Intrusion Detection for Cybersecurity of Smart Meters

Chih-Che Sun, *Member, IEEE,* D. Jonathan Sebastian, *Student Member, IEEE,* Adam Hahn, *Member, IEEE,* and Chen-Ching Liu, *Life Fellow, IEEE*

***Abstract*—The integration of Information and Communications Technology (ICT) enables real-time communication for smart meters to participate in power system operations. However, Advanced Metering Infrastructures (AMI) are vulnerable to cyber attacks. Both utilities and power consumers may become victims of cyber intrusions. In this paper, a two-stage cyber intrusion protection system is proposed. At the first stage of intrusion detection, a Support Vector Machine (SVM) is used as a detection algorithm to discover suspicious behaviors inside a smart meter. At the second stage, the Temporal Failure Propagation Graph (TFPG) technique is used to generate attack routes for identifying attack events. Finally, the proposed pattern recognition algorithm is used to calculate the similarity between a detected abnormal event and pre-defined cyber attacks. A higher similarity value implies a higher chance that a smart meter is under attack. An AMI security test platform has been developed to: (1) Collect training/testing data for SVM, (2) Simulate and analyze cyber attack events, and (3) Validate the proposed cyber attack protection system. The test platform consists of Network-Simulator 3 (NS-3) software to simulate an AMI network environment and single board computers (SBCs) to emulate the IEEE 802.15.4 communication between a grid router and a smart meter.**

***Index Terms*—Advanced metering infrastructure (AMI), smart meters, cyber-physical system security, intrusion detection.**

I. Introduction

SMART grid technologies have been deployed to enable the new functions and services, improving the reliability, security, and efficiency of a power system. Metering infrastructure plays a significant role between power supply and demand ends. To upgrade the service quality and provide new services, many utilities adopt AMI components including software (e.g., meter data management system) and hardware (e.g., smart meters and grid routers). Compared to conventional electric energy meters, such as mechanical meters and Automatic Meter Reading (AMR) meters, smart meters are equipped with a two-way communication module to exchange data (e.g., customer's information, power readings, and control commands) between customers and a utility. Based on the real-time data acquisition and control capability, AMI facilitates power flow reading, load forecasting, demand response, outage management, system monitoring, and dynamic pricing programs. However, cyber-physical system (CPS) security has become a significant concern to the smart grid infrastructure, as well as AMI devices. In 2015 and 2016, cyber attacks on the Ukrainian power grid [1], [2] have demonstrated that power grids are vulnerable to cyber intrusions.

Cyber security of the AMI network is widely recognized as a critical issue [3-6]. For power consumers, data privacy is a primary concern as current meters are upgraded to smart meters [7]. To guarantee the confidentiality of data, a new communication protocol has been proposed [8]. In [9], an encryption scheme has been developed for AMI network messages with minimal computation and communication overheads in encryption and decryption operations. For utilities, data integrity and availability attacks can threaten the quality of power grid services and revenues. To prevent energy theft, various studies have proposed different detection algorithms by analyzing historical and present consumption data [10-13]. Reference [14] discusses energy theft through the pricing system. It is aimed at a long term detection technique to capture anomaly pricing events. Due to vulnerabilities of wireless communication and physical devices, meter tampering is one of the potential attacks. In [15], a collaborative intrusion detection mechanism is proposed to detect False Data Injection (FDI) attacks. The work of [16] introduces a specification-based intrusion detection system for advanced metering infrastructures. Any sequence of operations executed outside the system's specifications is considered a security violation. To develop a comprehensive solution, the authors of [17] propose an IDS architecture which covers the entire AMI network, including AMI headend (e.g., meter data management system), grid router, and smart meters. Machine Learning (ML) based detection algorithms can handle multiple attack types.

This paper proposes an IDS that includes two detection processes for smart meters to identify malicious behaviors which are intentionally driven by humans. In comparison with existing detection systems, the proposed design can handle different intrusion types rather than only focus on a specific intrusion type (e.g., energy theft or FDI). The individual purposes of the two detection processes are: (i) collecting intrusion evidence, and (ii) confirming an intrusion event through the detected abnormal behaviors in the system. At the first stage, the SVM technique is used to identify suspicious behaviors in a smart meter and report to the IDS. Relative to other intrusion detection techniques (e.g., knowledge- and

This paper is based upon work supported by the Department of Energy under award number DE-OE0000780 and National Science Foundation under award number ECCS-1824577.

C.-C Sun, D.J. Sabastian, and A. Hahn are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, USA (email: chih-che.sun@wsu.edu; d.sebastiancardenas@wsu.edu; a.hahn@wsu.edu)

C.-C. Liu is with the Power and Energy Center, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA, and Washington State University, Pullman, WA 99164, USA (email: ccliu@vt.edu).

anomaly-based), ML-based detection systems are easier to maintain due to the fact that detection accuracy can be achieved by re-training the model of the classifier when new system data is available.

SVM classifier is a useful tool to detect abnormal behaviors. It provides fast response and does not require heavy computational effort. This feature meets the requirement of IDS for smart meters. However, the reported abnormal behaviors may include the communication failure events which are not caused by cyber attacks. In order to reduce the false alarm ratio by excluding the communication failure events, a comprehensive anomaly-based detection algorithm is developed. The SVM classifier is used to avoid excessive usage of the second-stage detection algorithm. Only high risk events are sent to the anomaly-based IDS for advanced inspection. Therefore, the two-stage detection process may cause an extra processing burden. However, most of low risk events bypass the second-stage detection. Hence, the proposed two-stage detection scheme is able to reduce the false alarm ratio and limit the usage of computation power for smart meter applications. In addition, SVM requires less training time than Neural Network (NN) algorithms (e.g., feedforward, recurrent, and convolutional) which can handle a vast amount of data in AMI networks. A shorter training time implies that smart meters can update the SVM-based IDS in a more responsive manner. When an unknown attack event is identified, the new SVM model can be trained and sent within a shorter time to seek the defense power of smart meters.

At the second stage, the intrusion detection process calculates the similarity between the reported abnormal behaviors and pre-defined intrusion events. To determine whether the intrusion alarm is caused by a random system failure or a cyber attack, the potential attack routes are proposed to provide information with abnormal behavior sequences in four types of cyber attack events. If a detected abnormal event is matched with any of the predefined sequences, it is considered an intrusion event. In this research, a CPS test platform has been developed and used to emulate the operation and communication of smart meters. It is a source to collect the smart meter's data for SVM training and testing purposes. In addition, the performance of the proposed IDS has been validated by simulating cyber attacks on this test platform. The test results demonstrate that the proposed detection algorithms are practical for the detection of simulated attacks on emulated AMI devices. The main contributions of this paper are as follows:

1) Developed an on-line detection IDS method that considers the limited computational capability of a smart meter.

2) A pattern matching algorithm is proposed to identify cyber attack events. This is achieved by creating realistic attack paths using the TFPG technique.

3) A realistic cyber-physical system test platform has been developed for smart meters. It is used for validating and evaluating the AMI network, impact of cyber attacks, and performance of IDS. It is also able to generate the training data for the SVM-based detection algorithm.

In the remaining of this paper, Section II describes the vulnerability of smart meters, including hardware and communication components. Section III presents the proposed intrusion detection system for smart meters. Section IV discusses the components of the AMI test platform at Washington State University (WSU). Section V provides the test results of the proposed detection system. The conclusion and future work are stated in Section VI.

II. Cyber Security Vulnerability of Smart Meters

Since most of the AMI devices are not installed in a monitored environment, attackers may study the weaknesses of both wireless communication and physical devices and then launch cyber attacks. This section will discuss the cyber security vulnerabilities of a smart meter.

A. Hardware Vulnerabilities

Fig. 1 shows five primary compartments in a smart meter: (i) Central Processing Unit (CPU), (ii) Random Access Memory (RAM), (iii) communication module, (iv) flash memory (EEPROM), and (v) energy sensors. Since software/hardware components of smart meters are similar to those of other ICT devices, cyber attackers may adapt intrusion techniques from those employed in other software systems. In a smart meter, firmware controls the critical functions that handle the low-level sensor data, data conversion, and data reporting. Since most functionalities are accomplished through software, new functions can be added by performing updates. Firmware upgrades can be deployed using over the air mechanisms, or manually uploaded by using the on-board optical port. Firmware-based attacks can hinder the device's ability to operate as intended; multiple hardware components can be targeted when tampered firmware or settings are compromised by attackers. The possible attack behaviors for different targeted components are:

- **CPU (A1)**: Exhausting CPU's computational resources by installing malware that causes dummy operations.
- **Communication module (A2)**: The communication channels can be disabled or manipulated in unintended manners. In addition, AMI devices communicate in frequency bands that can be easily monitored, jammed, or compromised.
- **RAM (A3)**: RAM exhaustion can also cause metering and communication applications to freeze or slow down. Operating Systems (OS) kernels terminate running application(s) or reboot to handle these faults.
- **Flash memory (A4)**: Attackers can modify recorded consumption data, device calibration, and operation modes can be altered by modifying configuration registers.
- **Sensor (A5)/actuator compromise (A6)**: By sending a tripping command, the utility system can disconnect a customer.
- **Inter-board communications (A7)**: All components shown in Fig. 1 adopt low-level communication protocols that can be analyzed and modified to suit the attacker needs. Due to physical access requirements, these attacks tend to be isolated.

In summary, attackers can launch various types of cyber attacks to impact operations in a distribution system. The consequences of these attacks are reduced utility's revenues, violation of customers' privacy, or, in the worst case, power outages.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.3010230, IEEE Transactions on Smart Grid

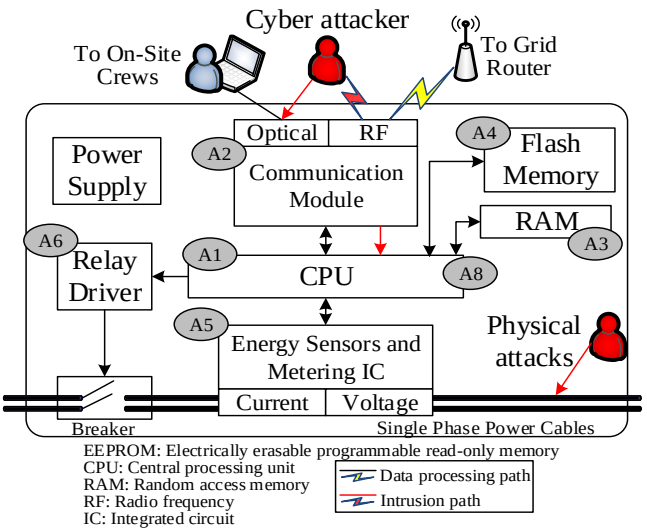


Fig. 1. Hardware components inside a smart meter with potential attack targets.

B. Vulnerability of Wireless Communication

End to end communication in the AMI environment is achieved using a mixture of network architectures, communication protocols, and interfaces between the control center and field devices. Network architectures include: (i) Local Area Network (LAN), (ii) Wide Area Network (WAN), (iii) Neighborhood Area Network (NAN), and (iv) Home Area Network (HAN). Fig. 2 shows the communication structure of an AMI system. This paper is focused on securing the communication path within the NAN domain. The initial AMI meters deployed in North America used Zigbee, while newer models use the IEEE 802.15.4g standard, either at the sub-GHz (i.e., 900MHz) or 2.4 GHz [18]. Both frequency bands fall under the Industrial, Scientific, and Medical (ISM) regulatory domain. Therefore, frequencies are public and can be used by other devices. Furthermore, the wide availability of sniffers, signal modulators, and demodulators raises the overall risk levels since these tools are accessible and affordable. To reduce these risks, AMI devices use encrypted messages for data communication, for achieving integrity and confidentiality, while using meshed networks to provide availability under the CIA triad requirements [19]. However, security flaws have been discovered even with these mitigation efforts. Some security issues are:

- **Privacy issues:** Packet encryption protects the payload content, but it fails to protect the identity of the sender and receiver (MAC addresses) [1]. Furthermore, researchers have been able to identify the usages (e.g., control commands and consumption reports) of different network packets even when they are encrypted [2]. Such knowledge can be used for attacks that target specific operations.
- **Integrity**: By using hardware forensics, local HAN passphrases can be recovered. These in conjunction with spoofed MACs can be used to create false network messages if the devices are not authenticated.
- **Availability:** Signal jamming, as well as DoS attacks, can limit message transmission, leading to situations where the control center cannot send commands, or the device is unable to report its status.

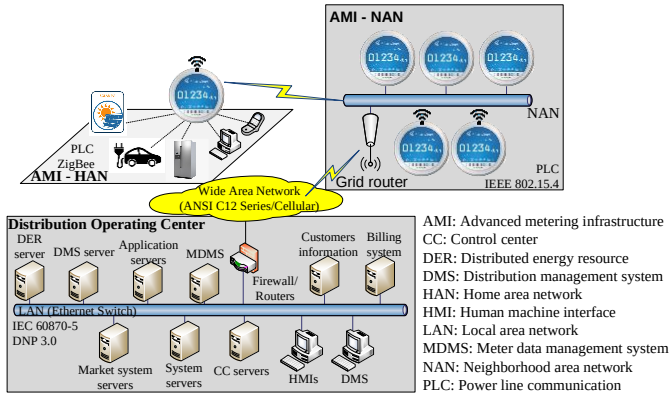


Fig. 2 Communication structure of an AMI network.

III. Intrusion Detection System

This section introduces two detection algorithms for: (i) detecting abnormal behaviors in a smart meter, and (ii) recognizing cyber attack attempts. The IDS for smart meters should consider the limited computational resources. Although existing IDSs provide excellent defense capabilities against cyber attacks, the detection function may consume significant computational resources and impact the operation of smart meters. The framework of the proposed IDS includes three parts: (1) anomaly detection (SVM classifier), (2) intrusion detection (pattern matching algorithm), and (3) information flow between smart meters and a control center for exchanging training data and SVM model. This multi-stage algorithm is designed to outperform other SVM-based anomaly detection techniques when computational resources are limited. Fig. 3 depicts the architecture of the proposed IDS. The specific functions for each block are described in the following subsections.

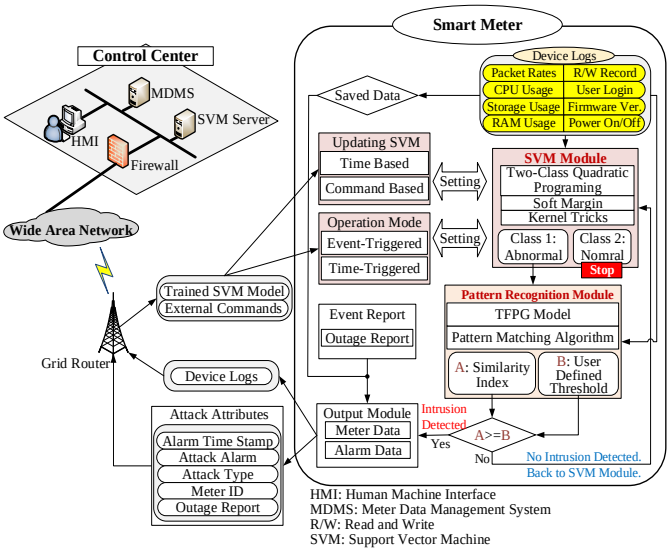


Fig. 3 Structure of the proposed intrusion detection system for smart meters.

A. Support Vector Machine Detection Technique

SVM is a kernel-based supervised learning algorithm to analyze associated data for solving classification and regression problems. SVM classifiers find an optimal hyperplane to separate data points by maximizing the margin between a hyperplane and support vectors in each class. Equation (1)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.3010230, IEEE Transactions on Smart Grid

denotes the optimization problem for the soft-margin hyperplane with the nonnegative slack variable ξ:

$$\begin{aligned} &\text{Minimize:} && Q(w,b,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{M}\xi_i \\ &\text{Subject to:} && y_i(w^T x_i + b) \geq 1 - \xi_i, \\ & && \xi_i \geq 0, \\ & && i = 1,\cdots,M \end{aligned} \tag{1}$$

where w and x_i $(i = 1,\cdots,M)$ are m-dimensional weight and input vectors, respectively. The symbol b is a bias term, while y_i is the class indicator. The tradeoff between the maximization of the margin and minimization of the classification error is determined by the margin parameter C.

To reduce the impact on training ability caused by the margin parameter in soft-margin SVMs, kernel tricks are used to improve the linear separability of training data. By using a nonlinear vector function $\phi(x) = (\phi_1(x),\cdots,\phi_l(x))$, the m-dimensional input vector x can be mapped into the l-dimensional feature space. The decision function in the feature space is expressed as:

$$D(x) = w^T\phi(x) + b \tag{2}$$

In terms of solving the quadratic optimization problem of SVM, each training data point is in the form of dot products. To simplify the calculation of dot product terms, $\langle\phi(x_i),\phi(x_j)\rangle$, a kernel function K is introduced:

$$K(x_i,x_j) = \phi(x_i)\cdot\phi(x_j) \tag{3}$$

The properties of a training dataset affect the performance of kernel functions. In general, existing kernel functions can be categorized into two classes, and it can be a guideline for selection of a feasible kernel [20]. Reference [21] provides a comprehensive list of kernel functions for SVMs.

1) *Local kernels:* Only nearby data points can affect the SVM model. It has a higher learning ability, but the generalization ability is lower. It is used as a general-purpose kernel when there is no prior knowledge about the training dataset.

2) *Global kernels:* Allowing data points from a greater distance to affect the SVM model. It has a higher generalization ability, but the learning ability is lower.

In Fig. 4, Radial Basis Function (RBF) and Polynomial kernels are selected as the local and global kernel functions for the test, respectively [22]. To calculate the kernel values for each kernel function, the test input is set as $v = 1$. Fig. 4(a) shows the closer the test input, the greater the kernel value for the different free parameters (σ). This result implies only the nearby data points have an influence on the kernel value. Due to a local kernel function that may discard or weaken the influence of some training data points, the SVM model loses the generalization property. However, it pays more attention to a certain number of data points located in a smaller range. Thus, it improves the learning ability by increasing the depth and sacrificing the breadth of the information. The global effect of the Polynomial kernel function of different degrees is presented in Fig. 4(b). It shows that every data point from the set μ has an influence on the kernel value of the test input v.

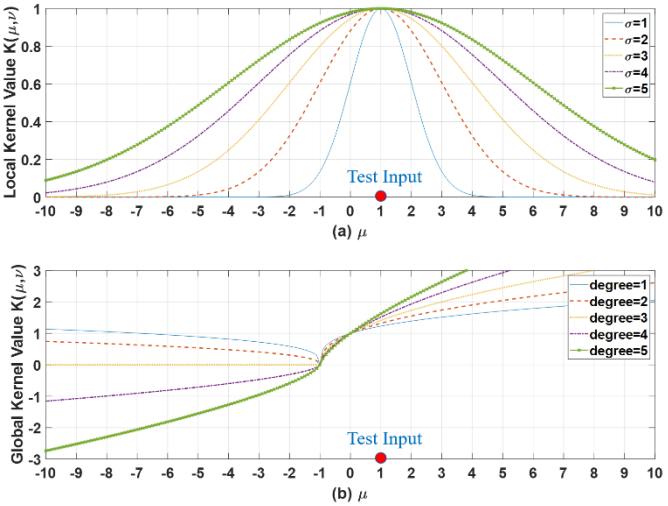


Fig. 4 Kernel values of (a) a local kernel function (RBF) and (b) a global kernel function (Polynomial).

Typically, smart meters have limited processing power. To minimize the consumption of computational resources in a smart meter, the proposed SVM based detection process integrates with two auxiliary control blocks: (i) updating, and (ii) operation mode, for the timing about updating SVM model and triggering the detection process. According to the classification process of the proposed SVM setting, an abnormal event indicator ADS_{ind} is given to indicate the status of input data.

$$ADS_{ind} = \begin{cases} 1, & if\ D(x) = Class1\ (Abnormal) \\ 0, & if\ D(x) = Class2\ (Normal) \end{cases} \tag{4}$$

Once $ADS_{ind} = 1$ is given by the SVM classifier, the second stage of the proposed IDS is activated to identify an intrusion event by collecting evidence from device logs of a smart meter. Otherwise, the detection process stays silent until the next round of system inspection.

B. Pattern Recognition Algorithm for Intrusion Detection

Arbitrarily reported abnormal events cannot serve as conclusive evidence to identify an attack caused by system failures (e.g., communication delay, low battery, and poor data sampling). Too many false intrusion alarms may affect the operation of a distribution system. Therefore, an anomaly-based detection mechanism is developed to perform the inspection. To help intrusion detection systems successfully identify malicious behaviors by attackers, this paper proposes to construct attack routes for determining anomaly paths of each threat type. A TFPG [23] is a model-based diagnosis technique for a dynamic system. It was used for capturing the causal and temporal relationships between failures and consequences in a system. This feature can also be used for modeling temporal relationships between abnormal behaviors (cause) and attack types (effect). Fig. 5 shows an example TFPG model for describing cyber attacks in smart meters. In the TFPG model, abnormal event nodes and arrows illustrate different attack routes. In this paper, four types of cyber attacks are included in the proposed TFPG model:

— **Denial of Service:** Attackers may use a transmitter to create a tremendous amount of wireless signal, congesting the communication channel(s) of smart meters. The dummy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.3010230, IEEE Transactions on Smart Grid

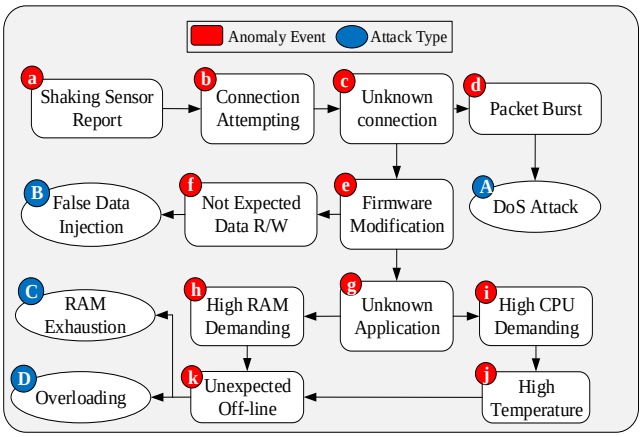


Fig. 5 Attack routes for smart meters.

network packets block the communication with other meters or a grid router. This attack type does not impact the integrity and confidentiality of smart meters' data, but the low availability has a negative effect on the power system services.

— **False Data Injection:** Attackers are able to access victim smart meters and send the commands via an AMI network. The commands include: (1) requisition of data (consumption data and/or meter status log file from a victim meter and (2) request of modifying (i.e., overwrite, insert and delete) any of the data points stored in the meter. The falsified data may impact power system services and mislead the operators to take unwanted actions on the power system.

— **RAM Exhaustion:** A malware could be installed in a smart meter, generating dummy data to fill the memory. When the available memory capacity is low, some application processes become slow or even freeze. It may cause data loss, device freeze, or frequent reboot.

— **CPU Overloading:** A malware may be installed in a smart meter, generating processes that consume heavy computational power (e.g., matrix multiplication). Except for the consequence of RAM exhaustion, smart meters may be physically damaged because of the heat produced by CPU operations.

A series of abnormal events will be considered an intrusion behavior only if they are detected in a sequence that matches the predefined attack routes. Otherwise, the detected abnormal events will be regarded as a system failure. A description of abnormal behaviors is provided in TABLE I. In the design of the proposed IDS, two assumptions are made: (i) intruders' actions follow the sequence in the proposed attack routes, and (ii) IDSs have a false negative problem and fail to capture one or more abnormal events. Under these assumptions, the edit distance can be utilized as the method for attack pattern recognition [24], [25].

In the TFPG model, each abnormal event is assigned an English letter from the alphabet as shown in Fig. 5. Each path, $P \in \{P_1, P_2, P_3, P_4\}$, from the first abnormal event node (i.e., node a) to an attack type node (i.e., nodes A, B, C, and D) is considered a correct sequence in a dictionary as shown in TABLE II. The similarity is measured by the edit distance $\boldsymbol{d}$, between the input and predefined patterns in the dictionary. Once the first abnormal behavior is detected, the IDS starts to record the sequence of abnormal events. At each time stamp, it

TABLE I
ABNORMAL EVENTS FOR SMART METERS IN TFPG MODEL

	Abnormal Behaviors	Description
a	Shaking Sensor Report	Smart meters have an onboard sensor to detect suspicious vibration events.
b	Connection Attempt	Too many incorrect password attempts are likely from an unauthorized user.
c	Unknown Connection	Smart meters have fixed communication parent/children nodes. Any exception is regarded as an abnormal behavior.
d	Packet Burst	Smart meters are configured to send beacon and measurement data at every fixed time cycle. The incoming command from a control center is not a typical case.
e	Firmware Modification	The firmware should be kept at the latest version.
f	Not Expected Data R/W	The measurement data is written and sent to an MDMS at every fixed time cycle.
g	Unknown Application	Smart meters are not allowed to install any third-party software by customers.
h	High RAM Demanding	The routine tasks of smart meters are not designed to over consume the RAM.
i	High CPU Demanding	The routine tasks of smart meters are not designed to over consume the CPU.
j	High Temperature	The electronic components can only work within a specific range of temperature.
k	Unexpected Off-line	Smart meters are designed to operate 24 hours a day.

TABLE II
ATTACK ROUTE SET GENERATED FOR SMART METERS

Attack Path	Attack Type	Dictionary (Sequence of Abnormal Events)
P_1	DoS Attack (A)	bcd
P_2	False Data Injection (B)	abcef
P_3	RAM Exhaustion (C)	abceghk
P_4	Overloading (D)	abcegijk

computes the minimum edit distance $\boldsymbol{ED}$. The calculation is done by the Wagner-Fischer algorithm [26]. The $\boldsymbol{ED}$ is defined as the minimum number of edit operations that match one pattern to another. In this paper, the edit operations include (i) W_{del}: delete a single symbol, (ii) W_{ins}: insert a single symbol, and (iii) W_{trans}: transposition of two successive symbols. Each operation is counted as a unit cost by giving $W_{del} = W_{ins} = W_{trans} = 1$. The calculation process is based on the observation between all prefixes of the first pattern $\boldsymbol{a}$ as well as the second pattern $\boldsymbol{b}$, where the lengths are $\boldsymbol{m}$ and $\boldsymbol{n}$, respectively. A matrix is created to hold each edit distance of prefixes of two patterns. All the values in the matrix are filled by repeating the observation between prefixes of two patterns. Then, the last computed distance, d_{mn}, is the distance ($\boldsymbol{ED}$) between two full strings. In [27], the computation of the edit distance between two finite strings, "$\boldsymbol{a}$" and "$\boldsymbol{b}$", is defined as "$traces$". A trace, $T_{a,b}$, from sequence $\boldsymbol{a}$ to $\boldsymbol{b}$, is a sequence of ordered pairs of integers (i,j) that satisfy:

1) $1 \le i \le m$ and $1 \le j \le n, where\ m$ and n are lengths of string $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively.

2) Any of two pairs (i_1, j_1) and (i_2, j_2) in $T_{a,b}$, (a) $i_1 \ne i_2$, $j_1 \ne j_2$; (b) $i_i < i_2\ iff\ j_1 < j_2$.

Take two strings "Ryan" and "Ray" as an example. A person can easily match Ryan to Ray in two steps: (1) delete "n" and (2) swap "y" and "a." In this case, the edit distance is 2. However, computers need to execute a series of comparison

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.3010230, IEEE Transactions on Smart Grid

processes from left to right, character by character. First, the prefixes for two strings are: {R}, {Ry}, {Rya}, {Ryan}, and {R}, {Ra}, {Ray}. Then, the entire comparison step is listed as follows: ({R},{R}), ({R},{Ra}), ({R},{Ray}), ({Ry},{R}), ({Ry},{Ra}), ({Ry},{Ray}), ⋯, ({Ryan},{Ray}). Therefore, all pairs of prefixes are compared to obtain the edit distance of the two strings. The calculation for the matrix elements can be formulated as:

$$d_{i0} = \sum_{k=1}^{i} W_{ins}(a_k) \qquad for\ 1 \le i \le m \tag{5}$$

$$d_{0j} = \sum_{k=1}^{j} W_{del}(b_k) \qquad for\ 1 \le j \le n \tag{6}$$

$$d_{ij} = \begin{cases} d_{i-1,j-1}, & if\ a_i = b_j \\ d_{i-2,j-2} + W_{trans}(b_{j-1}, b_j), & if\ a_{i-1}a_i = b_j b_{j-1} \\ \min\{d_{i-1,j} + W_{ins}(a_i), d_{i,j-1} + W_{del}(b_j)\ \}, & otherwise \end{cases} \tag{7}$$

$$ED = d_{mn} \tag{8}$$

Wagner-Fischer Algorithm

Input: $a = a_0, a_1, \cdots, a_m$ and $b = b_0, b_1, \cdots, b_n$
Output: Edit Distance (***ED***)

```
// Using Eq (5) and Eq (6) to fill the first row and first column.
for i = 0 to m do
    d_i0 = i;
end for
for j = 0 to n do
    d_0j = j;
end for
// Using Eq (7) to fill the matrix other than the first row and column.
for i = 1 to m do
    for j = 1 to n do
        if (a_i = b_j)
            d_ij = d_{i-1,j-1};
        elseif (a_{i-1}a_i = b_j b_{j-1})
            d_ij = d_{i-2,j-2} + 1;
        else
            d_ij = min{d_{i-1,j} + 1, d_{i,j-1} + 1};
        end if
    end for
end for
// Using Eq (8) to obtain the edit distance
Edit Distance (ED) = d_mn;
```

Fig. 6 demonstrates how to calculate the edit distance by using Wagner-Fischer algorithm. Assuming a sequence of abnormal behaviors is detected in an FDI attack. Due to non-ideal factors, there exist missing or mistaken reports in the abnormal event sequence. In the example in Fig. 6, the IDS fails to capture event "e" and is mistaken in the sequence of events "b" and "c." Thus, the IDS calculates the similarity between "acbf" (detected event) and the attack path P_2 in TABLE II. The

		Pattern ***b*** **(n= 4)** Detected Sequence				
		Null (Ø)	a	c	b	f
Pattern ***a*** **(m= 5)** From Dictionary (P_2)	Null (Ø)	0	1	2	3	4
	a	1	0	1	2	3
	b	2	1	2	1	2
	c	3	2	1	1	2
	e	4	3	2	2	2
	f	5	4	3	3	2 (**ED**)

Fig. 6 Computing distances with matrix by Wagner-Fischer algorithm.

elements in the first row and column are decided by (5) and (6), respectively. Then, the rest of the blanks can be calculated by (7). Once the last element (corner at the bottom right) is filled, the ***ED*** between two patterns is obtained.

Using the pattern recognition algorithm, the detected abnormal event is compared with each pre-defined cyber attack in the dictionary (TABLE II) and obtained a ***ED*** value. Among all four calculated ***ED*** values ($ED_1, ED_2, \cdots, ED_4$) according to the attack paths ($P_1, P_2, \cdots, P_4$), the least edit operation is considered the most likely attack type. Then, an attack similarity index, IDS_{ind}, is defined as:

$$IDS_{ind} = max\ \{1 - \frac{ED_i}{Length(P_i)}\} \tag{9}$$

where $i = \{1,2,3,4\}$. Once IDS_{ind} is greater than a user-defined threshold value V_{th}, the detected event is regarded as an intrusion event. Otherwise, it requests another round of inspection of the SVM detection process. The threshold value can be regarded as the sensitivity of the second-stage detection process. A greater threshold value setting requires stronger evidence to identify a cyber attack event. In other words, the time order of the detected anomaly event should be similar to one of the predefined attack paths. Therefore, an extremely high threshold value may cause extra false negatives. In contrast, false positive alarms may increase if the threshold value is low. Therefore, a lower threshold value is suggested in a new or an unknown communication environment to increase the detection rate. On the other hand, a higher threshold is able to reduce the false positives in a well-known and stable communication network.

If ADS_{ind} still equals to 1 in the second round inspection, it reports a system failure alarm to the control center. Otherwise, the proposed IDS stays silent until the next round of system inspection. According to the different combinations of ADS_{ind} and IDS_{ind}, an operator can conclude the system status, which is shown in TABLE III.

TABLE III
SYSTEM STATUS ACCORDING TO REPORTS OF ADS AND IDS

ADS_{ind}/IDS_{ind}	0	1
0	No Suspicious Event	False Alarm(s)
1	Arbitrarily System Failure	Intrusion Event

IV. CYBER-PHYSICAL SYSTEM TESTBED

Training data plays a significant role in every ML-based algorithm and directly affects the performance. Currently, most of the AMI cyber security studies use IDS databases, KDD Cup 99' [28], DARPA 1998 [29], and ADFA-LD [30], to train/test

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.3010230, IEEE Transactions on Smart Grid

an ML-based detection system. Since these datasets were not developed in an AMI network environment, the training result might not be applicable for smart meters. In some other ML applications in AMI (e.g., load forecasting), a proprietary smart meter database is used. These datasets are provided by utilities with their proprietary AMI system. The attributes (e.g., current, voltage, power consumption, and frequency) are not suitable for IDS studies since they only account for power system behaviors. To acquire a feasible dataset in AMI, a CPS testbed has been developed at WSU. Smart City Testbed (SCT) [31] was built for studying the effectiveness of cyber intrusions and mitigation techniques. In this paper, the SCT is extended by adding AMI components. Commercial-grade SBCs capable of operating under IEEE 802.15.4 are available. The performance of CPU and peripheral electronic components is sufficiently high to emulate a real smart meter in terms of the computational and wireless communication capability. The WSU SCT includes 19 actual smart meters that are installed at a student dormitory on campus. To provide communications, the AMI network messages are captured by a transceiver that supports IEEE 802.15.4 communication. After analyzing the network traffic pattern, SBCs are configured to generate the same traffic pattern for collecting the training data for the proposed IDS.

A. Hardware Setting

The selected SBC has a similar hardware structure which is introduced in Section III. The specifications of TABLE IV show the computational capability of the SBC. An SBC has two individual flash memories. The smaller one is for the O/S and core components, whereas applications, meter data, and log files are stored in the embedded Multi-Media Controller (eMMC) memory. The board runs a minimal Linux kernel with a BusyBox shell [32]. This combination provides a basic set of UNIX based commands that are intended to be used for systems with minimal resources. The onboard communication module provides a sub-GHz (902-928 MHz) IEEE 802.15.4 radio used for mesh networking. According to the different modulators, the data rate can be set between 12.5 to 600 Kbps. To establish a realistic AMI communication environment, two SBCs are set as a smart meter and a grid router with 200 Kbps data rate on channel 1 (906 MHz).

TABLE IV
SPECIFICATION OF SINGLE BOARD COMPUTER

Components	Specification
Microprocessor	ARM Cortex A8 32-bit @ 450 MHz with 32 KB L1 cache and 256KB L2 cache
RAM	128MB
EEPROM	256 bytes
Flash Memory	256MB Onboard 4GB eMMC
Communication Module	IEEE 802.15.4g
Power Supply	5 Volts and 2 Amps
O/S	BusyBox

B. Co-Simulation of Emulated AMI Devices and NS-3

The testbed has two parts: (1) simulation and (2) emulation. Since emulation of an AMI network requires numerous physical devices and system configurations, it is not feasible due to limited availability of the equipment and engineering costs. In addition, the interoperability of different hardware increases the difficulty of developing an AMI testbed. In contrast, simulation methods cannot reflect the real communication behaviors since they do not establish communication links by generating network packets. To eliminate the drawbacks of an individual emulation or simulation method, a hybrid test platform for a large-scale AMI network is developed in this research. NS-3 is an open-source discrete-event network simulator that provides multiple sets of C++ and/or Python libraries to develop a test communication network. By using the IEEE 802.15.4 library, a communication model of the AMI network is designed with 5 NANs and over 900 communication nodes. The topology of physical devices is referred to as the existing cellular network. In the proposed intrusion detection method, the smart meters are assumed to send telemetry data (i.e., power consumptions and meter operating status) to the control center using their local NAN. This is achieved by defining a star-like topology that connects the control center and the outfield NANs. Therefore, no multi-hop, or mesh-like communications are required for multi-NAN traversal.

NS-3 provides the TapBridge model to integrate physical communication hosts into network simulations, bridging the real-world environment with the virtual-simulation. Fig. 7 depicts the co-simulation method for the physical devices and a simulated AMI network. This paper is focused on the emulation of smart meters by commercial-grade SBCs. The emulation platform is used to collect training data, simulating cyber attacks and analyzing the impact, and validating the proposed IDS. The NS-3 simulation is to ensure the data exchanging, including power consumption data, beacon signal of smart meters, and distribution of SVM model, will not congest the AMI network.

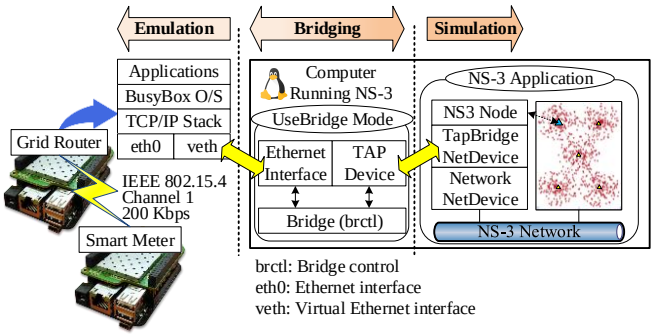


Fig. 7 Co-simulation of IEEE 802.15.4 communication.

V. SIMULATION RESULTS AND ANALYSIS

The proposed AMI test platform is used to simulate different types of attacks and analyze the effectiveness of the proposed detection system at each stage. The performance of different ML algorithms is compared. Three attack scenarios are generated for validation of the proposed pattern recognition algorithm.

A. Training Process of ML Algorithms

The Operating System (O/S) of the SBC supports executing Portable Operating System Interface (POXIS) commands to monitor the device attributes, providing input data for training and testing of the SVM and NN models. No prior knowledge of the cyber attacks on smart meters is assumed. It is generally difficult to determine the criticality of a smart meter's measurements that can most impact the accuracy of the SVM

model. In addition, new types of cyber attacks may appear at any time, causing different symptoms for a smart meter. Therefore, the proposed strategy in this research is to include every measurement of the computer system that can be monitored by a smart meter, such as CPU, RAM, storage, and network traffic readings. TABLE V lists the total of 19 features in this research. To generate the training data, the network packets are generated and sent by the SBCs, which are used to emulate a smart meter, a grid router, and AMI communication. The two sets of the training data are listed as follows:

1) *Normal Data:* Sending one beacon signal every 15 seconds and three copies of a power consumption data point every 20 minutes. In the test dataset, 5000 instances are collected under this class label.

2) *Attack Data:* Except for the routine packet sending, one of the designed attack behaviors is executed simultaneously. The cyber attacks include CPU overloading, memory exhaustion, and packet burst. A total of 1173 instances fall under the attack class in the test dataset.

The Python tool, Scikit-Learn [33], is used for NN and SVM implementation with two typical kernel functions from different categories (i.e., global and local) which are listed in TABLE VI. To enhance the credibility of test results, the random selection method is used to choose a subset from the overall dataset as training data. Three groups of training are conducted for SVM model according to the training ratios, i.e., 80%, 70%, and 60%. Moreover, to demonstrate the influence of kernel functions, different values are applied to kernel parameters, d and γ, in Polynomial and RBF kernels, respectively. Note that degree, d, is a natural number and γ is a positive parameter which is defined as the radius of influence of selected support vectors. To compare the detection performance between different ML algorithms, a Multi Layer Perceptron (MLP) model is selected as a NN algorithm. It has 10 hidden layers with 10 neurons in each layer, and the training ratio is set at 80%.

NS-3 is used to experimentally verify that the overhead traffic, induced by transmission of the SVM training data, does not cause a heavy burden on a large-scale AMI network. In this work, the *utilization ratio* is proposed as a metric to determine the utilization overhead induced by the SVM data-transfer process. That is,

$$Utilization\ ratio = \frac{Occupancy\ of\ communicaiton\ channel}{Sending\ cycle} \quad (10)$$

The pre-trained SVM model has a size of 353 Kbytes, and the throughput of the communication channel is assumed to be 200 Kbps. As a result, it takes 14.12 seconds for the data to travel from the grid router to a smart meter. Since each smart meter sends measurements every 20 minutes, the sending cycle will be set to 1200 seconds. Based on (10), the SVM data-transfer introduces a small overhead (~1.18%), indicating that the proposed SVM-based IDS has a low impact on the operation of an AMI network.

B. Performance of SVM Classifier

Although SVM-based classifiers are often compared to other techniques such as Naïve-Bayes and K-nearest neighbor (KNN) there are some advantages (and disadvantages) that must be considered in terms of computational and time complexities that were analyzed during the development phase of this study. The comparison is summarized as follows:

- **Space complexity:** KNN implementations need to store every data point in the original data set, which can be modeled as $O(n)$. In contrast, SVM classifiers are able to store their training data within the $O(1)$ space. This reduced space complexity is an important aspect to consider when the systems are executed in memory-constrained devices such as smart meters. In terms of space complexity, SVM is preferred.
- **Time complexity:** Time complexity must be considered under two scenarios, training and evaluation. For the first case, KNN has an $O(0)$ complexity, while SVM-based solutions have a relatively high training complexity $O(max(n,d)min(n,d)^2)$, where d represents the dimensional features and n is the number of training examples. Under the evaluation scenario, KNN has an $O(n)$ complexity, while SVM has a complexity of $O(n_{sv})$ (where, n_{sv} is the number of support vectors).

TABLE V
INPUT FEATURES FOR TRAINING AND TESTING PROCEDURES

Category	Features	Unit	Description
RAM	used	Kbytes	RAM that has been occupied.
	free	Kbytes	RAM that can be accessed and utilized.
	buff	Kbytes	RAM is used for file buffers.
	cached	Kbytes	RAM is used for cache memory.
CPU Usage	usr	%	User space processes.
	sys	%	Time spent on running the kernel.
	nice	%	Priority level of processes.
	idle	%	No executing processes.
	io	%	I/O peripherals (e.g, hard drive disc).
	irq	%	Hardware interrupt routines.
	sirq	%	Software interrupt routines.
CPU Average Load	1 min	100%	The running thread (task) demand on the system as an average number of running plus waiting threads.
	2 mins	100%	
	5 mins	100%	
Network Traffic	RX Pkts	count	Number of network packets that have been transmitted/received.
	TX Pkts	count	
	RX data	Kbytes	Size of transmitted/received data.
	TX data	Kbytes	
Storage	Usage	%	Used storage capacity.

TABLE VI
TESTED KERNEL FUNCTIONS FOR SVM

Kernel Name	Category	Kernel Function
Polynomial	Global	$K(\mu,\nu) = (\mu \cdot \nu + 1)^d$
Radial Basis Kernel (RBF)	Local	$K(\mu,\nu) = \exp(-\gamma\|\mu - \nu\|^2)$,

C. Evaluation of ML-Based Detection Techniques

Two common metrics, Detection Rate (DR) and accuracy, are used to evaluate the performance of SVM models. DR is defined as a ratio between numbers of detected and total attack samples, whereas accuracy is measured by the overall True Positive (TP) and True Negative (TN) rates. The outcome of performance metrics is the average values from the 100-rounds test with different selected training/testing dataset as well as the same kernel function, size of datasets, and kernel parameters.

1) *SVM and kernel functions:* TABLE VII provides the testing result of the proposed SVM method with the two kernel functions. It shows that Polynomial kernel does not possess monotonically increasing or decreasing properties when: (i) d, and (ii) training ratio are monotonically increased/decreased. It

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2020.3010230, IEEE Transactions on Smart Grid

TABLE VII
PERFORMANCE COMPARISON OF SVM MODELS FOR SMART METER ADS

Kernel Function		Polynomial			RBF		
Parameter		d			γ		
Metric	Train Ratio	1	2	5	1	2	5
DR (%)	80%	97.06	96.96	96.77	98.31	98.47	98.71
	70%	96.79	96.79	96.54	97.88	98.41	98.63
	60%	96.52	96.45	96.40	96.68	98.06	98.31
Accuracy (%)	80%	98.15	98.56	98.33	99.40	99.41	99.44
	70%	98.02	98.38	98.22	99.24	99.25	99.37
	60%	97.90	98.25	98.18	98.88	99.17	99.30

can be assessed that local kernel functions are more suitable for the collected smart meter data compared with global kernel functions, indicating that there is no strong connection among data features.

2) *Comparison of NN and SVM:* To show the advantage of SVM, the same training process is applied to the MLP model with the minimal setting which can achieve a similar accuracy level. TABLE VIII shows the NN algorithm spends more time to complete the training. In contrast, the longest training time among all the tests of SVM is 0.52 seconds with respect to RBF kernel with $\gamma = 5$ and 80% of training ratio. Comparing to the SVM, the NN algorithm takes over 2.5 times more seconds in the training process. To ensure the ML model can be updated timely when the new AMI data is available, training efficiency is a critical factor to affect the performance of the IDS.

In this test, SVM is shown to be a better ML algorithm for real-time applications in AMI networks, and it is able to identify abnormal behaviors from the network traffic and usages of smart meters (TABLE V).

TABLE VIII
COMPARISON BETWEEN ML ALGORITHMS

ML Algorithm	Train Ratio	Accuracy (%)	Training Time (s)
NN (MLP)	80%	98.22	1.33
SVM (RBF)		98.71	0.52

D. Attack Scenarios for Smart Meters

1) *CPU Overloading (Case1):* In this scenario, attackers are able to access the smart meter physically and open the cover to view the structure of the electronic components. Based on what they learned, they try to crack the login password by brute-force and modify a smart meter's firmware, allowing unauthorized users to install malware. In the following, the malware is installed and executed by the attacker, which is used to create a high volume of dummy load to exhaust the CPU. Since the computing resource is overused, the system becomes slow and freezes. Finally, the smart meter automatically reboots. Therefore, an off-line record is written to the log file after the CPU is overloaded.

2) *RAM Exhaustion (Case2):* Assuming attackers already have the login information of a smart meter. After penetrating the smart meter's internal system (e.g., filesystem and O/S), the malware is installed to create dummy data to fill the RAM. All the application processes in the target meter gradually slow down and freeze. Eventually, the smart meter reboots and loses all the unsaved data.

3) *Denial of Service (Case3):* Attackers have identified and tested the PANID and the communication channel of the victimized smart meter. With the information, attackers use a wireless signal transmitter to create heavy communication traffic by sending dummy network packets to the target.

E. Validation of the Proposed IDS

Since the SVM provides high accuracy in the first stage detection process, the abnormal behaviors trigger the alarm in all the three test cases. Once ADS_{ind} is changed from 0 to 1, the IDS starts to collect the time information of detected abnormal behaviors for the second stage detection process. The test results are provided in TABLE IX.

In Case 1, the IDS fails to detect abnormal event "j" which shows an abnormal temperature of CPU. The sequence of detected abnormal behaviors is aligned along the time axis as "$abcegik$." The proposed pattern matching algorithm obtains IDS_{ind} by finding the maximal similarity between the detected sequence and the pre-defined attack sequences. In this test scenario, the length of P_4 is 8, and the corresponding $\boldsymbol{ED_4}$ is 1. Therefore, IDS_{ind} is calculated as 0.875 by (9), which is the greatest value among all four attack paths. It indicates that the series of suspicious behaviors intends to launch a CPU overloading attack. Since IDS_{ind} is greater than the threshold, i.e., $V_{th} = 0.6$, this event is judged to be an attack. In Case 2, attackers do not physically access the target, and there is not an abnormal report from the shaking sensor. The detected abnormal event sequence is "$bcghk$." The path P_3 generates the largest similarity index IDS_{ind}. The detection system reports this attack event as a RAM exhaustion attack. In the last test case, Case 3, the target meter receives a couple of packets from an unknown source address during testing of PANID. This behavior is recognized as the connection attempting. During the attack stage, the communication channel is congested. In this attack event, only "b" and "d" are captured by the IDS. IDS_{ind} is 0.667, indicating that a DoS attack is recognized. Although the event "c" is missing in the attack sequence, the IDS can still identify the cyber attack and the attack type.

TABLE IX
TEST RESULTS OF PROPOSED IDS

Test Case	Detected Abnormal Sequence	Wagner-Fischer Algorithm: Edit Distance for each Attack Path				IDS_{ind}
		ED_1	ED_2	ED_3	ED_4	
Case 1	a→b→c→e→g→i→k	6	4	2	1	0.875
		$P4$: CPU overloading attack (D)				
Case 2	b→c→g→h→k	4	7	2	5	0.714
		P_3: RAM exhaustion attack (C)				
Case 3	b→d	1	5	7	8	0.667
		P_1: DoS attack (A)				
Length of Attack Paths		P_1: 3	P_2: 5	P_3: 7	P_4: 8	

VI. CONCLUSION

The growing number of smart meters on the customer side raised cyber security concerns about potential vulnerabilities of the new technologies. It is shown that intruders can launch a cyber attack by utilizing the vulnerability of hardware components and communication systems of smart meters. This paper proposes an IDS with the two-stage collaborative detection process for smart meters. The SVM classifier is applied as the abnormal behavior detection mechanism in the first stage. As soon as a suspicious behavior is detected, the second stage intrusion detection process is activated. According to the predefined attack routes, which are based on the TFPG technique, the pattern recognition algorithm is able to calculate the similarity index, indicating the likelihood of an intrusion

event as well as the attack type.

An AMI test platform has been developed to provide a co-simulation environment, including emulating wireless communication between a smart meter and a grid router, simulating cyber attacks, collecting training/testing data, and validating the proposed detection system. In this work, the simulated 5 NANs are identical; however, this does not limit the AMI network simulation applicability. Users can apply different network topologies and bridge NANs with physical devices according to their needs. Since the proposed SVM-based detection system only requires local NAN data to classify normal versus abnormal data, it can be claimed that each NAN can operate in a parallel manner with respect to other NANs by only using a limited amount of computing power, $O(n_{sv})$. Therefore, the proposed intrusion detection method is able to scale across multiple NANs as long as the computing requirements of each NAN are met.

The simulation results show that the SVM classifier exhibits good performance with kernel functions in the specific category. Compared to NN algorithms, SVM has an advantage in the shorter training time. This feature allows the proposed SVM model to be frequently updated to maintain a high level of detection accuracy. In the three test attack scenarios, the ML-based detection algorithm identifies abnormal behaviors and triggers the next stage detection process to investigate the sequence of the detected abnormal behaviors. The results show that all test scenarios are recognized by the IDS successfully.

To improve the detection accuracy of the SVM, more features to represent physical system behaviors can be added into the dataset, e.g., power measurement readings from feeders and neighboring meters. In this work, a star-like communication topology is used in the NS-3 simulator to evaluate the network performance after integrating the proposed IDS. Future research needs to be conducted to incorporate other AMI network topologies, a task required for simulation of large-scale AMI networks.

ACKNOWLEDGEMENT

The authors greatly appreciate the reviewers for the valuable comments that are incorporated in the revision.

REFERENCES

[1] SANS and Electricity Information Sharing and Analysis Center (E-ISAC), "Analysis of the Cyber Attack on the Ukrainian Power Grid," Mar. 18, 2016. [Online]. Available: http://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf

[2] Dragos Inc., "Analysis of the Threat to Electric Grid Operations," Jun. 13, 2017. [Online]. Available: https://dragos.com/wp-content/uploads/CrashOverride-01.pdf

[3] C. C. Sun, A. Hahn, and C. C. Liu, "Cyber Security of a Power Grid: State-of-the-Art," *Intl. J. of Electrical Power & Energy Systs.*, vol. 99, pp.45-56, Jan. 2018.

[4] Q. Sun, H. Li, Z. Ma, C. Wang, J. Campillo, Q. Zhang, F. Wallin, and J. Guo, "A Comprehensive Review of Smart Energy Meters in Intelligent Energy Networks," *IEEE Internet of Things J.*, vol. 3, no. 4, pp. 464-479, Aug. 2016.

[5] Y. Liu, S. Hu, and A. Y. Zomaya, "The Hierarchical Smart Home Cyberattack Detection Considering Power Overloading and Frequency Disturbance," *IEEE Trans. Industrial Informatics*, vol. 12, no. 5, pp. 1973-1983, Oct. 2016.

[6] K. I. Sgouras, A. N. Kyriakidis, and D. P. Labridis, "Short-term Risk Assessment of Botnet Attacks on Advanced Metering Infrastructure," *IET Cyber-Physical Syst.: Theory & Applications*, vol. 2, no. 3, pp. 143-151, Oct. 2017.

[7] S. Finster and I. Baumgart, "Privacy-Aware Smart Metering: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1088-1101, Secondquarter 2015.

[8] Y. Yan, R. Q. Hu, S. K. Das, H. Sharif, and Y. Qian, "An Efficient Security Protocol for Advanced Metering Infrastructure in Smart Grid," *IEEE Network*, vol. 27, no. 4, pp. 64-71, Aug. 2013.

[9] A. Alsharif, M. Nabil, M. M. E. A. Mahmoud, and M. Abdallah, "EPDA: Efficient and Privacy-Preserving Data Collection and Access Control Scheme for Multi-Recipient AMI Networks," *IEEE Access*, vol. 7, pp. 27829-27845, 2019.

[10] P. Jokar, N. Arianpoo, and V. C. M. Leung, "Electricity Theft Detection in AMI Using Customers' Consumption Patterns," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 216-226, Jan. 2016.

[11] Y. Liu and S. Hu, "Cyberthreat Analysis and Detection for Energy Theft in Social Networking of Smart Homes," *IEEE Trans. Computational Social Syst.*, vol. 2, no. 4, pp. 148-158, Dec. 2015.

[12] Y. Liu and S. Hu, "Cyberthreat Analysis and Detection for Energy Theft in Social Networking of Smart Homes," *IEEE Trans. Computational Social Syst.*, vol. 2, no. 4, pp. 148-158, Dec. 2015.

[13] S. McLaughlin, B. Holbert, A. Fawaz, R. Berthier, and S. Zonouz, "A Multi-Sensor Energy Theft Detection Framework for Advanced Metering Infrastructures," *IEEE J. Selected Areas in Communications*, vol. 31, no. 7, pp. 1319-1330, Jul. 2013.

[14] Y. Liu, S. Hu, and T. Ho, "Leveraging Strategic Detection Techniques for Smart Home Pricing Cyberattacks," *IEEE Trans. Dependable and Secure Computing*, vol. 13, no. 2, pp. 220-235, 1 Apr. 2016.

[15] X. Liu, P. Zhu, Y. Zhang, and K. Chen, "A Collaborative Intrusion Detection Mechanism Against False Data Injection Attack in Advanced Metering Infrastructure," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2435-2443, Sept. 2015.

[16] R. Berthier and W.H. Sanders, "Specification-Based Intrusion Detection for Advanced Metering Infrastructures," *IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, Pasadena, CA, USA, pp. 184-193, Dec. 2011.

[17] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez, "Data-Stream-Based Intrusion Detection System for Advanced Metering Infrastructure in Smart Grid: A Feasibility Study," *IEEE Syst. J.*, vol. 9, no. 1, pp. 31-44, Mar. 2015.

[18] R. Ullah, Y. Faheem, and B. Kim, "Energy and Congestion-Aware Routing Metric for Smart Grid AMI Networks in Smart City," *IEEE Access*, vol. 5, pp. 13799-13810, 2017.

[19] W. Stallings and L. Brown, "Computer Security Concepts" in *Computer Security Principles and Practice*, 2nd ed. London, UK: Pearson, 2012, ch. 1, sec. 1, pp. 10-17.

[20] L. Liang, Q. Wang, and Y. Chen, "Application of Support Vector Machine in Online Monitoring of Wastewater Treatment based on Combined Kernel Functions," in *2011 Intel. Conf. Electrical and Control Engineering*, Yichang, pp. 3840-3843, 2011.

[21] J. Zhang, "A Complete List of Kernels Used in Support Vector Machines," *Biochemistry & Pharmacology: Open Access*, vol. 4, no. 5, pp. 195, Oct. 2015.

[22] V. L. Brailovsky, O. Barzilay, and R. Shahave, "On Global, Local, Mixed and Neighborhood Kernels for Support Vector Machines," *Pattern Recognition Letters*, vol. 20, no. 11-13, pp. 1183-1190, 1999.

[23] S. Abdelwahed, G. Karsai, N. Mahadevan, and S.C. Ofsthun, "Practical Implementation of Diagnosis Systems Using Timed Failure Propagation Graph Models," *IEEE Trans. Instrumentation and Measurement*, vol. 58, no. 2, pp. 240-247, Feb. 2009.

[24] T. Okuda, E. Tanaka, and T. Kasai, "A Method for the Correction of Garbled Words Based on the Levenshtein Metric," *IEEE Trans. Computers*, vol. C-25, no. 2, pp. 172-178, Feb. 1976.

[25] N. D. L. R. K. Chaurasiya, and S. Ghosh, "A Novel Weighted Edit Distance-Based Spelling Correction Approach for Improving the Reliability of Devanagari Script-Based P300 Speller System," *IEEE Access*, vol. 4, pp. 8184-8198, 2016.

[26] W. Masek and M. A. Paterson, "Faster Algorithm Computing String Edit Distances," *Computer System Sci.*, vol. 20, pp. 18-31, 1980.

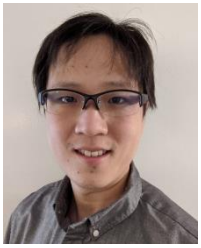
[27] R. A. Wagner and M. J. Fischer, "The String-to-String Correction Problem," *J. Assoc. Comput. Mach.*, vol. 21, no. 1, pp. 168-173, 1974.

[28] The KDD99 dataset [Online]. Available: http://kdd.ccs.uci.edu/databases/kddcup99/task.html

[29] I. S. T. G. MIT Lincoln Lab, "DARPA Intrusion Detection Data Sets," http://www.ll.mit.edu/mission/communications/

[30] G. Creech and J. Hu, "The ADFA Intrusion Detection Datasets," [Online]. Available: https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-IDS-Datasets/
[31] C. C. Sun, J. Hong, and C. C. Liu, "A Co-Simulation Environment for Integrated Cyber and Power Systems," in *2015 IEEE Intl. Conf. Smart Grid Commun. (SmartGridComm)*, Miami, FL, pp. 133-138, 2015.
[32] N. Wells, "BusyBox: A Swiss Army Knife for Linux," *Linux J.*, Nov. 2000. [Online]. Available: http://busybox.net/
[33] F. Pedregosa et al., "Scikit-Learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Feb. 2011.

BIOGRAPHIES



Chih-Che Sun (S'15-M'20) received the Ph.D. degree from the Department of Electrical Engineering and Computer Science at Washington State University, Pullman, WA, USA, in 2019.

He is currently a postdoctoral research staff with Lawrence Livermore National Laboratory, Livermore, CA, USA. His research interests include cyber-physical systems (CPS) security, and modeling and simulation.

D. Jonathan Sebastian (M'20) received his B.E. and M.Sc. degrees in electrical engineering from Instituto Politécnico Nacional, Mexico City, Mexico, in 2013 and 2015, respectively. He is currently finalizing his Ph.D. in computer science at Washington State University, Pullman, WA, USA.

Within the professional field, he has collaborated in the development of training simulators and market analysis tools that are in use by the industry. His current research interests include cybersecurity of cyber-physical systems and data privacy with an emphasis towards IoTs and IBRs.

Adam Hahn (S'06-M'13) is an assistant professor in the Department of Electrical Engineering and Computer Science at Washington State University. His research interests include cybersecurity of the smart grid and cyber-physical systems (CPS), including intrusion detection, risk modeling, vulnerability assessment, and secure system architectures. He received M.S. and Ph.D. degrees from the Department of Electrical and Computer Engineering at Iowa State University in 2006 and 2013. Previously, he worked as a Senior Information Security Engineer at the MITRE Corporation.

Chen-Ching Liu (S'80–M'83–SM'90–F'94-LF"19) received the Ph.D. degree from the University of California, Berkeley, CA, USA. He is currently American Electric Power Professor and Director of the Power and Energy Center, Virginia Tech, Blacksburg, VA, USA. He was the Chair of IEEE PES Technical Committee on Power System Analysis, Computing, and Economics. Dr. Liu is the U.S. Member of CIGRE Study Committee D2, Information Systems and Telecommunication. He is a Member of the U.S. National Academy of Engineering.