# Device Fingerprinting in a Smart Grid CPS

Chuadhry Mujeeb Ahmed*
University of Strathclyde
mujeeb.ahmed@strath.ac.uk

Nandha Kumar Kandasamy*
Singapore Institute of Technology
NANDHA001@e.ntu.edu.sg

Darren Ng Wei Hong†
Singapore University of Technology and Design
darren_ng@alumni.sutd.edu.sg

Jianying Zhou*
Singapore University of Technology and Design
jianying_zhou@sutd.edu.sg

## ABSTRACT

Data integrity attacks on the various meter readings found in smart grid systems can be executed to be undetectable by current detection algorithms used in smart grid systems. These unobservable cyberattacks present a potentially dangerous threat to grid operations. Data integrity attacks that involve the compromise of various meter readings such as voltage and current levels can lead to threats ranging from trivial problems such as energy usage miscalculations to dire consequences resulting from the breakdown of the entire smart grid system through overloading the generators. An efficient detection algorithm to detect these attacks on various sensors embedded in the smart grid system is proposed. Due to manufacturing imperfections, discretizing the sensor readings produces variations in the readings that are unique to each sensor. A fingerprint of this sensor noise (variations in readings) is modeled through the use of machine learning techniques. Under a malicious spoofing attack, the noise pattern deviates from the fingerprinted pattern and hence enabling the proposed detection scheme to identify these attacks. A novel ensemble learning method is used to identify the Intelligent Electronic Device (IED). Experiments are performed on the Electric Power and Intelligent Control (EPIC) testbed. It is shown that a set of IEDs under the different stages of the power generation process can be uniquely identified with an accuracy greater than 90% based on the fingerprint.

**ACM Reference Format:**
Chuadhry Mujeeb Ahmed, Nandha Kumar Kandasamy, Darren Ng Wei Hong, and Jianying Zhou. 2020. Device Fingerprinting in a Smart Grid CPS. In *Proceedings of the 6th ACM Cyber-Physical System Security Workshop (CPSS '20), October 6, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3384941.3409588

## 1 INTRODUCTION

Cyber-Physical Systems (CPS) are integration of computing and networking elements with physical processes [8]. Embedded Computers and networks, monitor and control the physical processes through the use of Intelligent Electronic Device (IED) that computes the sensor readings. In particular we will consider examples of power generation systems in this work, a critical part in smart grid systems. A smart grid system consists of physical components such as the electric power generators, power substations, transmission and distribution lines, and cyber components such as the smart meters, IEDs, controllers, transmission control centers and Human Machine Interface (HMI) components that are interconnected via a communications network [2]. Advancements in micro-embedded systems and communication networks enabled existing physical systems to become digitized. The digital components also exposes the physical processes to malicious entities on the cyber domain [6]. Recent activities of cyber attacks and sabotages on these systems have raised security concerns on the reliability and security of smart grid systems.

Challenges in Smart Grid Systems and in CPS are fundamentally different from traditional IT systems. Real-time availability of the service also provides a stricter operational environment than most traditional IT systems and many CPS are legacy systems, that were designed without security in mind. Because Smart grid systems deal with the generation and supply of electricity, the manipulation through a cyber attack might result in damage to the physical property or the people who depend on these critical infrastructures, as was the case in Ukraine where the power supply was disrupted for the residents due to a cyber attack [7]. Data integrity is therefore an important security requirement for CPS. Sensor readings can be spoofed through the sniffing of packets between the communication of the IEDs and the HMI as classical man-in-the-middle attack [12]. Data integrity attacks on sensor measurements and their impacts and consequences have been studied largely in theory, including data manipulation injection, replay attacks and stealthy attacks. These previous studies proposed attack detection methods based on statistical fault detectors such as Cumulative Sum (CUSUM) or Chi-square, those can be deceived [11].

In this work, we propose an attack detection framework. The proposed detection scheme is a fingerprinting method to authenticate IEDs in smart grid systems when their readings are received at the HMI. This scheme is unique and provides a novel way in extracting the noise imperfections of the IEDs. The sensor noise is captured during the different stages of the power generation process. Process stages such as the Smart Home, Transmission and Micro-Grid readings have been tested and experimented upon. Because of sensor manufacturing imperfections, these IEDs can be differentiated from their noise pattern. These variations are minute in nature and are therefore hard to control or reproduce, making a spoofing of sensor reading of these noise profile challenging. A fingerprint is based on a set of time domain and frequency domain features that are extracted from the data collection of the IED readings. A multi-class support vector machine (SVM) is used to classify the noise patterns of various line readings found within each IED. Optimization in the decision function used and the use of different kernels, which

*This work was carried when authors were working at SUTD. Supervisory team was led by Prof. Jianying Zhou and assisted by Mujeeb and Nandha. Mujeeb and Nandha helped MS student Darren by collecting data from EPIC testbed under different scenarios, and help in writing this paper.
†Darren performed the analysis on the data from EPIC testbed as part of his MS thesis.

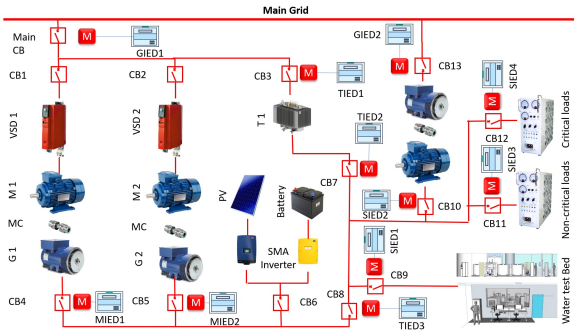


**Figure 1: Electrical layout of the EPIC (actual) test-bed. Electrical power lines are shown in red color lines. MC - Mechanical coupling, CB - Circuit Breaker, IEDs - intelligent electronic devices, the prefix G, M T and S stands for Generation, Micro-grid, Transmission and Smart Home respectively.**

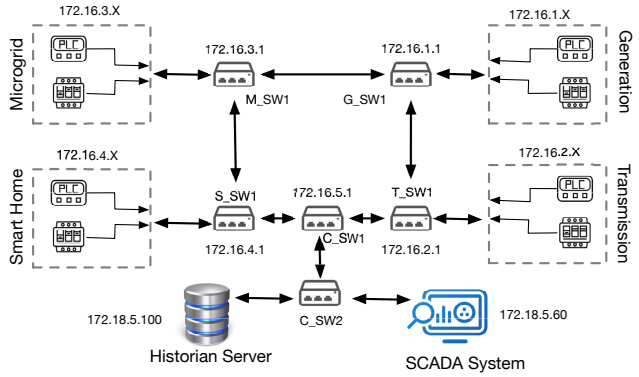


**Figure 2: A simplified network diagram for EPIC test-bed. Each dotted box represents a subnet with the respective IP addresses and corresponds to individual inner rings. A X in the IP address means that a device in that subnet would have the similar subnet mask and then unique X as its own IP. The connection $M_SW1 -- > S_SW1 -- > C_SW1 -- > T_SW1 -- > G_SW1 - - > M_SW1$ is the outer HSR ring**

is a set of mathematical functions used to transform the feature space of the dataset into higher orders for better separation of data have also been experimented on. Bagging algorithms which forms a class of algorithms which build several instances of a black-box estimator on random subsets of the original training set and then aggregate their individual predictions to form a final prediction. These methods are used as a way to reduce the variance of a base estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

A group of algorithms which includes Random Forest Classifier, Gradient Boosting Classifier and Ada Boost Classifier have been experimented on. These algorithms form the overall ensemble algorithm which takes in the output of each of the machine learning algorithms and use a majority vote to predict the final class labels. Such a classifier can be useful for a set of equally well performing model in order to balance out their individual weaknesses. IED and their respective lines identification accuracy is observed to be as high as 93%, and at least 90% for a range of sensors. The major contribution of this work includes the following: a) A novel fingerprinting framework that is built for the identification of IEDs based on sensor noise. b) A detailed evaluation of the proposed scheme as an IED identification mechanism, for a class of IED spoofing attacks. c) Extensive empirical performance evaluation on the EPIC testbed. A comparison of the performance of the various machine learning algorithms implemented is also presented. This work evaluates the device fingerprinting technique in the context of the smart grid power generation process.

## 2 BACKGROUND: EPIC TESTBED

This work is carried out on the EPIC (Electric Power and Intelligent Control) test-bed [1] which is a smart gird test-bed designed specifically for cyber security studies and technology evaluations. The test-bed is an industrial grade system capable of supplying power to a mini water treatment plant plant and a mini water distribution plant [4, 9], thus enabling studies such as cascading effect of cyber attacks, i.e., how a cyber attack on one critical infrastructure affects the reliability of other. As mentioned above, EPIC has been designed and implemented with industry grade equipment meeting all the standards and regulations of a smart grid CPS. Further, EPIC provides an emulated environment that cannot be observed in other test-beds, i.e., all the four sections of a typical power grid is available. The four sections are not only physically zoned into generation, micro-grid, transmission and smart home but also via network segregation for IEDs and Programmable logic controllers (PLCs). Advanced metering infrastructure (AMI) meters are integrated along with the IEDs to measure current, voltage, power and frequency at different electrical nodes. Each section of EPIC is described below,

The electrical layout of EPIC is shown in Figure. 1 and a basic description is given below. For further details on the electrical layout and physical process of EPIC, an interested reader is referred to the EPIC test-bed papers [1, 2].

- **Generation:** Due to the limitation on having fossil fuel powered prime-movers, the Generation stage is created using two induction motors that are powered by variable speed drives (15kW). The induction motors are mechanically coupled to respective 3-phase synchronous generators (10KVA each) that act as the master (reference grid) for the electrical system.
- **Micro-Grid:** Renewable energy component for electrical system is provided via solar Photo-voltaic (PV) array (34kW) and energy storage system (18kW) with inverters to harness solar energy and also include the behaviour of distributed energy resource dynamics into the system.
- **Transmission:** The transmission stage is sectioned into two groups, 1) a direct connection to downstream loads representing a pure micro-grid and 2) connection through a transformer (105kVA) to represent the line-impedance and tap-changing functionalities in transmission systems.

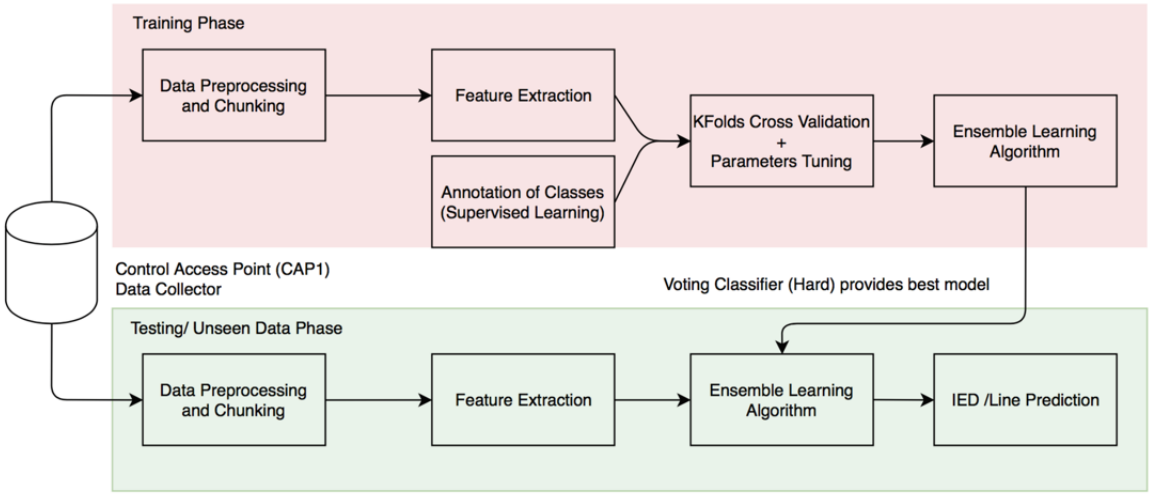


**Figure 3: Overview of the proposed technique.**

- **Smart Home:** The smart home acts as the power distribution system's load, the load is configurable using programmable load banks (45kVA) containing RLC[1] loads and a motor (10kW) load for representing pumps, ventilation systems etc. Besides, as mentioned before two water test-beds are connected to the EPIC test-bed to form a cluster of critical infrastructure.

Similar to the physical components, the network of epic is also sectioned into four groups each representing the respective control zones, namely, power generation, transmission, micro-grid, and smart home. The IEDs connected to the individual nodes are connected using high speed seamless redundancy (HSR) rings which ensures $n-1$ redundancy. The HSR rings connecting the IEDs are referred as inner rings, and each inner ring has a PLC for controlling the process of the respective section. The IEDs and PLCs are named with suffix of the corresponding section, for example an IED in generator section is named as GIEDx (x is numerical), whereas the IED in smart home section is named as SIEDx. All the inner rings are connected to each other via a another HSR ring referred as outer ring, the outer ring enables the communication between the inner rings. $n-1$ redundancy is ensured for the outer ring as well. The Supervisory Control and Data Acquisition (SCADA) system is connected to the outer ring. Figure 2 shows the communication network architecture in EPIC test-bed. The data flow from IED SIED1 will start at the physical measurement "M" shown in Figure 1, the measure data is then converted in IEC61850 network data by SIED1 which is then made available to SPLC and the SCADA system via IEC61850 MMS (Manufacturing Message Specification) variables. IEC61850 GOOSE (Generic Object Oriented Substation Event) variables are used only for Transmission section and only made available among the IEDs.

[1]Resistor-Inductor-Capacitor

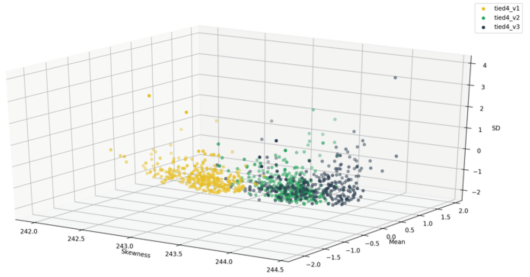

**Figure 4: Scatter plot of TIED v1, v2 and v3 the addition of skewness as the third dimension shows separability in the dataset. The results of these experiments conclude that the noise patterns of these IEDs follow a unique variation.**

## 3 OVERVIEW OF THE PROPOSED TECHNIQUE

Overview of the proposed technique is shown in Figure 3. The data is collected using the log management tool in the testbed. We have studied the system design and functionality of the power generation process of EPIC testbed and identified 3 different key stages and IEDs to focus the experiments on. We collected the data under regular operation (no attack) with the predefined loads. As shown in Figure 3, we will discuss the process of collecting the dataset from the historian HMI interface. Then, we will present the data preprocessing and chunking process followed by the fingerprinting scheme [3, 5] which involves the use of various supervised machine learning algorithms embedded into a single ensemble learning cluster where each of the individual machine learning algorithm learners are trained to solve the same problem and a voting classifier is used [10].

IED Noise: The IEDs send the readings of the sensors through the use of the access point to the control unit. Here, $X_k$ (refers to the set of sensor readings of each of the IEDs at time $k$. $X_k = \{x_k^1, x_k^2, ..., x_k^z\}$

where $z$ corresponds to the total number of IEDs found in the EPIC testbed system. Due to manufacturing imperfections of the IED components, the IED measurements of the various system variables such as voltage and current contains noise. It has been observed that when the intended value of the system variable is set, the readings are always displaced by a small margin of up to 2 decimal places. For example, when the voltage value is set to 240V, based on more than 5000 different data points observed, the values received by the historian logging system shows that the reported value vary around the true value, indicating the presence of noise found in these IEDs. This observation is seen similarly for each of the IEDs and their individual line readings.

The presence of these noise patterns indicates a possibility that these fluctuations can be individually identified for each of the IEDs. The data collected is preprocessed to remove any possible loss of data during the data collection stage due to unforeseen technical issues and are processed into chunks of size $m$. These chunks are then analyzed and statistical features such as mean and standard deviation are extracted. Both frequency domain and time related features are examined. The noise and its time/frequency domain features exhibit clustering characteristics indicating that the features extracted from the data points belonging to the same IEDs tend to have small distances to data points of the same IEDs thus forming dense areas of clustered points throughout the visualization. Furthermore, these features extracted from the noise are profiled using standard deviation, mean and skewness to show further possible correlation. In Figure 4, we can see that there is a distinct separation between TIED4 v1 from TIED4 v2 and v3. Although the separation is distinctively clearer, it is important to note that the noise pattern found in the dataset from v3 is very similar to that of v2 thus, identifying v3 from v2 might not provide desirable results.

Next, a set of machine learning algorithms are used to classify the IEDs from one another. Noise fingerprint can be generated over time as the process of power generation is in progress. During normal operations, the fingerprint will be used to identify if the data reading received by the Control Access Point (CAP) is indeed from the IEDs or if it is injected by the attacker.

**Data Preprocessing:** Data is collected from the IEDs right at the moment EPIC is started to the end. After the collection of data is done, we would need to do certain data preprocessing to prepare the dataset for both feature extraction and then classification. We first remove all of the intermediate 0 values as we are only concerned with values of the predefined range. For voltage, we have decided that the predefined value would be at 240V. The dataset is then compiled to their respective IEDs such as TIED4, MIED1, MIED2 and SIED4. In total, there were 3,680 raw data points collected.

**Data Chunking**: After the collection of data is done, we would need to create chunks of dataset. Because the features are statistical in nature, we would need to create chunks of data to extract these features. Chunking allows us to understand the right size of data needed to capture the variance in the set and how much data is needed to train the machine learning algorithm to produce desirable results. A list of varying chunk sizes was tested upon such as (5,10,2,40) and based on the accuracy performance, chunk size 20 produced the best results. Figure 5 shows the accuracy results of the chunk size on the support vector machine algorithm with the only variable difference as the chunk size, $m$. We could see that

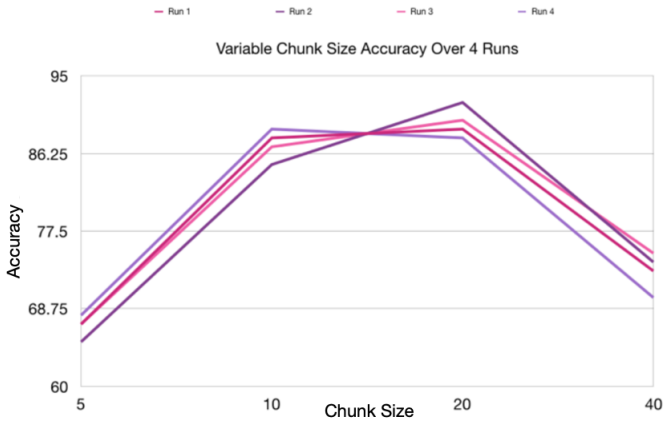


**Figure 5: Accuracy percentage results of varying chunk sizes.**

chunk size 5 produces a fairly low accuracy performance as 5 data points does not capture a lot of the variance in the data set thus each chunk produced did not tend to follow a similar pattern. On the other spectrum, chunk size 40 captures most of the variance in the data set but because the data collected is limited, using a chunk size of 40 meant that the overall number of fingerprint data points to classify is greatly reduced. Hence, the machine learning algorithm is not able to produce a the desired accuracy count.

**Table 1: List of features used.**

| Feature | Description |
|---|---|
| Mean | $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ |
| Std-Dev | $\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i)^2}$ |
| Mean Avg. Dev | $D_{\bar{x}} = \frac{1}{N}\sum_{i=1}^{N}\lvert x_i - \bar{x}\rvert$ |
| Skewness | $\gamma = \frac{1}{N}\sum_{i=1}^{N}(\frac{x_i-\bar{x}}{\sigma})^3$ |
| Kurtosis | $\beta = \frac{1}{N}\sum_{i=1}^{N}(\frac{x_i-\bar{x}}{\sigma})^4 - 3$ |
| Spec. Std-Dev | $\sigma_s = \sqrt{\frac{\sum_{i=1}^{N}(y_f(i)^2)*y_m(i)}{\sum_{i=1}^{N} y_m(i)}}$ |
| Spec. Centroid | $C_s = \frac{\sum_{i=1}^{N}(y_f(i))*y_m(i)}{\sum_{i=1}^{N} y_m(i)}$ |

Vector $x$ is time domain data from the sensor for $N$ elements in the data chunk. Vector $y$ is the frequency domain feature of sensor data. $y_f$ is the vector of bin frequencies and $y_m$ is the magnitude of the frequency coefficients.

**Feature Extraction:** As shown in Table 1, there are seven different features used to construct the fingerprint. The Fast Fourier Transform (FTT) algorithm is an algorithm that samples a time series data over a defined period and separates it into its corresponding frequency components with the purpose to extract its spectral features. The fingerprinted data points from the chunking is then labelled according to their respective IEDs for supervised learning.

**Cross-Validation:** Cross-validation is a type of model validation technique used in the assessment of how the performance

results of a given statistical analysis will generalize to an unseen data set. It is a resampling procedure used to evaluate the algorithm model on unseen data, i.e., how the model is expected to perform in general when used to make classifications on data not used during the training phase of the model. It takes in a single parameter k, which is the number of sub-divisions given to the overall dataset. For K different groups, take the first group as the hold out or test data set and combine the remaining groups as the training data set. The machine learning model is fitted, trained and evaluated based on the hold out set. The process is repeated for different hold out sets and their overall model evaluation score is returned. K-Folds cross-validation is implemented with the use of the scikit-learn python module.

## 4 THREAT MODEL

In this paper, we will only consider specific cyber-attacks on the data integrity of these IED readings and not physical attacks due to the nature of the smart grid system and the EPIC testbed system, direct contact to these sensors is very dangerous and not easily accessible to both an intruder or even an insider (operator) who has direct access to these physical IEDs. First, we profile the attacker and lay down assumptions of the attacker backed with justification. Following on, we will introduce one such possible cyber-attack on the EPIC testbed system. Attack here is defined as a sequence of one or more malicious actions intended to move a CPS such as EPIC system to an undesirable state while the attacker refers to an individual, or a group, that intends to, or has launched attacks on a CPS. The attacker model is a formal description of the capabilities and knowledge base of an attacker and provides useful information when designing protective and detective measure, and for comparing the chances of successfully launching an attack. The attacker may have its motivation stemming from a set of intentions such as denial of service, falsifying data and the actual usage of the power supply, diverting resources, performance degradation or even damaging a component.

### 4.1 Attacker Model

Assumptions on the Attacker: It is assumed that the attacker is an Insider, an employee who has information access to the system network and has knowledge on the operational process of the EPIC testbed or smart grid system. On top of that, the attacker has knowledge on all of the different IEDs present in each of the 4 stages of the power generation process. The adversary has perfect knowledge of the various sensor readings and can modify them arbitrary. An example would be the following, the adversary would like to reduce his apparent consumption of electricity and thus would inject falsified readings of current and voltages to make it appear as if he was not using the actual amount of electricity. The goal is to minimize the apparent consumption without being detected.

The attacker is a strong adversary who is able to launch cyber-attacks both within the premise and out of the premise of the system. One of the key vulnerabilities of the smart grid system is its communication link between the IEDs and the HMI. As such, an attacker can compromise these communication links simply through the use of a Man-in-The-Middle (MiTM) attack. This could also lead to a data integrity attack against one or more IEDs take place when legitimate data of the IED is tampered with, replaced or deleted before its transmission to the data concentrate unit is completed successfully. For example, by interjecting the communication link between the access points to the control unit, the attacker can sniff the packets sent over the communication link and decide to either drop the packet, delay the sending of the packet or reload it with a different payload. Therefore, we will need to find a robust method to authenticate the IED readings received on the end of the control unit and to the Historian logging system. While a malicious insider can break into the network communication between the IEDs and the control unit, an outsider may also be able to break into the network if appropriate firewall is not installed. Furthermore, we also assume that the attacker would want to be stealthy and undetected, hence placing this attack under the unobservable attack category.

### 4.2 Attack Model

Data Injection Attacks: This class of attacks refers to one where the attacker intercepts the communication link, sniffs the packet sent and injects or modifies the actual IED sensor readings with a falsified one that is generated randomly or with a specific intend. The adversary may inject fictitious data into the HMI to either portray increased electricity consumption, or to reduced it. It affects the normal operations of the system as power load balances might be redistributed to other areas as a result of this data injection. For example, manipulating the current and voltage readings can lead to a misleading reading of power since $P = IV$. Thus, the system state is disrupted, and power will be redistributed and generated higher than the source actually needs. This is a direct attack on the integrity of the system. The proposed solution uses a fingerprinting methodology to identify that the readings are indeed coming from the IEDs and not from an attacker. This is made possible due to the intricate nature of the imperfections. Because the current readings and voltage readings have been fingerprinted, changing their values (i.e. if the operating voltage is set to be at 240V but the spoofed reading shows 220V), the operator might increase the voltage by 20V and when it has been increased to 260V, the attacker can send packets showing the desired 240V but the actual voltage is now at 260V. Similarly, this scenario works for Current as well. Because $P = IV$, an increase in either of Current or Voltage will lead to an increase in power supply. Over or under powering over prolonged period can also be used for fault injection. Under powering can increase signal propagation delay and can lead to setup time violations in hardware platforms. In our experiments, we consider three types of data injection attacks.

**Bias Data Injection Attack**: The goal of the bias data injection attack is to deceive the controller by sending an increased/decreased value of the actual reading by a constant $c$, so as to deceive the controller into thinking that the received values are the true IED sensor readings. For example, in this situation, the voltage measurements are increased while the actual voltage level is invariant. Hence, the controller will continue to reduce the voltage supplied until it reaches zero. The attack vector is defined as follows:

$$y'(k) = \hat{y}(k) \pm c \tag{1}$$

Where $c$ is a constant added at each time instant. Negative constant could be used for attacks such as the one targeting damage by

deceiving the controller to increase the current to unacceptable levels.

**Geometric Data Injection Attack**: Geometric attack is similar to bias data attack but consist of two additional coefficients $\alpha$ and $\beta$. The constant is now modified to increase exponentially instead of constantly. The attack vector is defined as follows:

$$y'(k) = \hat{y}(k) \pm \beta\alpha^{n-k} \tag{2}$$

Where $\alpha \in (0,1)$ and $\beta$ is a multiplier to be adjusted for maximum damage. $n$ here represents the last measurement received in the sequence and $k$ is the measurement number starting from 0.

**Zero-Alarm Attack**: zero-alarm attack is crafted such as to remain undetectable by traditional statistical methods like cumulative sum (CUSUM). The CUSUM detection mechanism is fed with the sequential data of the incoming IED readings. It computes the current cumulative sum and does so by performing a change point detection. Because CUSUM depends largely on two predefined constant values, the threshold, $\tau$ and the bias,$b$, the adversary can easily remain undetected by choosing the attack vector $\delta_k$ that stays within the confines of the threshold value. The impact and limitations of these statistical methods as detection mechanisms have already been widely covered in literature.

It is important to note that for such zero-alarm attacks, if the attacker wants to remain undetected, he cannot damage the system but can always still impact the integrity of the system.

### 4.3 Attack Execution

Because of the nature of the EPIC testbed system, an actual MITM attack was not possible to carry out. This is due to the restrictions of the laboratory as attacks can lead to dangerous scenarios. A successful attempt at data injection attack might bring dire consequences. In reality however, the adversary would need to intercept the data traffic between the communication lines of the IED's access point to the controller's access point. The attack could be physical or logical MiTM, in which a physical MiTM is physically intrusive but has no impact on the configuration of the plant and the logical MiTM does not need physical intrusion but has serious impact on the configurations of the plant as the restoration is likely to consume 4-5 working days. Packets are then inspected and modified (payload changed depending on type of attack) but however, modified data readings are injected into the sequence of legitimate data points found in the historian logging system. The above attack is equivalent to a malware that modifies the configurations in the SCADA software to assign the IED's IEC61850 server (one under attack) to a local port and feeds the modified data by polling the actual IED and creating a malicious server on the above local port. From there, the CUSUM and the proposed detection mechanism were tested for accuracy and performance.

### 4.4 Generating Attack Data

As mentioned above, the simulated attack instead focuses on generating the attack data which will be used by the adversary to inject into the controller unit. A script has been written to read an incoming packet payload and modify it according to whether it is a bias, geometric or zero-alarm attack. For the zero-alarm attack, to remain undetected, the function created to generate the attack falsified data is a random number generator that generates a random floating number based on the current reading received and does so in the ranges of a predefined standard deviation. This allows the random floating numbers to hover around a predefined value, but the changes are within the threshold value set for CUSUM. Thus, allowing the adversary to stay undetected while still being able to manipulate the data readings.

## 5 PERFORMANCE EVALUATION

In this section, a brief background of the statistical detection scheme, CUSUM will be provided, followed by the performance of the machine learning algorithms in identifying each of the IEDs, the model evaluation methods used as well as the evaluation of the proposed detection mechanism scheme.

### 5.1 CUSUM

Statistical detectors estimate state values in each turn of the sequence and compares it with the IED measurement reading. The difference between the two values provides a value that stays within the threshold value under normal operation. As such, the threshold value and the bias value are important key variables as they affect the false alarm rate. To begin, we have identified two hypotheses to be tested, $\mathcal{H}_0$ which represents the hypothesis where a given measurement depicts normal process behavior and $\mathcal{H}_1$, which represents the hypothesis where a given measurement depicts anomalous process behavior (with attack).

---

**CUSUM:** $S_{0,i} = 0, \quad i \in \mathcal{I}$,

$$\begin{cases} S_{k,i} = \max(0, S_{k-1,i} + z_{k,i} - b_i), & \text{if } S_{k-1,i} \leq \tau_i, \\ S_{k,i} = 0 \text{ and } \tilde{k}_i = k-1, & \text{if } S_{k-1,i} > \tau_i. \end{cases} \tag{3}$$

**Design parameters:** bias $b_i > 0$ and threshold $\tau_i > 0$.
**Output:** alarm time(s) $\tilde{k}_i$.

---

We have to choose both threshold $\tau_i$ and bias $b_i$ such that both are greater than 0.

### 5.2 Model Evaluation Methods

To prevent cases of over-fitting or under-fitting of the dataset, the technique used to separate the dataset into $S_{train}$ and $S_{test}$ is crucial. Therefore, Cross validation is used as a means of model evaluation method. Cross validation belongs to a class of model evaluation method that has significant performance over the use of residuals. The limitations with residual evaluations include the fact that they do not provide an indication of how well the learner will do when it has to make predictions for unseen data (independent data). A simple solution to eliminate this problem is to prevent the use of the entire data set as a whole when training the classifier. A segment of the data is removed before the training process. After which when the training phase has been completed, the data that was removed earlier can now be used to test the performance of the classification model. This forms the basis for a whole class of model evaluation methods called cross validation.

**Hold Out Method:** The simplest form of cross validation, the data set is first separated into two distinct sets, $S_{train}$ and $S_{test}$. Next, the model is trained only using the training set that was separated and following which the model is then used to predict the classes or labels of the data in the testing set. Hence, the testing set is used to estimate the prediction error rate of the trained classifier algorithm. However, from the experimentation, it is shown that when the dataset is sparse or minute, we may not be able to set aside a portion of the dataset for testing. Therefore, in the initial phase of the project, the Hold out method was removed in favor of the k-fold cross-validation method with three-way data splits so that the parameters of the algorithm can be tuned.

**K-Fold Cross-validation:** Is an improvement over the standard holdout method. The algorithm works as such, the dataset is divided into $k$ subsets, and the holdout method is repeated $k$ different times with each time, a sliding window of fixed value (subset) is used as a test set while the rest $k-1$ subsets are used as training sets. Even though this method takes $k$ times as much computation as compared to the holdout method, the variance of the resulting estimate is reduced as $k$ is increased, thus resulting in better model validation.

**Choosing** $k$: The number of $k$ folds can affect the variance of the resulting estimate. With a large $k$ value, the bias of the true rate estimator will be small, hence it will be more accurate. However, the variance of the true error rate will be large and also the computational time as it has to run $k$ number of times. With a small $k$ value, the number of experiments and computation time are reduced, with a small variance of the estimator and the bias of the estimator will be large.

Using the first machine learning algorithm, SVM, K-Folds was running repeatedly with a range of $K \in \{1, 10\}$ and found that the most suitable number of folds based on accuracy against computational time effort is 4. As such, the value of $k$ will be set at 4 when comparing amongst the algorithm.

## 5.3 Zero-Alarm Attack Design

A zero-alarm attack is designed in such a way that it stays undetected by the CUSUM detectors. As shown in the CUSUM procedure, we can write (3) in terms of the estimation error $e_k$:

$$S_{k,i} = \max(0, S_{k-1,i} + |C_i e_k + \eta_{k,i} + \delta_{k,i}| - b_i), \tag{4}$$

if $S_{k-1,i} \leq \tau_i$; and $S_{k,i} = 0$, if $S_{k-1,i} > \tau_i$.

Consider the attack:

$$\delta_{k,i} = \begin{cases} \tau_i + b_i - C_i e_k - \eta_{k,i} - S_{k-1,i}, & k = k^*, \\ b_i - C_i e_k - \eta_{k,i}, & k > k^*. \end{cases} \tag{5}$$

For all given $k \geq k*$, zero alarm has been raised. The assumption made here is that the adversary knows exactly $S_{k-1,i}$ , the value of the CUSUM sequence one timestamp before inducing the attack. This would allow him to set the falsified data such that it will not trigger the alarm of the CUSUM scheme. The IED readings received by the controller will thus take the following form:

$$\bar{y}_{k,i} = \begin{cases} C_i \hat{x}_{k,i} + \tau_i + b_i + C_i \hat{x}_k - \eta_{k,i} - S_{k-1,i}, & k = k^*, \\ b_i + C_i \hat{x}_{k,i}, & k > k^*. \end{cases} \tag{6}$$

| IED | Line 1 | Line 2 | Line 3 |
|---|---|---|---|
| TIED4.I | 93.12% | 89.12% | 92.10% |
| TIED4.V | 91.15% | 92.15% | 78.89% |
| MIED1.V | 93.34% | 91.23% | 93.14% |
| MIED1.I | 88.12% | 83.45% | 87.12% |
| MIED2.V | 89.12% | 82.45% | 93.14% |
| MIED2.I | 86.34% | 87.45% | 88.34% |
| SIED4.V | 91.23% | 93.14% | 76.54% |
| SIED4.I | 89.45% | 88.34% | 85.23% |

**Table 2: Multiclass identification between each Line measurement reading for each IEDs.**

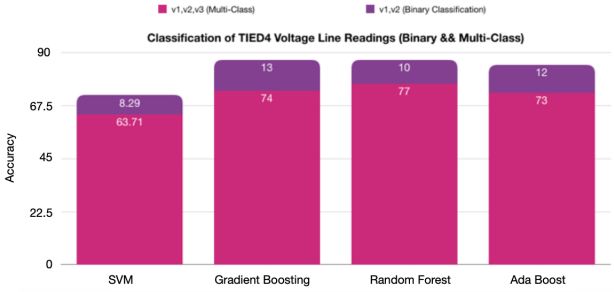


**Figure 6: Machine Learning Classifier comparison when used to identify IED v1, v2, v3 from one another in a multi-class problem.**

## 5.4 Performance Metrics

The experiments were carried out for each of the IEDs and their respective line values found within the EPIC testbed system. A binary classification model is used to identify if the measurement received by the controller unit is indeed from the IED access point transmission (normal) or is malicious (attack). Let $I$ be the total number of IEDs. We define $TP_i$ as the true positive for IED $i$ when it correctly classifies the IED based on ground truth while the false positive is defined as $FP_i$. Similarly, we take the false negative as $FN_i$ and is defined as the wrongly rejected classification while $TN_i$ is the rightly rejected class. Therefore, the overall accuracy for each of the IEDs in $I$ can be defined as the following:

$$acc = \frac{\sum_{i=1}^{c} TP_i + \sum_{i=1}^{c} TN_i}{\sum_{i=1}^{c} TP_i + \sum_{i=1}^{c} TN_i + \sum_{i=1}^{c} FP_i + \sum_{i=1}^{c} FN_i}. \tag{7}$$

## 5.5 IED Identification Accuracy

In Table 2, the IED identification accuracies were given for 24 different IEDs. The IEDs belong to 3 different processes found within the power generation process. We can see that the lowest identification accuracy was 76.54% and this is due to the fact that SIED4.V3 has very similar noise pattern to both V1 and V2 thus the identification accuracy is a lot lower. This is however, not a worrying factor as the noise pattern of the IEDs are very difficult for the adversary to mimic. On average, the IED identification hovers around a high 90% range for all 24 IEDs. The results shown in Table 2 is the average

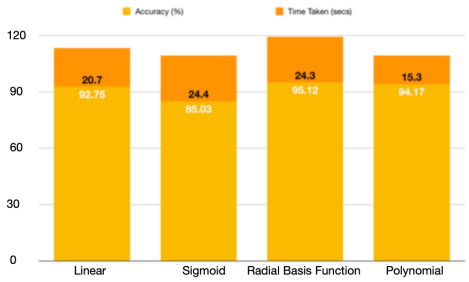


**Figure 7: Graph showing the average performance of each kernel function.**

of 100 different runs from the ensemble algorithm. In the case of identifying the sensors from one another, a multi-class classification model is used.

## 5.6 Different Machine Learning Algorithm Performance

In Figure 6, we can see that the performance of the ensemble algorithm class is at least on average 10% better than the support vector machines. This is due to the fact that the identification of IEDs amongst each other is not linearly separable. As such, the ensemble algorithm which consist of the gradient boosting algorithm, the adaptive boosting algorithm and the Random Forest algorithm performs better. The ensemble algorithm uses the voting classifier concept where the main idea is to combine conceptually different machine learning algorithms and use a majority vote or the average of the predicted probabilities through the use of soft weighted voting to predict the class labels. This is especially useful for a set of equally well performing models as it balances out each of their individual weakness. In the case of soft voting, when the weights are provided, the predicted class probabilities for each of the classifiers in the set are collected and weighted accordingly, and finally averaged. The deciding class label is hence derived from the class with the highest probability count. The grid search optimizer is also used for the voting classifier to tune the hyper-parameters of the individual estimators. Because of the performance and validity of the ensemble algorithm over the support vector machine, the ensemble algorithm is used as the machine learning classifier to identify the IEDs. The ensemble algorithm was also used as a comparison against the statistical detector, CUSUM. These results show and prove that the noise-based device fingerprint through the use of the ensemble learning algorithm provides a very high accuracy in prediction against malicious data.

Four different kernels (Sigmoid, Linear, Radial Basis Kernel Function (RBF) and Polynomial) for SVM were used in the experiments experiments. These were tested on 5 separate runs using k-folds cross-validation and the results are averaged and compared. The Figure 7 shows the average accuracy performance for each of the different kernels in the 5 separate runs. From the experimentation, we can conclude that the top 2 performing kernels are the RBF and polynomial kernels. RBF, however took considerably more computation time than polynomial. These 2 kernels are used against a grid-search parameter estimation to improve the algorithm's performance. The polynomial kernel yields an accuracy rate similar to that of RBF but computes 21% faster, the kernel polynomial with grid-search parameter estimation will be used for the final comparison against the other algorithms.

## 5.7 Attack Detection Performance

The experiments were carried out for each of the IEDs and their respective line values found within the EPIC testbed system. A binary classification model is used to identify if the measurement received by the controller unit is indeed from the IED access point transmission or if it is modified data coming from the adversary.

**Threshold and Bias Selection:** As mentioned in the CUSUM section that the threshold and bias should be selected such that the false alarm rate is not too high but also not too low such that it is not able to detect any form of attack. The values of the threshold and bias has been tested, and the final selected values are 3.5 and 2. This in-turn leads to a very low false alarm rate while still being functional to detect if there are any anomaly present. Figure 8 shows the IED measurements of TIED4 current L1 over time. It can be seen that over a span of 2500 seconds, there are only three false alarms. Further analysis shows that these false alarms corresponds to when the process is completed, and the current is reducing to 0. If the threshold and bias is tweaked to remove these false alarms, CUSUM will not be efficient to detect against max-min attacks, where the adversary would set the measurement readings to either 0 (min) or extremely high (max) in an attempt to cause disruption to the system.

**Constant Bias Attack:** Figure 9 shows that during the transmission state, the malicious data was injected when $k = 21s$ (21 seconds since the start of the transmission process in EPIC testbed). The bias attack used was $\delta_1 = 2$. CUSUM was able to detect the attack immediately. The proposed mechanism was able to detect the attack as well but because it uses a chunk size of 20, it has to wait for 20 seconds before it is able to detect it.

**Geometric Attack:** Similar to the constant bias attack, both CUSUM and the proposed mechanism were able to detect the attack. The attack was also launched at $k = 21s$ (21 seconds since the start of the transmission process in EPIC testbed).

**Zero-Alarm Attack:** Figure 10 shows the zero-alarm attack when it was launched at $k = 21s$ (21 seconds since the start of the transmission process in EPIC testbed). Because the attack was designed such that the no alarms were raised, CUSUM was not able to detect when the attack was launched. Since the adversary has complete knowledge of the system including the CUSUM detector, he can deliberately set and launch the attack such that CUSUM detector would not be able to detect it. Figure 10 shows the measurement readings received during the experiment. It can be seen from the graph plots that the adversary spoofed the measurements in a way that it stays within the confines of the threshold and bias value, thus remaining undetected while reducing the values of the measurement to near zero. On the other hand, because the spoofed values do not follow the intrinsic noise pattern fingerprinted for each individual IEDs, the spoofed data does not match the pattern fingerprinted using the proposed mechanism. Our proposed technique here removes the limitation of CUSUM detectors as it

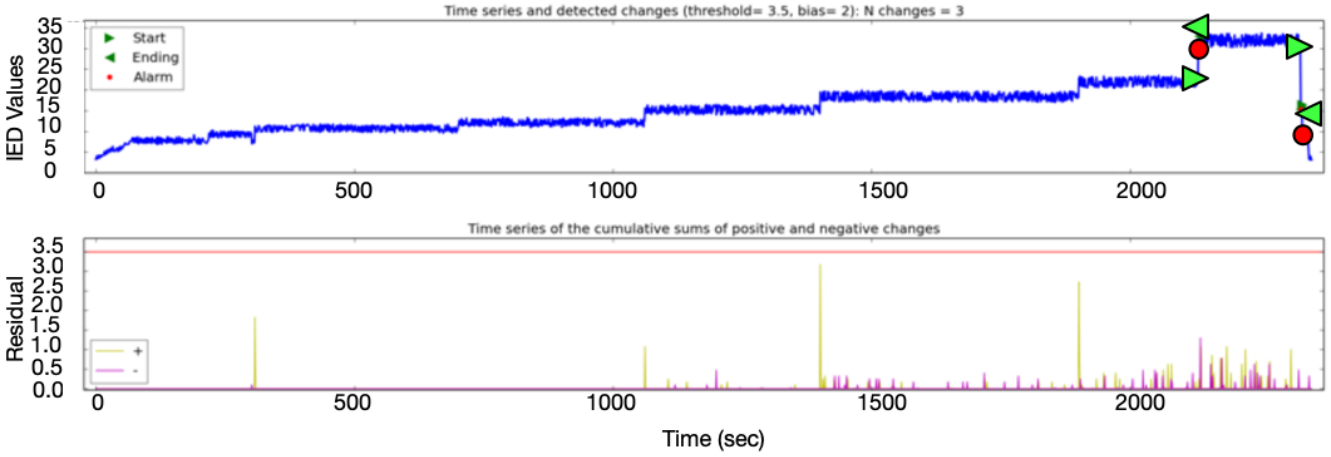


**Figure 8: CUSUM threshold and bias setting with 3 false alarm over 2500 seconds.**

was able to detect the attack as the noise pattern coming from the crafted attack does not match the one fingerprinted for the IEDs during training phase.

## 6 CONCLUSIONS

A novel method to fingerprint the noise patterns present in the IEDs of the smart grid system is presented. From the experiments, it is shown that the noise pattern of each IED can be fingerprinted uniquely and thus can be identified individually with high confidence. With the extraction of both time and frequency domains, these key feature attributes were passed into machine learning classifiers for training purposes. Four different machine learning classifiers were tested and experimented on. With the best method as the ensemble learning algorithm comprising of adaptive boosting, gradient boosting and random forest with grid search optimization combined together through the use of a voting classifier mechanism where the class label is decided based on the highest probabilities. A binary classification is used to detect malicious data from actual data that is received from the IEDs as opposed to one from the adversary. Our results have shown that the proposed mechanism eliminates the limitations of statistical detectors such as CUSUM and was able to detect zero-alarm attacks.

## REFERENCES


[1] Sridhar Adepu, Nandha Kumar Kandasamy, and Aditya Mathur. 2019. EPIC: An Electric Power Testbed for Research and Training in Cyber Physical Systems Security. In *Computer Security*, Sokratis K. Katsikas, Frédéric Cuppens, Nora Cuppens, Costas Lambrinoudakis, Annie Antón, Stefanos Gritzalis, John Mylopoulos, and Christos Kalloniatis (Eds.). Springer International Publishing, Cham, 37–52.

[2] Chuadhry Mujeeb Ahmed and Nandha Kumar Kandasamy. 2021. A Comprehensive Dataset from a Smart Grid Testbed for Machine Learning Based CPS Security Research. In *Cyber-Physical Security for Critical Infrastructures Protection: First International Workshop, CPS4CIP 2020, Guildford, UK, September 18, 2020, Revised Selected Papers 1*. Springer, 123–135.

[3] Chuadhry Mujeeb Ahmed, Aditya P. Mathur, and Martín Ochoa. 2020. NoiSense Print: Detecting Data Integrity Attacks on Sensor Measurements Using Hardware-Based Fingerprints. *ACM Trans. Priv. Secur.* 24, 1, Article 2 (sep 2020), 35 pages. https://doi.org/10.1145/3410447

[4] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. 2017. WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks* (Pittsburgh, Pennsylvania) *(CySWATER '17)*. ACM, New York, NY, USA, 25–28. https://doi.org/10.1145/3055366.3055375

[5] Chuadhry Mujeeb Ahmed, Jianying Zhou, and Aditya P. Mathur. 2018. Noise Matters: Using Sensor and Process Noise Fingerprint to Detect Stealthy Cyber Attacks and Authenticate Sensors in CPS. In *Proceedings of the 34th Annual Computer Security Applications Conference* (San Juan, PR, USA) *(ACSAC '18)*. ACM, New York, NY, USA, 566–581. https://doi.org/10.1145/3274694.3274748

[6] Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, and Shankar Sastry. 2009. Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security*. 5.

[7] Defense Use Case. 2016. Analysis of the Cyber Attack on the Ukrainian Power Grid. (2016).

[8] E. A. Lee. 2008. Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. 363–369. https://doi.org/10.1109/ISORC.2008.25

[9] A. P. Mathur and N. O. Tippenhauer. 2016. SWaT: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. 31–36. https://doi.org/10.1109/CySWater.2016.7469060

[10] Gauthama Raman MR, Chuadhry Mujeeb Ahmed, and Aditya Mathur. 2021. Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation. *Cybersecurity* 4, 1 (2021), 1–12.

[11] C. Murguia and J. Ruths. 2016. Characterization of a CUSUM model-based sensor attack detector. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. 1303–1309. https://doi.org/10.1109/CDC.2016.7798446

[12] David I Urbina, Jairo A Giraldo, Alvaro A Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. 2016. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1092–1105.

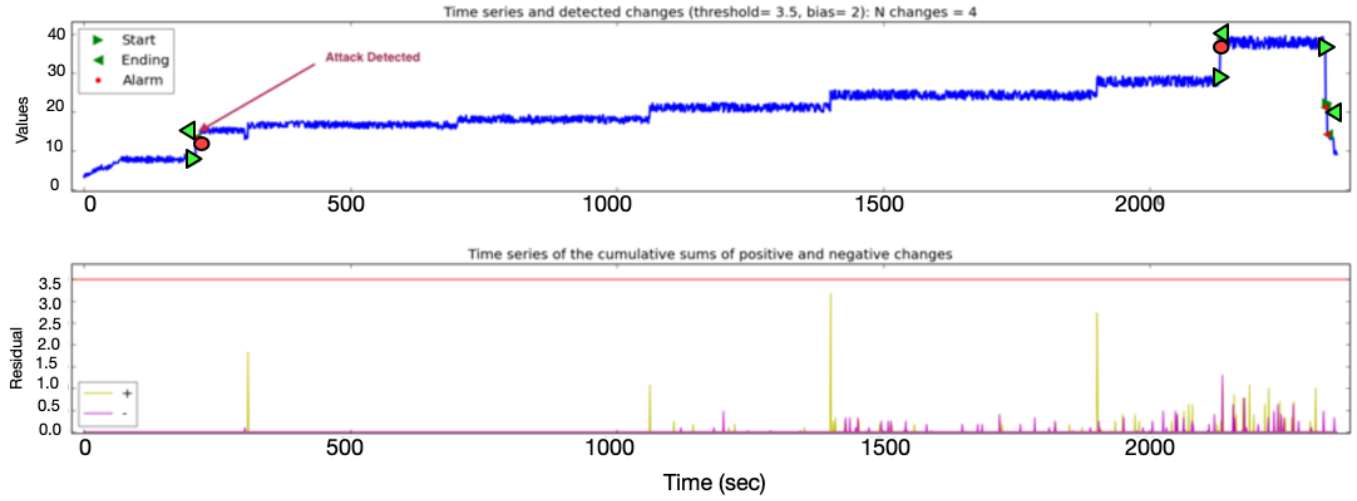


**Figure 9: Constant Bias Attack with CUSUM detector. Attack Detection was made with the Alarm marked in the diagram.**

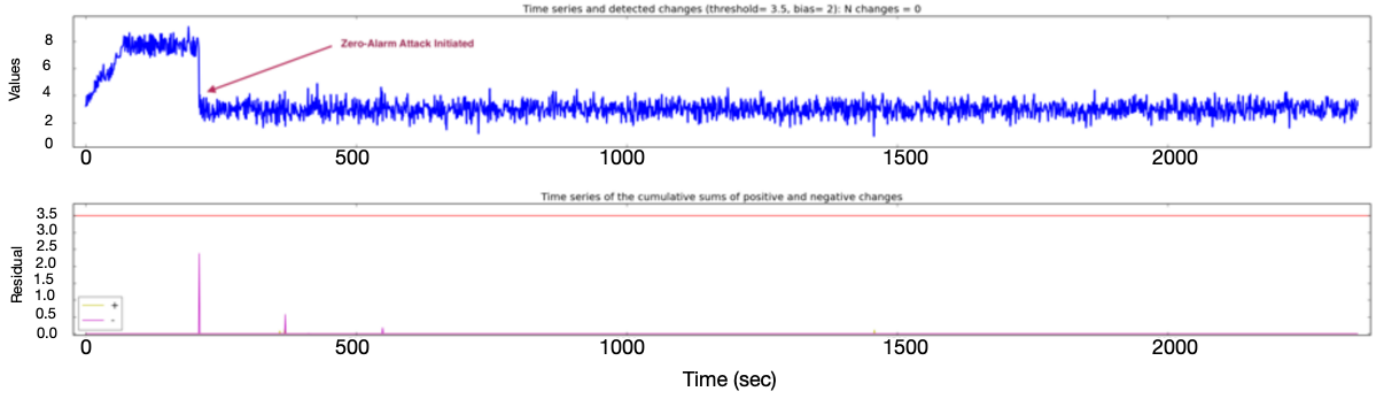


**Figure 10: Zero-Alarm Attack with CUSUM detector. Attack Detection was not made, CUSUM not able to detect the attack.**