in Fig. 2. Together, the two components in P_A retain 63.63% of the variance in A, and the marginal variance retained by each further component is negligible.

There are two advantages to retaining only the first two components. First, maximum variance is retained by these components, so discarding further components effectively discards the noise in the consumption patterns. Second, it allows us to visualize a vector of $20,160$ dimensions in a two dimensional space. This then facilitates anomaly detection in this 2D space, as we will discuss later.

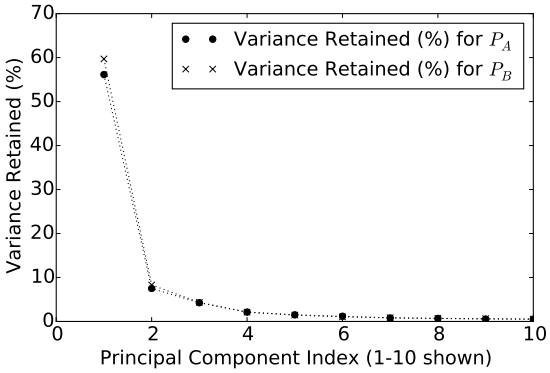
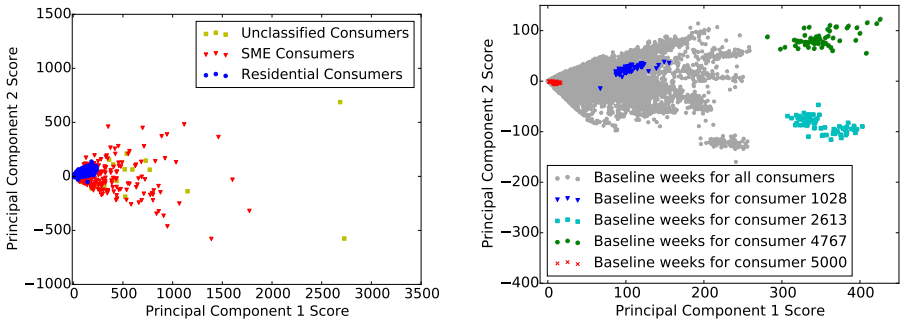


Fig. 2: Variance (%) retained by principal components of matrices A & B.

Construction of PCA Biplots We transform A into the P_A space by taking the product $P_A A = Y_A$, where Y_A is the $2 \times N_A$ PCA score matrix. The two rows of Y_A are called the *Principal Component 1 Score* and the *Principal Component 2 Score*. The scatter plot of the two scores is the PCA biplot shown in Fig. 3(a). Points that lie close together in this 2D space describe consumers, or columns in A, whose consumption patterns are similar. Given that the comparison is over 60 weeks at a half-hour granularity, the large extent to which the consumers cluster together in the biplot was unexpected, and indicates that most of the consumers in the dataset have highly similar consumption patterns.

In Fig. 3(a), we used the labels in the CER dataset to distinguish the points in the biplot by consumer type. These labels revealed an interesting behavior where most residential consumers were found to cluster together in the 2D space. This indicated that their consumptions were similar to each other. However,



(a) Observed clustering of consumers by type in the P_A space

(b) Observed clustering of consumers and their consumption weeks in the P_B space

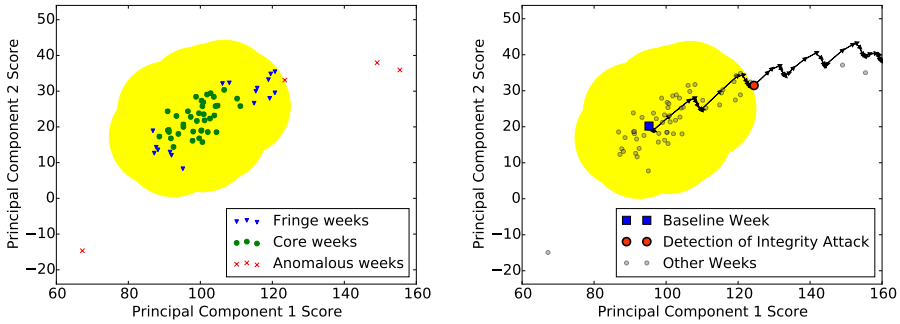
Fig. 3: Principal Component Analysis biplots describing the structure and similarities within the dataset.

SMEs varied greatly, which might reflect the unique electricity consumption requirements of their businesses.

In order to capture the relationship among consumers' individual patterns across different weeks, we reshaped A to get another matrix B; it contains $M_B = 48 * 7 = 336$ rows (one for each half-hour of the week) and $N_B = 2,982 * 60 = 178,920$ columns (one for each week of each consumer in the 60-week period). Again, we reduced the dimension of each week from $M_B = 336$ dimensions to 2 dimensions, retaining 68% of the variance in B as shown in Fig 2. The corresponding principal component matrix, P_B, is 2×336, and the PCA score matrix, $Y_B = P_B B$, is $2 \times N_B$.

Note that although A and B contain the same number of elements, their Principal Component Scores are of different dimensions and describe completely different characteristics of the data. The scores in Y_B tell us how similar the 60 weeks of consumption are in the training set across all consumers and they are plotted in Fig. 3(b). We observe a dense clustering of points corresponding to each consumer in the Y_B matrix, which captures how similar the consumption weeks are for each consumer, in comparison to weeks of other consumers. This can easily be seen in Fig. 3(b), where we have colored the points corresponding to four very different consumers and their consumption weeks.

A closer look at the weeks for consumer 1028 in Fig. 3(b) revealed a single blue triangle at around $(70, -15)$ in the plot that is significantly distant from the others. It corresponds to Week Index 23 in Fig. 1(a), which is clearly anomalous and probably a vacation week. There are other anomalous points that are distant from the dense cluster. As the anomalies are inherent in the dataset, we assume that they are natural anomalies, and not the consequence of attacks. Attacks, which modify the consumption signals in a manner that changes their pattern, cause a shift in the location of the original point (corresponding to a week) to a completely new one on the biplot, as shown in Fig. 4(b).



(a) Decision boundary and types of points in the DBSCAN clustering results

(b) Movement of baseline point with increase in attack duration

Fig. 4: Principal Component Analysis biplots for Consumer 1028 capturing (a) the decision boundary for anomalous points and (b) the movement of a baseline week of consumption in the principal component space with increase in duration of an integrity attack (Random Scale Attack, discussed in Section 4.3).

Clustering Points in the Principal Component Space A natural density-based clustering of points in the principal component space is observed in Fig. 3. Therefore, we employ the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [7] to determine which points correspond to regular weeks and which points correspond to anomalous weeks. An inherent benefit of DBSCAN lies in the fact that it works well for irregular geometries of dense clusters, and that it does not assume any underlying probability density of the points. The non-convex boundaries and treatment of dense clusters make DBSCAN better suited to our application, as opposed to other hierarchical, centroid-based, and distribution-based clustering methods.

The DBSCAN algorithm has two configurable parameters: *eps* and $MinPts$. These are used to obtain dense neighborhoods. In a two-dimensional Euclidean space, such as our principal component space, the circular region of radius *eps* centered at a point is referred to as the *eps neighborhood* of the point. A *core point* is a point that contains $MinPts$ points within its *eps* neighborhood. All points that lie within the *eps* neighborhood of a core point are considered members of a dense cluster.

In our specific case, we are clustering 60 points that correspond to the weeks of consumption, for each consumer in our training set. These points were extracted from Y_B. We define $MinPts$ as the number of points that achieves a simple majority (which in this case is 31). As a result, a single continuous cluster corresponding to normal weeks is produced, because any two *eps* neighborhoods containing $MinPts$ points must intersect at at least one point.

Points that lie within the *eps* neighborhood of a core point, but are not core points themselves, are referred to as *fringe points*, as they usually lie at the fringes of the dense neighborhood. The algorithm considers all points that are neither core points nor fringe points to be "noise." This noise is how we define anomalous points in the dataset.

Fig. 4(a) illustrates the result of the DBSCAN algorithm for consumer 1028. The green points are core points, and the blue triangles are fringe points. Since we chose *MinPts* to represent a simple majority, the circular *eps* neighborhoods overlap to form a single dense region, indicated by the yellow region in Fig. 4(a). All points within this region are considered to be normal. Anomalies, which may be caused by attacks, are points that do not lie in this region. The red crosses correspond to natural anomalies, which the algorithm would flag as false positives, as they lie outside the yellow "safe" region.

Clearly, the detection takes place in the 2D space spanned by P_B's basis vectors, which span the weekly consumptions of all consumers. In order to reverse-engineer the PCA detector for the purpose of circumventing it, the attacker would need to recreate this 2D space by gaining access to the meters of all consumers. In contrast, he would only need to compromise the smart meter of a single victim in order to reverse-engineer the Average Detector for that victim. *Therefore, the PCA-based detection method is more secure, because circumventing it requires full knowledge of all consumers' smart meter readings.*

Although the DBSCAN authors provide recommendations in [7] on how to set *eps*, these methods are not scalable. Specifically, they suggest calculating a list of core distances for each point and observing a knee-point at which a threshold should be set for *eps*. Given that there are $2,982$ sets of points in our dataset (one for each consumer), eyeballing knee-points for each set is not feasible, so we needed to find an alternative. OPTICS, described in [1], can be used to determine cluster memberships for a single set of points containing multiple clusters. However, this method is not suitable for our study, in which we are defining a single cluster per set in $2,982$ sets of points.

We set *eps* based on S_n, a measure proposed by statisticians in [16]. S_n looks at a typical distance between points, which makes it a good estimator of *eps*. In contrast, the Median Absolute Deviation (MAD) and the Mahalanobis distance measure the distance between points and a centroid, which is not how *eps* is defined. And, unlike the standard deviation, S_n is robust to outliers. In addition, S_n is applicable to asymmetric geometries of points, like those in 3(b).

4 Evaluation of Detection Methods

We used data-driven simulation methods to evaluate the performance of our PCA-based detection method for multiple consumers in our dataset. We present a quantitative comparison of the performance of this method with that of the Average Detector method.

4.1 Runtime Memory Cost of Implementing the Detectors

The memory cost of the PCA-based method depends on the size of the input set that needs to be verified. For each consumer, if the input set is a week of readings at a half-hour time resolution, the principal component matrix would have a 2×336 dimension. The dimensionality of the principal component matrix

remains the same as that of the input set. It does not increase as more input sets are evaluated, but can be updated without further memory costs, as shown in [17]. Also, it is not a function of how many consumers are in the system and thus occupies $O(1)$ memory.

In comparison, the Average Detector occupies $O(N)$ memory, scaling with the number of consumers in the system. For each consumer, this detector needs to access the minimum and maximum of the averages of each input set it ever processed. Therefore, this detector maintains 2 scalar values (a maximum and a minimum) per consumer.

In our evaluation, we do not consider the scenario where input sets are restricted to readings obtained from a single day. However, we briefly describe the procedure for evaluating such a scenario, because it might be useful in practice. In this scenario, we would separately perform PCA for each day of the week to obtain the two principal components. Therefore, our model for each day of the week would be a 2×48 matrix, as there are 48 half-hours in a day. Depending on the input set's day of the week, we would then rotate the input set into its corresponding principal component space. Following this, we would use the DBSCAN algorithm to detect whether the input set was an outlier. Note that we would need 7 principal component matrices in this case, one for each day of the week; the total memory requirement remains 2×336.

4.2 Evaluation Methodology

In order to evaluate both methods, we injected attacks that modify the smart meter readings. Our objective was to test the robustness of the two methods to such modifications, so we gradually varied the attack duration and recorded both false positives as well as false negatives. In this case, a false positive occurs when the input set is classified as a suspected attack, when it was actually not altered. A false negative occurs when the input set was compromised but the detection method classifies it as normal behavior.

Let $Attacked : IS_k \rightarrow \{0, 1\}$ be a function that takes the value 1 when the Input Set for a consumer IS_k is compromised and 0 when IS_k is not compromised. We measured the *false positive rate* (FPR) and *false negative rate* (FNR) defined in terms of probabilities as follows:

$$\begin{aligned} FPR &= P(Classification = Suspected\_Attack \quad | Attacked(IS_k) = 0) \\ FNR &= P(Classification = Normal\_Behavior \quad | Attacked(IS_k) = 1) \end{aligned} \tag{2}$$

In order to make a fair comparison between the two methods, we standardized the size of the input set to half-hour values over a week. Therefore, the input set contained 336 readings. We could have equivalently limited the standardized size of the input set to a day, but the stealthy attacks that we injected would take multiple days to be detected. A stealthy attack in this sense is one where the readings are not significantly altered, and examples are described in Section 4.3.

Our injection approach was broken down into two tests. In both tests, we constructed an initial input set containing the elements of any previous week in

the training data that corresponded to a core point according to DBSCAN. This set is guaranteed to be free from anomalies and was used to complete the input set with 336 readings. Beginning with the first half-hour element of this input set, we modified the consumptions in chronological order. The modifications were made differently for the two tests, as described next.

The first test was a test for false negatives and the readings were modified using the attack injection methods explained in Section 4.3. The ideal result would have been the classification of all input sets as suspected attacks. By modifying the input sets in a chronological sequence, we varied the *attack duration* from one half-hour period to the entire week. For each attack duration, a new input set was created. The *time to detection* (TTD) for a detector is captured by the input set that corresponded to the shortest detectable attack duration. Better detectors have smaller TTDs.

The second test was a test for false positives. In this case, the input set was constructed using readings from the *test set*, which contained 14 weeks from the CER dataset (see Section 3.1). For each half-hour index of the input set, we randomly picked a reading from the test set with the same half-hour index. This random choice was made from a discrete uniform distribution with range $[1, 14]$. The ideal result would have been the classification of all input sets as normal, since $P(Attacked(IS_k) = 0)$ for all of them.

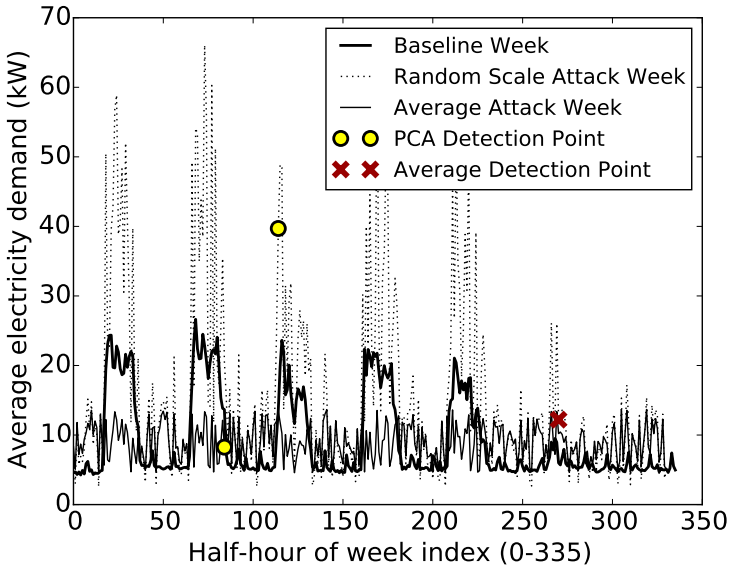


Fig. 5: Physical Manifestation of Random Scale ($\alpha = 0.5, \beta = 3$) and Average ($\gamma = 0.5$) Attacks on Consumer 1028. The attacks are launched at time 0. The PCA Detector has a TTD of 114 half-hours for the Random Scale Attack, and 84 half-hours for the Average Attack. The Average Detector has a TTD of 270 half-hours for the Random Scale Attack, and cannot detect the Average Attack.

4.3 Attack Injection Methods

In order to evaluate false negatives, we discuss two specific types of attack injection methods that modify the baseline week of readings (input set):

Random Scale Attack: At each half-hour time point, the consumption signal is multiplied by a uniform random variable $R \sim Unif(\alpha, \beta)$ where $0 < a < b$. The consumption is under-reported when R is a fraction below 1, and over-reported when it is a fraction above 1. If on average the values are over-reported, it can cause instability as shown in [20]. If they are under-reported, however, they can lead to electricity theft, as shown in [12].

Average Attack: At each half-hour time point, the consumption signal is replaced by the average value of the baseline week, multiplied by a uniform random variable $G \sim Unif(1-\gamma, 1+\gamma)$ where $\gamma \in [0, 1]$. The reported consumption effectively oscillates around the average. An attacker may use this method in a time-of-use electricity pricing scheme by under-reporting consumption readings when the price is high and over-reporting them when the price is low, while maintaining the average for a given day. This assumes that the electric utility uses redundant meters to verify aggregate consumptions at the end of the day.

Fig. 5 illustrates the physical manifestation of these attacks. As previously described, the false data was injected into the reported readings starting from half-hour index 0 all the way to half-hour index 355. This simulates the duration of the attack, and reveals the TTD for each detector.

For the PCA detector, the point, which corresponds to a week in the principal component space, moves as the attack duration increases. This movement, in discrete steps, is captured by the trajectory in Fig. 4(b). When the consumption pattern has been sufficiently disturbed, the point moves beyond the detection boundary defined by the DBSCAN algorithm. Beyond this duration (the TTD), the point continues to move farther away from the dense cluster and will continue to be classified as a suspected attack.

4.4 Results

We have thus far used Consumer 1028 for illustration purposes, and now extend our evaluation to all $2,982$ consumers in the dataset. The Random Scale Attack was simulated on all consumers while the Average Attack was simulated on a subset of $2,814$ consumers. This subset contained only those consumers who exhibited variation in the baseline consumption that was greater than the variation introduced by the Average Attack; specifically, the ratio of the standard deviation to the mean of the baseline week was greater than γ.

For each consumer, we created $2 * 336 * 1,000$ input sets, for a combination of 2 tests (false positive and false negative), 336 discrete attack durations (for 336 half-hours in a week), and $1,000$ trials to capture the range of the uniform random variables. Increasing the number of trials from 100 to $1,000$ resulted in a 5.4% increase in the range of means observed for a uniform random sample of 336 values. Further increasing the number of trials from $1,000$ to $100,000$ increased the range by just 6.2%. We therefore decided to use $1,000$ trials to

reduce the cost of the simulation without losing a large fraction of the range of the uniform random variables.

Given the large size of the simulation (it took around $3,840$ CPU hours to complete), we limited the attack parameter space to the values given in Fig. 5. We then calculated the FNR and TTD for the two different types of attacks. Table 1 captures the metrics across all consumers.

To save space in Table 1, we introduce some new notation. *RS Attack* is the Random Scale Attack and *A Attack* is the Average Attack. $dPCA$ denotes the event that an attack was successfully detected by the PCA detector within the 336 half-hour time frame in all $1,000$ trials. This event applies to each consumer. $P(dPCA)$ is a probability that denotes the fraction of consumers for whom this event held true. $dAVG$ is the corresponding event for the Average (AVG) detector. Result 1 (in Table 1) tells us that for 84.9% of consumers, the PCA detector was successful in detecting the Random Scale Attack. The PCA detector performs better against both attacks, as indicated in the *Win* column. Although the PCA detector did not perform as well as we had hoped for the Average Attack, it was a significant improvement on the AVG Detector.

Note that the detectors work in many of the $1,000$ trials conducted for each consumer, but we wanted to test robustness under the stochastic attacker behavior. Therefore, our results conservatively captured only the consumers for whom the detectors worked in all $1,000$ trials.

Results 2, 3, & 4 describe the TTD for the PCA and AVG detectors. The best case (min), average case (mean), and worst case (max) TTDs for the PCA detector are lower than the corresponding values for the AVG detector for most consumers, which again makes the PCA detector better. Note that the probability of having a higher TTD can be inferred from the probability of having equal and lower TTDs, which are given in the table. Result 5 compares the probability of having lower FNRs. Note that the remaining probability is accounted for by the case where the FNRs are equal.

Table 1: Evaluation Results for False Negative Tests

	RS Attack		A Attack	
Metric	Value	Win	Value	Win
1. $P(dAVG)$	0.635	PCA	0.040	PCA
$P(dPCA)$	0.849		0.081	
2. $P(mean(PCA_{TTD}) < mean(AVG_{TTD})\|dPCA \ \& \ dAVG)$	0.652	PCA	0.520	PCA
$P(mean(PCA_{TTD}) = mean(AVG_{TTD})\|dPCA \ \& \ dAVG)$	0.001		0.000	
3. $P(min(PCA_{TTD}) < min(AVG_{TTD})\|dPCA \ \& \ dAVG)$	0.689	PCA	0.480	TIE
$P(min(PCA_{TTD}) = min(AVG_{TTD})\|dPCA \ \& \ dAVG)$	0.025		0.040	
4. $P(max(PCA_{TTD}) < max(AVG_{TTD})\|dPCA \ \& \ dAVG)$	0.513	PCA	0.600	PCA
$P(max(PCA_{TTD}) = max(AVG_{TTD})\|dPCA \ \& \ dAVG)$	0.121		0.040	
5. $P(PCA_{FNR} < AVG_{FNR})$	0.630	PCA	0.079	PCA
$P(PCA_{FNR} > AVG_{FNR})$	0.232		0.033	

On false positive tests, the AVG detector outperformed the PCA detector overall. In fact, the AVG detector had a perfect result: $P(AVG_{FPR} = 0) = 1.0$ and $P(AVG_{FPR} = 1) = 0.0$. For the PCA detector, however, $P(PCA_{FPR} = 0) = 0.637$ and $P(0 < PCA_{FPR} < 1) = 0.363$. This means that for 63.7% of consumers, false positives were not detected in all $1,000$ trials. For the remaining consumers, the consumption patterns changed dramatically in the test set, leading to at least one false positive reported by the PCA detector in the $1,000$ trials. However, the consumption did not increase or decrease beyond the AVG detector thresholds, leading to the success of the AVG detector.

In summary, we have shown that the PCA detector probabilistically outperforms the AVG detector on false negative tests. We suspect that the PCA detector can be improved to reduce the false positive rate for the 36.3% of consumers. This might be achieved by correlating their consumption pattern deviations with deviations observed in the patterns of other consumers. In a vacation weak, for example, the PCA detector would suspect an attack due to deviation in consumption patterns. However, if other consumers show deviations for the same week, it provides evidence against classification as a suspected attack.

5 Conclusion

In this paper, we proposed a PCA-based anomaly detection method that utilities can use to detect integrity attacks on smart meter communications in an Advanced Metering Infrastructure. We provided quantitative arguments describing design choices for this method and presented a quantitative evaluation of the method with respect to the Average Detector proposed in related work.

In future work, we intend to use the framework developed in this paper to build a tool that can allow us to perform a more comprehensive evaluation of detection strategies under different attack parameters. Also, we plan to investigate the false positives of the PCA method by correlating simultaneous anomalies across multiple consumers.

Acknowledgments. This material is based upon work supported by the Department of Energy under Award Number DE-OE0000097. The smart meter data used in this paper is accessed via the Irish Social Science Data Archive - `www.ucd.ie/issda`. The providers of this data, the Commission for Energy Regulation, bear no responsibility for the further analysis or interpretation of it. We thank Shweta Ramdas, Jeremy Jones and Tim Yardley for their support.

References

1. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points to Identify The Clustering Structure. ACM SIGMOD Record 28(2) (Jun 1999)
2. BC Hydro: Smart metering program (2014), `"https://www.bchydro.com/energy-in-bc/projects/smart_metering_infrastructure_program.html"`

3. Berthier, R., Sanders, W.H., Khurana, H.: Intrusion Detection for Advanced Metering Infrastructures: Requirements and Architectural Directions. In: Proceedings of IEEE SmartGridComm '10. pp. 350–355. IEEE (Oct 2010)
4. Brauckhoff, D., Salamatian, K., May, M.: Applying PCA for traffic anomaly detection: Problems and solutions. In: Proceedings of IEEE INFOCOMM '09 (2009)
5. ComEd: Smart meter (2015), `https://www.comed.com/technology/smart-meter-smart-grid/Pages/smart-meter.aspx`
6. Cyber Intelligence Section: Smart grid electric meters altered to steal electricity (May 2010), `http://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/`
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of KDD'96. vol. 96, pp. 226–231 (1996)
8. HP Security Research: Cyber Risk Report 2015 (2015)
9. Jiang, R., Lu, R., Wang, L., Luo, J., Changxiang, S., Xuemin, S.: Energy-theft detection issues for advanced metering infrastructure in smart grid. Tsinghua Science And Technology 19(2), 105–120 (April 2014)
10. Jung, D., Badrinath Krishna, V., Temple, W.G., Yau, D.K.: Data-driven evaluation of building demand response capacity. In: Proceedings of IEEE SmartGridComm'14. pp. 547–553. IEEE (2014)
11. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. In: Proceedings of ACM SIGCOMM '04. ACM, New York, NY, USA (2004)
12. Mashima, D., Cardenas, A.A.: Evaluating electricity theft detectors in smart grid networks. In: Proceedings of RAID'12, vol. 7462. Springer Berlin Heidelberg (2012)
13. McLaughlin, S., Holbert, B., Zonouz, S., Berthier, R.: AMIDS: A multi-sensor energy theft detection framework for advanced metering infrastructures. In: Proceedings of SmartGridComm'12. pp. 354–359 (Nov 2012)
14. McLaughlin, S., Podkuiko, D., Miadzvezhanka, S., Delozier, A., McDaniel, P.: Multi-vendor penetration testing in the advanced metering infrastructure. In: Proceedings of ACSAC'10. pp. 107–116. ACM, New York, NY, USA (2010)
15. Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6 2(11), 559–572 (1901)
16. Rousseeuw, P.J., Croux, C.: Alternatives to the Median Absolute Deviation. Journal of the American Statistical Association 88(424), 1273–1283 (1993)
17. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Fifth International Conference on Computer and Information Science. Citeseer (2002)
18. Shyu, M.L., Chen, S.C., Sarinnapakorn, K., Chang, L.: A Novel Anomaly Detection Scheme Based on Principal Component Classifier. DTIC (ADA465712) (2003)
19. Shyu, M.L., Chen, S.C., Sarinnapakorn, K., Chang, L.: Principal component-based anomaly detection scheme (2006)
20. Tan, R., Badrinath Krishna, V., Yau, D.K., Kalbarczyk, Z.: Impact of integrity attacks on real-time pricing in smart grids. In: Proceedings of ACM CCS'13. pp. 439–450. ACM, New York, NY, USA (2013)
21. Vellaithurai, C., Srivastava, A., Zonouz, S., Berthier, R.: CPI INDEX : Cyber-Physical Vulnerability Assessment for Power-Grid Infrastructures 6(2) (2015)