ABSI: An Adaptive Binary Splitting Algorithm for Malicious Meter Inspection in Smart Grid

Xiaofang Xia, Yang Xiao, *Senior Member, IEEE*, and Wei Liang, *Senior Member, IEEE*

***Abstract*— Electricity theft is a widespread problem that causes tremendous economic losses for all utility companies around the globe. As many countries struggle to update their antique power systems to emerging smart grids, more and more smart meters are deployed throughout the world. Compared with analog meters which can be tampered with by only physical attacks, smart meters can be manipulated by malicious users with both physical and cyber-attacks for the purpose of stealing electricity. Thus, electricity theft will become even more serious in a smart grid than in a traditional power system if utility companies do not implement efficient solutions. The goal of this paper is to identify all malicious users in a neighborhood area in a smart grid within the shortest detection time. We propose an adaptive binary splitting inspection (ABSI) algorithm which adopts a group testing method to locate the malicious users. There are two considered inspection strategies in this paper: a scanning method in which users will be inspected individually, and a binary search method by which a specific number of users will be examined as a whole. During the inspection process of our proposed scheme, the inspection strategy as well as the number of users in the groups to be inspected are adaptively adjusted. Simulation results show that the proposed ABSI algorithm outperforms existing methods.**

***Index Terms*— Electricity theft, smart grid, security, group testing.**

I. Introduction

As a promising power infrastructure, smart grid is being introduced to more and more countries, such as USA, Japan, and China [1], [2]. To make electrical grids “smart”, a multitude of modern hardware and software techniques are integrated into power systems [3], [4]. For example, analog meters in traditional power systems are upgraded to digital smart meters, which have capabilities of computation, communication, and remote control [5]–[7]. Besides, a cyber layer is added to the metering system. Unfortunately, while these techniques bring us convenience and efficiency, they also enable malicious users to apply numerous new ways to steal electricity, where malicious users are referred to as the users stealing electricity.

Compared to analog meters which can be tampered with by only physical attacks, such as directly tapping into power lines and bypassing energy meters, smart meters can also be manipulated with cyber attacks. It is reported that users with a moderate level of computer knowledge are able to hack into the digital chips of smart meters, with low-cost tools and software readily available on the Internet [8]–[10]. Another commonly used method to steal electricity is to bribe employees in utility companies. These employees will then log into the electricity consumption database of their utility companies, and manipulate malicious users’ readings to smaller numbers and even make them unregistered.

Almost all utility companies around the globe, especially those in many emerging market countries [11], suffer from electricity theft. Currently, according to a new study published by Northeast Group, LLC, the world loses $89.3 billion annually due to electricity theft, among which the top 50 emerging market countries lose $58.7 billion per year [11]. The highest losses were in India ($16.2 billion), followed by Brazil ($10.5 billion) and Russia ($5.1 billion). It is said that 80% of worldwide electricity theft occurs in private dwellings and 20% on commercial and industrial premises. Provided that utility companies do not implement efficient solutions, electricity theft will become even more serious in smart grids than in traditional power systems [12]–[14].

Many research works have been done on detection of electricity theft in smart grid. Among these works, the most important techniques are classification-based methods and power measurement-based methods. The classification-based methods usually apply various machine learning methods, such as support vector machine and extreme learning machine, to analyze users’ fine-grained electricity consumption readings, aiming at recognizing users’ abnormal behaviors highly related to electricity theft [15]. However, these methods usually have the shortcomings of low detection rates and high false positive rates. With regard to the power measurement-based methods,

Manuscript received January 6, 2018; revised May 21, 2018 and July 1, 2018; accepted July 2, 2018. Date of publication July 10, 2018; date of current version August 8, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0101300, in part by the U.S. National Science Foundation under Grant CNS-1059265, in part by the National Natural Science Foundation of China under Grant 61374200 and Grant 71661147005, and in part by the Smart Manufacturing Standardization Project of the Ministry of Industry and Information Technology of the People’s Republic of China–Research and Demonstration of Operation Management and Equipment Interconnection and Interoperation Standards in Digital Workshop for Robotics Manufacturing. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wei Yu. *(Corresponding authors: Yang Xiao; Wei Liang.)*

X. Xia is now with the Key Laboratory of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China. She was also with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, USA.

Y. Xiao is with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290 USA (e-mail: yangxiao@ieee.org).

W. Liang is with the Key Laboratory of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: weiliang@sia.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2018.2854703

their basic idea is to install redundant devices to monitor users' electricity consumptions. This category of methods can usually identify malicious users accurately. Nevertheless, some approaches (for instance, the mutual inspection strategy in paper [16]) require to deploy in the smart grid a large quantity of extra devices such as sensors and smart meters, which will significantly increase the cost.

For the purpose of cost saving, the authors in paper [17] propose to install a limited number of inspectors for each neighborhood area network (NAN) where inspectors are actually function-enhanced smart meters with larger memory and stronger computation capability. Clearly, fewer inspectors inevitably suggest longer detection time of malicious users. With the goal of identifying all malicious users within the shortest detection time, a series of inspection methods based on logical binary trees are proposed in papers [17]–[19]. Since we recently observe that the electricity theft detection issue has some common features with the group testing problem (which will be explained later), in this paper, we propose to apply a group testing method to electricity theft detection to locate malicious users. The proposed electricity theft detection method in this paper is called Adaptive Binary Splitting Inspection (ABSI) algorithm in which groups of users are tested together and the group size is changed dynamically during the testing process. There are two considered inspection strategies in this paper: a scanning method in which users will be inspected individually [17], and a binary search method by which a specific number of users will be examined as a whole. During the inspection process of our proposed scheme, the inspection strategy as well as the number of users in the groups to be inspected will be adaptively adjusted. The main contributions of this paper are highlighted as follows: First, we propose to apply a group testing method to electricity theft detection to locate malicious users in smart grid in which the inspection strategy as well as the number of users in the groups to be inspected are adaptively adjusted. Second, we provide the performance analysis of the ABSI algorithm, e.g., estimating the minimum upper bound of the number of malicious users and the maximum number of inspection steps (detection time). Third, simulations are conducted to evaluate the performance of the ABSI algorithm. Simulation results show that the proposed ABSI algorithm outperforms existing methods.

The rest of this paper is organized as follows. In Section II, we review some related works. In Section III, we define the problem to be addressed in this paper. In Section IV, we propose the ABSI algorithm and demonstrate how it works. Performance analysis is provided in Section V. Simulation results and conclusion of the paper are reported in Section VI and Section VII, respectively.

II. Related Works

In this section, we review some research works that have been done on detection of electricity theft. As aforementioned, the most two popular categories among these works are the classification-based methods and the power measurement-based methods.

The classification-based methods [20]–[26] leverage various data mining and machine learning technologies to train a classifier with a sample database [24]. For example, in the paper [21], a genetic algorithm and a support vector machine are combined together to identify malicious users. In the paper [20], the extreme learning machine and its online sequential version are utilized. In the paper [25], for improving accuracy, optimum-path forest classifier and the distribution state estimator are jointly used. In the paper [24], the multi-class support vector machine are used together with clustering techniques as well as transformer meters. The above approaches usually involve extracting patterns of customer behaviors from users' historical electricity consumption data. The main purpose is to reveal any significant behaviors highly related to electricity theft [20]. These methods have the advantage of moderate costs. This is mainly because the data which they are dependent upon are naturally generated in users' daily life. Thus, utility companies do not need to pay extra money for the data. However, their shortcomings are also obvious and should not be dismissed. As argued in paper [27], these methods have relatively low detection rates but relatively high false positive rates. This is mainly due to the data imbalance issue [24]. That is to say, the normal samples can be easily obtained, while the abnormal samples rarely or do not exist for a given user. Furthermore, we should not neglect the fact that in the real world, there are many non-malicious factors, such as the normal moving in/out of residents, the change of electrical appliances, and the change of seasonality. These non-malicious factors also affect users' consumption patterns, but are not related to electricity theft. The classification-based methods are not effective unless they can deal with these factors properly.

With regard to the power measurement-based techniques, their basic idea is to install redundant devices to monitor users' electricity consumptions. The mutual inspection strategy [16] requires one extra smart meter to be installed for each user at the end of utility companies. The inspection strategy proposed in paper [28] demands that the number of grid sensors to be deployed in the smart grid should be the same with the total number of users. The above two strategies can identify the malicious users immediately these users commit electricity theft. However, the cost is too huge. A much more economical way is to install one or several devices such as central observer meter or inspectors, as devised in the papers [17]–[19], [29]–[37]. Nevertheless, it unavoidably comes with longer detection time.

To shorten the detection time, Xiao *et al.* [17], Xia *et al.* [18], [19] leverage a binary inspection tree (BIT) as a logic structure to facilitate the inspection process. The Adaptive Tree Inspection (ATI) algorithm [17] is a heuristic inspection approach. By leveraging some information collected during the inspection process, it enables inspectors to skip some internal nodes on the BIT and directly inspect the nodes at lower levels. It shortens the detection time to a large extent, especially when the ratio of malicious users is low. The Binary Coded Grouping-based Inspection (BCGI) algorithm [36], [38] groups users in the NAN according to the binary sequences of their identification numbers. It can locate malicious users by only one inspection step. However, the BCGI algorithm works only when there is a unique malicious user in the NAN.

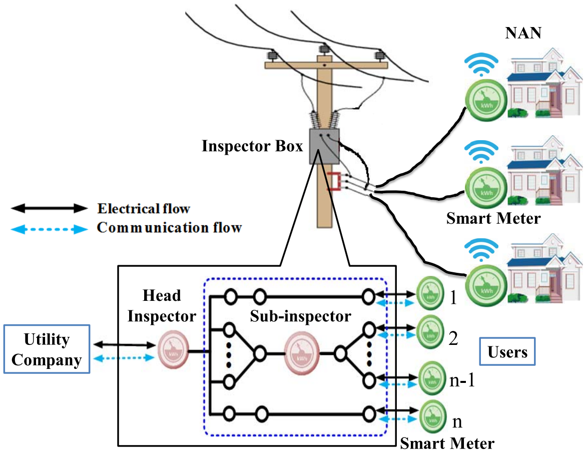


Fig. 1. A conceptual framework for the malicious meter inspection. In each NAN, there installs an inspector box which contains a head inspector and several sub-inspectors. The head inspector is responsible for detecting the existence of reading anomalies; the sub-inspectors take charges of identifying the malicious meters exactly.

In World War II, the group testing problem was first introduced to accelerate and economize the procedure of weeding out individuals infected with syphilitic [39]. The electricity theft detection problem and the group testing problem have a lot in common, among which the following two aspects are the most important: (1) the objects being inspected/analyzed can be classified into two categories; (2) they both aim to conduct as few inspections/ analysis as possible. Specifically, for the electricity theft detection, the users in an NAN can be categorized into malicious users and honest users, where the honest users refers to the users not committing electricity theft. Since each inspection conducted by the inspectors is time-consuming and longer detection time means more economical losses, utility companies intend to locate the malicious users with as few inspections as possible. On the other hand, with regard to the group testing problem, blood samples which needs to be analyzed can be divided into infected samples and pure samples. Since analysis for these samples is cost-heavy, the United States Public Health Service and the Selective Service System intend to weed out the infected samples with the smallest number of analysis [39].

We observe the above similarities, and therefore, in this paper, we propose to apply group testing methods to electricity detection. We propose an Adaptive Binary Splitting Inspection (ABSI) algorithm which adopts a group testing method to locate the malicious users. There are two considered inspection strategies in this paper: a scanning method in which users will be inspected individually, and a binary search method by which a specific number of users will be examined as a whole. During the inspection process of our proposed scheme, the inspection strategy as well as the number of users in the groups to be inspected are adaptively adjusted.

III. Problem Statement

A simplified smart metering system for an NAN is illustrated in Fig. 1. As we can see, a smart meter is installed at each user's premises for the purpose of recording and then periodically reporting electricity consumptions to utility companies. Let n and $U = \{1, 2, \ldots, n\}$ denote the total number of users and the set of all users, respectively, in the NAN. Let q_j and q'_j denote user j's reported reading and actual electricity consumption, respectively. Based upon the relationship between q_j and q'_j, the users in the NAN can be classified into two categories: malicious users whose reported electricity consumptions are less than what they actually consume (i.e., $q_j < q'_j$) and honest users who genuinely report their electricity consumptions (i.e., $q_j = q'_j$). We assume that there is an inspector box [17] installed in the distribution room or on an electrical pole in an NAN, and it acts as a relay node as well as a monitoring device at the same time. The inspector box consists of two kinds of inspectors: a head inspector responsible for detecting the existence of reading anomalies, and several sub-inspectors which aim to exactly locating the malicious users in the NAN. We assume that inspectors are either secure or equipped with tamper-resistant components/functions. Notably, we assume that 1) the head inspector monitors all the users statically; 2) the set of users monitored by the sub-inspectors can be changed automatically or manually; and 3) the sub-inspectors can be effortlessly added into or removed from the inspector box [17]. To pinpoint malicious users, we require one head inspector and at least one sub-inspector in each neighborhood. In the real applications, utility companies may install multiple inspector boxes in different neighborhoods and multiple sub-inspectors in each box for shortening the detection time as much as possible. To a large extent, the budget of utility companies determines the number of inspector boxes and the number of sub-inspectors in each box to be installed.

Let k denote the total number of sub-inspectors in the inspector box. Then, the set of inspectors can be denoted by $I = \{0, 1, 2, \ldots, k\}$, where inspector 0 stands for the head inspector in particular and inspectors $1, 2, \ldots, k$ refer to the sub-inspectors. Let G_i denote the group of users monitored by inspector $i, i \in I$. Then, for the head inspector, we have $G_0 = U$; and for the sub-inspectors, we have $G_i \subset U, \forall i \in I \setminus \{0\}$, where "\" means the set difference operation. For inspectors $i, j \in I \setminus \{0\}$, we have $G_0 \cap G_j = G_j$ and $G_i \cap G_j = \emptyset$. For any inspector $i \in I$, when it works, it operates as follows: (1) measuring the total amount of electricity distributed to the users in G_i, which is denoted as r_i; (2) receiving these users' reported readings; (3) calculating the total amount of stolen electricity of all the users in G_i, which is notated by x_i. When an inspector conducts one time of the above operations, we say it performs one inspection step. Based upon the law of conservation of energy, we have

$$x_i = r_i - \sum_{j \in G_i} q_j - \delta_i, \tag{1}$$

where q_j denotes user j's reported readings, and δ_i represents the total amount of technical losses of the users in G_i. In this paper, we simply assume that δ_i can be estimated based upon some mathematical models [40].

The head inspector works all the time. If it detects reading anomalies, we can infer that there exist malicious users in the NAN. The sub-inspectors will then start to work. In this paper, our goal is to *minimize the number of inspection steps conducted by the sub-inspectors* for identifying all malicious users, which is formulated as the Malicious Meter Inspection (MMI) problem in paper [17]. Note that in this paper, the terms 'user' and 'meter' are interchangeable. Let m denote the number of malicious users in the NAN. We assume the minimum upper bound of m, which is denoted as λ, can be obtained and this is shown in a later section. Obviously, we have $0 \leq m \leq \lambda \leq n$. Since electricity theft is punished by fines and/or incarceration [41], [42] and most electricity theft related attacks can be successfully launched by a single user, malicious users usually do not collude with each other. This implies that few malicious users commit electricity theft simultaneously and that the inspection process usually does not last long. Therefore, in this paper, we also assume that the malicious users will not collude with each other, and it is reasonable for us to assume that there are no new malicious users appearing during the inspection process.

In this paper, we assume that once malicious users are located, utility companies disconnect their power accounts immediately and do not restore electricity until malicious users finish paying the whole balance. This assumption is rationale and consistent with the situation in the real world [43].

The main notations in this paper are listed in Table I. In general, we use the lowercase letters to denote variables and the uppercase letters to notate sets.

IV. Adaptive Binary Splitting Inspection

In this section, we demonstrate the working strategy of the Adaptive Binary Splitting Inspection (ABSI) algorithm. We assume that the minimum upper bound λ is previously known. Actually, under some reasonable assumptions, λ can be estimated, as shown in a later section.

A. Algorithm Description

We first demonstrate how the inspectors judge whether there are malicious users among the users being monitored by them. Considering that technical losses usually cannot be obtained accurately in the real world, we define a threshold, notated by ω, to help judge whether there are reading anomalies among the users being monitored. Specifically, if $x_i \geq \omega, i \in I$, the inspector i can infer that there exist malicious users in G_i. Taking the head inspector as an example, if and only if it finds out $x_0 > \omega$, it detects reading anomalies. With regard to the sub-inspectors, during the inspection process of finding all malicious users, their working strategies are presented as follows: (1) For any sub-inspector $i \in I \setminus \{0\}$, if and only if $x_i > \omega$ and there is only one user in G_i, this unique user will be identified as being malicious; (2) In contrast, for any sub-inspector $i \in I \setminus \{0\}$, if $x_i \leq \omega$, all users in G_i will be declared as being honest, regardless of the number of users being contained in G_i; (3) Specially, for the cases where $x_i > \omega$ and G_i contains multiple users, we can only conclude that there is at least one malicious user in G_i. In this case, the status of any user in G_i cannot be determined immediately, and more inspection steps need to be further conducted.

TABLE I

Notations

Notations	Descriptions
U	We denote by $U = \{1, 2, ..., n\}$ the set of all users in the NAN, with n being the total number of users.
m, λ	m denotes the number of malicious users in the NAN; and λ is the minimum upper bound of m. We have $0 \leq m \leq \lambda \leq n$.
I	We denote by $I = \{0, 1, 2, ..., k\}$ the set of all inspectors, with k being the total number of sub-inspectors. Specifically, inspector 0 stands for the head inspector, and inspectors $1, ..., k$ refer to the sub-inspectors.
q_j	The reported readings of user $j \in U$.
G_i	The set of users monitored by inspector $i \in I$. Specifically, we have $G_0 = U$ and $G_i \subset U, \forall i \in I \setminus \{0\}$, where "$\setminus$" denotes the set difference operation.
r_i	The total amount of electricity distributed to the users in $G_i, \forall i \in I$.
x_i	The total amount of stolen electricity of the users in $G_i, \forall i \in I$.
δ_i	The total amount of technical losses of the users in $G_i, \forall i \in I$.
ω	Since users' technical losses usually cannot be obtained accurately in the application, we define a threshold, denoted by ω, to help judge whether there are malicious users in G_i. Specifically, if $x_i \geq \omega$, it can be inferred that there exist malicious users in G_i.
W	The set of users whose status ("honest" or "malicious") has not yet been determined.
M	The set of users which have already been identified as being malicious.
H	The set of users which have already been identified as honest. Note that we have $U = W \cup M \cup H$.

Let W denote the set of users whose status ("honest" or "malicious") has not yet been determined. These users need to be further inspected in the following inspection process. Let M denote the set of users which have already been identified as being malicious. Let H denote the set of users which have already been identified as honest. Obviously, we have: (1) $U = W \cup M \cup H$; (2) $W \cap M = \emptyset$; (3) $W \cap H = \emptyset$; and (4) $M \cap H = \emptyset$. Note that the sets W, M, and H dynamically change as the inspection proceeds. At the beginning of inspection process, we initialize $W = U$, $M = \emptyset$ and $H = \emptyset$. When the inspection process is finished, we have $W = \emptyset$, and $U = M \cup H$.

Since there are a total number of at most λ malicious users in the NAN and the users in M are malicious, we can infer that during the inspection process, the maximum number of malicious users in W that remain to be identified is $\lambda - |M|$, where "$|\cdot|$" denotes the cardinality of a set. The basic idea of the ABSI algorithm can be put as follows. Among the users whose status has not been determined, if on average one user out of at least two users is malicious, i.e., $|W| \geq 2(\lambda - |M|) - 1$, the binary search method (whose working strategy will be explained later) will be applied to locate the malicious users; otherwise, the users in W will be inspected one by one, which

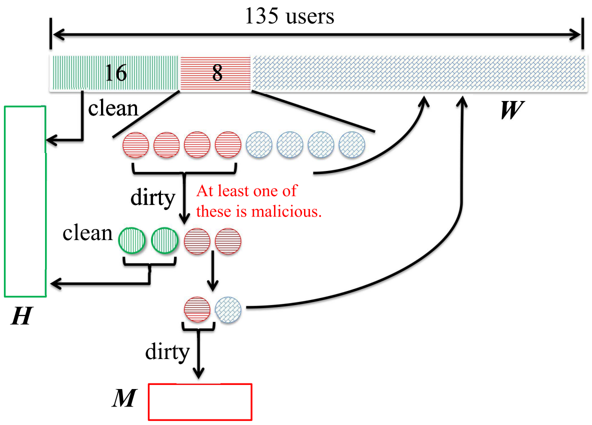


Fig. 2. An example to illustrate the ABSI algorithm. If all the users being inspected are honest, this inspection is referred to as "clean;" otherwise, it is "dirty." Notably, H denotes the set of users which have already determined as honest; M denotes the set of users which have already determined as being malicious; W denotes the set of users whose status has not yet been identified.

is referred to as the scanning method in the paper [17]. The features of the ABSI algorithm mainly lie in the following aspects: (1) During the inspection process, instead of simply sticking with one inspection strategy, the sub-inspectors adaptively adjust their inspection strategies, by dynamically choosing either the scanning method or the binary search method; (2) When the binary search method is chosen as the inspection strategy, the sub-inspectors extract a specific number of users from set W for inspection, rather than simply inspect all users in W as a whole,;

We now briefly introduce the scanning method. If it is chosen as the inspection strategy, one user will be extracted from set W. If sub-inspector $i \in I \setminus \{0\}$ finds this user is malicious, he/she will be put into the set M; otherwise, this user will be put into the set H.

In contrast, when the binary search method is chosen as the inspection strategy, the total number of users to be extracted from W is 2^α, where

$$\alpha = \lfloor \log_2 \frac{|W| - (\lambda - |M|) + 1}{\lambda - |M|} \rfloor. \tag{2}$$

For example, in Fig. 2, we assume that among a total number of 135 users, which are denoted as $U = \{1, 2, \cdots, 135\}$, there are at most $\lambda = 8$ malicious users. We define "a round of scanning" as one inspection step conducted by the scanning method. When we say "a round of binary search", it means several inspection steps which are conducted by the binary search method. Specifically, "a round of inspection" means either "a round of scanning" or "a round of binary search". For the first round of inspection, since $W = U$ and $M = \emptyset$, we can derive $|W| = 135 > 2(\lambda - |M|) - 1 = 15$. Hence, the binary search method will be applied. Based upon Equation (2), we have $\alpha = 4$. This means that 16 users, denoted as $\{1, 2, \cdots, 16\}$, will be extracted from U for the first round of binary search.

We next demonstrate how the binary search method proceeds. Assume that it is sub-inspector $i, i \in I \setminus \{0\}$ conducting the inspection. Then, when a round of binary search begins, we have $|G_i| = 2^\alpha$. As aforementioned, if $x_i \leq \omega$, all the users in G_i are honest; therefore these users will be added into the honest user set H. In this situation, this round of binary search is terminated, with only one inspection step conducted. On the other hand, for the case $x_i > \omega, i \in I \setminus \{0\}$, a total number of α inspection steps will be successively conducted in this round of binary search, after which one user will be identified as being malicious. For each such inspection step, the users in G_i will be divided into two halves. Let G_i' and G_i'' denote the first half and the second half of G_i, respectively. Let x_i' denote the amount of stolen electricity of the users in G_i'. After the sub-inspector i conducts the inspection step on the users in G_i', the value of x_i' can be obtained. If $x_i' > \omega$, the sub-inspector i will further split the users in G_i' into two halves, and the users in G_i'' will be put back into the set W. Otherwise, if $x_i' \leq \omega$ the users in G_i' will be determined as honest; and the users in G_i'' will be split into two halves. The sub-inspector i will then conduct an inspection on the first half of users in G_i' (or G_i''). The above procedure will be conducted for α times, until a malicious user is identified.

For example, in Fig. 2, we assume that all the 16 users being inspected at the first round of inspection are honest. Then, this round of binary search will be terminated after the first inspection. Thus, for the second round of inspection, we have $W = \{17, 18, \cdots, 135\}$, and the malicious user set still remains as $M = \emptyset$. It can be easily inferred that the binary search method will still be adopted in the second round of inspection. Since $\alpha = \lfloor \log_2 \frac{119-8+1}{8} \rfloor = 3$, we can know that 8 users will be extracted from user set U for the second round of binary search. Assume that we have $G_i = \{17, 18, \ldots, 24\}$ and $x_i > \omega$, these eight users will be then divided into $G_i' = \{17, 18, 19, 20\}$ and $G_i'' = \{21, 22, 23, 24\}$. We will then conduct an inspection step on the users in G_i'. Assume $x_i' > \omega$. The users in G_i' will be further divided into two halves, and the users in G_i'' will be put back with the users in W. After two more inspection steps, a user is identified as being malicious. To this moment, the second round of binary search is finished. In summary, a total number of four inspection steps are conducted during the second round of binary search. Note that in Fig. 2, if $x_i \leq \omega$, this inspection is referred to as "clean;" otherwise, it is "dirty."

To conclude, for a round of binary search with 2^α users, if all the 2^α users are honest, one inspection step will be performed; otherwise, $\alpha+1$ inspection steps will be conducted.

Clearly, if the head inspector cannot detect reading anomalies anymore, all malicious users are located and the inspection process can be terminated. With the cooperation of the head inspector, after all malicious users are located, the sub-inspectors can avoid useless inspection steps on the users whose statuses have not been determined but are actually honest.

The above strategies are concluded in **Algorithm** 1, where the lines 1 ~ 7 describe the basic idea of the ABSI algorithm. That is to say, if among the users which need to be further inspected, one user out of an average number of at least two users is malicious, the binary search approach is applied; otherwise, the scanning method will be exploited. As we have mentioned, during the inspection, the ABSI algorithm will not persist in either of the above two inspection strategies, but

Algorithm 1 Adaptive Binary Splitting Inspection (ABSI) Algorithm

```
Require: $W$
Ensure: $M, H$
Initialization: $W \leftarrow U, M \leftarrow \emptyset, H \leftarrow \emptyset$ {$M$ and $H$ are global}
ABSI ($W$):
 1: while $x_0 > \omega$ do
 2:   if $|W| \geq 2(\lambda - |M|) - 1$ then
 3:     BinarySearch($W$);
 4:   else
 5:     Scan($W$);
 6:   end if
 7: end while{end ABSI}
Scan ($W$):
 8: $G_i, W \leftarrow$ extractUsers($W$, 1); {extract 1 user from $W$ to $G_i$}
 9: if $x_i > \omega$ then
10:   $M \leftarrow M \cup G_i$;
11: else
12:   $H \leftarrow H \cup G_i$;
13: end if
14: ABSI($W$); {end Scan}
BinarySearch ($W$):
15: $\alpha \leftarrow \lfloor \log_2 \frac{|W|-(\lambda-|M|)+1}{\lambda-|M|} \rfloor$;
16: $G_i, W \leftarrow$ extractUsers($W, 2^\alpha$);
17: The sub-inspector $i$ conduct one inspection step to obtain $x_i$;
18: if $x_i > \omega$ then
19:   $k \leftarrow 0$;
20:   while $k \leq \alpha$ do
21:     if $|G_i| == 1$ then
22:       $M \leftarrow M \cup G_i$; break;
23:     else
24:       $G_i', G_i'' \leftarrow$ extractUsers($G_i, \frac{|G_i|}{2}$);
25:       The sub-inspector $i$ conduct one inspection step to obtain $x_i'$;
26:       if $x_i' > \omega$ then
27:         $G_i \leftarrow G_i'$; $W \leftarrow W \cup G_i''$;
28:       else
29:         $G_i \leftarrow G_i''$; $H \leftarrow H \cup G_i'$;
30:       end if
31:       $k++$;
32:     end if
33:   end while
34: else
35:   $H \leftarrow H \cup G_i$;
36: end if
37: ABSI($W$); {end BinarySearch}
```

will adaptively adjust the inspection strategy based on the relationship between $|W|$ and $\lambda - |M|$. The lines 8 $\sim$ 14 describe the scanning method; and the lines 15 $\sim$ 37 explain the binary search method. The function extractUsers(W, a) extracts a users from W, and returns two user sets - the first one containing the a users and the second one being the updated W.

B. Estimation of λ

As stated in [44], Binomial trials are a series of repeated independent trials which meet the following conditions: (1) There are only two possible outcomes for each trial; (2) The probability of a specific outcome remains the same throughout the trials; (3) The outcome of one trial does not affect the outcome of other trials.

Under the following two assumptions: 1) the probabilities of stealing energy of all users are the same; 2) all users are independent, all of the above conditions hold in our cases as follows: (1) For a specific user, he/she is either malicious or honest; (2) On the other hand, although the precise number of electricity thieves is one of the most difficult statistics to track, we can usually estimate the ratio of malicious users in the real world. For example, it was reported that 1% of users were stealing power in 1984 in the USA [45]. Let $p, 0 \leq p \leq 1$ denote the ratio of malicious users in the NAN. Then, we can infer that the probability of a user committing electricity theft is p under the assumption that the probabilities of stealing energy of all users are the same; and the probability of a user honestly reporting electricity consumptions is $1 - p$; (3) Additionally, we assume that users do not collude with each other to steal electricity.

Thus, each user in the NAN can be regarded as a Binomial trial with one of the following two possible outcomes: "malicious" or "honest".

We are interested in the number of malicious users in the NAN. According to the definition in [46], a random variable is a function S that assigns a rule of correspondence for every point v in the sample space V called the *domain*, a unique value $S(v)$ on the real line R called the range. Let us define the random variable $S : \{\text{"}malicious\text{"}, \text{"}honest\text{"}\}^n \rightarrow \{0, 1, 2, \cdots, n\}$, which returns the number of malicious users, denoted as m for consistency, in the NAN, i.e., $m \in \{0, 1, \cdots, n\}$. The probability that there are at most λ malicious users is the probability measure of the set of outcomes $\{v \in \{\text{"}malicious\text{"}, \text{"}honest\text{"}\}^n, S(v) \leq \lambda\}$, denoted as $\Pr(S \leq \lambda)$.

Theorem 1: Let us define the random variable $S : \{\text{"}malicious\text{"}, \text{"}honest\text{"}\}^n \rightarrow \{0, 1, 2, \cdots, n\}$, *where n is the total number of users and S returns the actual number of malicious users in the NAN. Let* $p, 0 \leq p \leq 1$, *denote the ratio of malicious users. Assume that the random variable S follows the Binomial distribution with parameters n and p, i.e.,* $S \sim B(n, p)$. *Then we can find a positive integer number* λ *so that* $\Pr\{S \leq \lambda\} \geq 1 - \epsilon$, *where* ϵ, $0 < \epsilon < 1$, *is an arbitrarily small constant.*

Proof: As the random number S follows the Binomial distribution with parameters n and p, the probability of at most λ malicious users in the NAN can be expressed as

$$\Pr\{S \leq \lambda\} = \sum_{k=0}^{\lambda} \binom{n}{k} p^k (1 - p)^{n-k}. \quad (3)$$

Since $\Pr\{S \leq \lambda\} = \Pr\{S \leq \lambda - 1\} + p^{\lambda}(1-p)^{n-\lambda} > \Pr\{S \leq \lambda - 1\}$, $\Pr\{S \leq \lambda\}$ increases monotonically with the value of λ. Also, we have $\Pr\{S \leq \lambda\}|_{\lambda=n} = 1$. Thus, by constantly decreasing the value of λ from n to 0, we must be able to find a minimum λ which satisfies $\Pr\{S \leq \lambda\} \geq 1 - \epsilon$. □

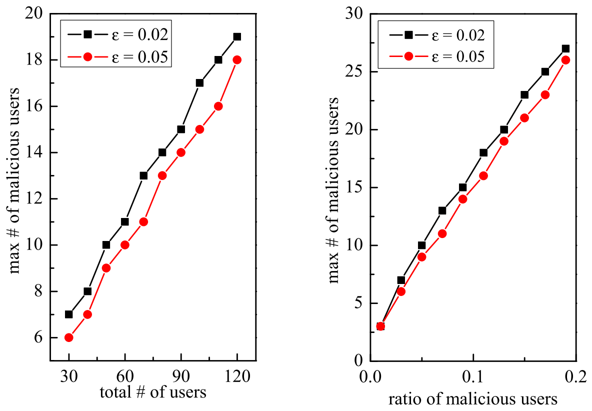
The procedure to find the minimum upper bound λ is presented in **Algorithm** 2. It starts from setting $\lambda = n$, and then calculates the value of $\Pr\{S \leq \lambda\}$ based upon Equation (3). If this probability is larger than $1-\epsilon$, then the value of λ is reduced by a half; otherwise, it is incremented by one. During the increase of λ, if we have $\Pr\{S \leq \lambda\} \geq 1 - \epsilon$, then this λ is what we want. The running time of the procedure of finding the minimum λ is $O(\sqrt{n}) + O(1)$.

Algorithm 2 Finding λ

```
Require: $n, p, \epsilon$
Ensure: the minimum upper bound $\lambda$
Initialization: $\lambda \leftarrow n$
$flag \leftarrow 0$;
while True do
    Calculate the probability $\Pr\{S \leq \lambda\}$ by Equation (3);
    if $\Pr\{S \leq \lambda\} \geq 1 - \epsilon$ then
        $\lambda \leftarrow \lfloor \frac{\lambda}{2} \rfloor$;
        if $flag == 1$ then
            Break;
        end if
    else
        $\lambda \leftarrow \lambda + 1$;
        $flag \leftarrow 1$;
    end if
end while
return $\lambda$
```

In Fig. 3, we assume that the number of malicious users follows the Binomial distribution. In Fig. 3(a), we set $p = 0.1$, and the total number of users n varies from 30 to 120. As we can see, for a given n, a smaller ϵ implies a larger minimum upper bound λ; and for a given ϵ, a larger n yields a larger λ. In Fig. 3(b), we set $n = 100$, and the ratio of malicious users p varies from 0.01 to 0.2. As it can be observed, for a given p, a smaller ϵ corresponds to a larger minimum upper bound λ; and for a given ϵ, a larger p implies a larger λ.

We now explain how we choose parameters ϵ and p in applications. Usually, the parameter ϵ is chosen by utility companies. For estimating λ as accurately as possible in the last subsection, we expect that ϵ is chosen as a small value, i.e., $0 < \epsilon \leq 0.05$. With regard to parameter p, we initialize it in line with some statistics that can be found on the Internet or other places. As aforementioned, it was reported that 1% of users were stealing power in 1984 in the USA [45]. In this case, we can initialize $p = 0.01$. After we apply the ABSI algorithm to pinpoint malicious users for several times, p can be determined as the average ratio of malicious users in the previous rounds of inspection. Assume that the ABSI algorithm has already been executed for a times to locate malicious users. Let m_i denote the number of malicious users located at the i-th time, where $i = 1, 2, \ldots, a$. Then, for the $(a+1)$-th time, we can determine $p = \frac{1}{a}\sum_{i=1}^{a}\frac{m_i}{n}$, where as defined earlier, n is the total number of users in the NAN.



(a)

(b)

Fig. 3. The number of malicious users follows the binomial distribution $B(n, p)$, where n is the total number of users and p is the ratio of malicious users. (a) $p = 0.1$, n varies from 30 to 120; (b) $n = 100$, p varies from 0.01 to 0.2.

V. Performance Analysis

In this section, we provide the performance analyses for the ABSI algorithm. The maximum number of inspection steps is given out. Besides, we analyze how much improvement can be made over the ABSI algorithm, from the perspective of information theory.

Lemma 1: Assume that there exist malicious users in an NAN with a total number of 2^{α} users. Then, with the application of the ABSI algorithm, we can find out one malicious user by at most α inspection steps.

Proof: The proof is conducted with mathematical induction. When $\alpha = 0$, there is only one user in the NAN. Since the head inspector has detected reading anomalies, there is no necessary for the sub-inspectors to do further inspections; and we can directly claim that this user is malicious. In this case, **Lemma** 1 obviously holds. Assume that we can identify one malicious user from 2^{α} users by at most α inspection steps. Then, for the cases where there are $2^{\alpha+1}$ users in the NAN, these users can be divided into two groups, each consisting of 2^{α} users. We now conduct one inspection on one of the above two groups. If the result is "dirty", we can infer that the group being inspected must contain malicious users. Note that in this case, we are not sure whether the group not being inspected contains malicious users or not. According to the assumption, we can know that one malicious user will be found out from these 2^{α} users by at most α inspection steps. On the other hand, if the result is "clean", we can locate one malicious user by conducting at most α inspection steps on the other group that has not yet been inspected. Thus, no matter in which case of the above two cases, we can find out one malicious user from $2^{\alpha+1}$ users by at most $\alpha + 1$ inspection steps. **Lemma 1** holds when there are $2^{\alpha+1}$ users in the NAN. □

From **Lemma** 1, we obtain the following corollary.

Corollary 1: Assume that there are malicious users in an NAN with a total number of n, $n \leq 2^{\alpha'}$, users, where

$\alpha' = \lceil \log_2 n \rceil$. Then, with the application of the ABSI algorithm, we can find out at least one malicious user by α' inspection steps.

Let $S(n,\lambda)$, $\underline{S(n,\lambda)}$, and $\overline{S(n,\lambda)}$ denote the number, the minimum number, and the maximum number, respectively, of inspection steps for the sub-inspectors to find out at most λ malicious users out of n users. With **Lemma** 1 and **Corollary** 1, we can obtain **Theorem** 2 (the proof in the Appendix A):

Theorem 2: Let $\beta = \lfloor \log_2 \frac{n-\lambda+1}{\lambda} \rfloor$ and $g = \lfloor \frac{n+1-(1+2^\beta)\lambda}{2^\beta} \rfloor$. For the ABSI algorithm, we have

$$\overline{S(n,\lambda)} = \begin{cases} n, & \text{if } n < 2\lambda - 1 \\ (\beta+2)\lambda + g - 1, & \text{otherwise.} \end{cases} \tag{4}$$

Next we conduct some analysis based on information theory.

Theorem 3: For any detection system with a meter or meters (acting as an inspector or inspectors) to detection another meter or a group of meters, the minimum number of inspection steps (combined if there are multiple inspectors) to find out at most λ malicious users out of n users is: $\left\lceil \log_2 \sum_{k=0}^{\lambda} \binom{n}{k} \right\rceil$.

Proof: Given a bound of λ malicious users, the total number of combinations of malicious users can be calculated as $\sum_{k=0}^{\lambda} \binom{n}{k}$. By the definition of self-information [47], the amount of information is $\log_2 \sum_{k=0}^{\lambda} \binom{n}{k}$. If the result of one inspection is either "dirty" or "clean", it is one bit of information; if the result of one inspection is inconclusive, it is zero bit information obtained, meaning that in nature the inspection step does not aid in getting fresh information. Therefore, we need at least $\left\lceil \log_2 \sum_{k=0}^{\lambda} \binom{n}{k} \right\rceil$ steps. □

Particulary in the ABSI algorithm, before the sub-inspectors conduct inspections, the head inspector has already detected reading anomalies, and this means that one bit of information has been obtained. Therefore, we can easily have:

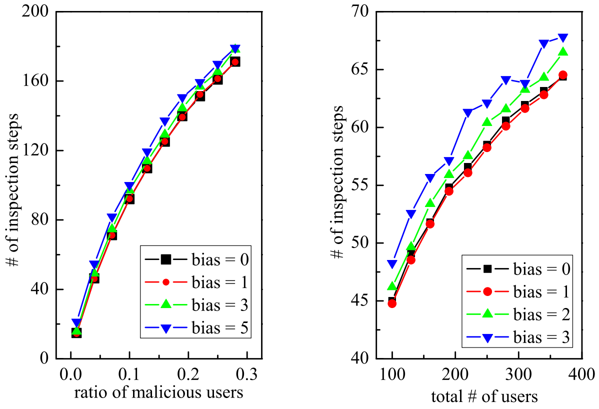
Lemma 2: For the ABSI algorithm, we have

$$\underline{S(n,\lambda)} = \left\lceil \log_2 \sum_{k=0}^{\lambda} \binom{n}{k} \right\rceil - 1. \tag{5}$$

We can know how much improvement can be made over the ABSI algorithm as the following theorem:

Theorem 4: We have $\overline{S(n,\lambda)} - \underline{S(n,\lambda)} \leq \lambda + 1$.

Proof: According to **Theorem** 2, we have $\beta = \lfloor \log_2 \frac{n-\lambda+1}{\lambda} \rfloor$, which is equivalent to $\beta + \xi = \log_2 \frac{n-\lambda+1}{\lambda}$, with $0 \leq \xi < 1$. Thus, we have $\log_2 \frac{n-\lambda+1}{\lambda} < \xi + 1$, from which we can derive $n + 1 - \lambda < \lambda 2^{\beta+1}$. Therefore, we can deduce $g = \lfloor \frac{n+1-\lambda-2^\beta\lambda}{2^\beta} \rfloor < \lambda$. This can be rewritten as $n - \lambda + 1 = 2^\beta\lambda + 2^\beta g + \theta$, with $g < \lambda$ and $\theta < 2^\beta$. Let $f = n - \lambda + 1$. Then, we have $\sum_{k=0}^{\lambda} \binom{n}{k} \geq \binom{n}{\lambda} + \binom{n}{\lambda-1} = \binom{n+1}{\lambda} = \binom{f+\lambda}{\lambda}$. Since $\binom{f+\lambda}{\lambda} = \frac{(f+\lambda)(f+\lambda-1)\cdots(f+1)}{\lambda!} > \frac{(f+1)^\lambda}{\lambda!}$, we can obtain $\sum_{k=0}^{\lambda} \binom{n}{k} > \frac{(2^\beta\lambda+2^\beta g+\theta+1)^\lambda}{\lambda!}$. Due to $\lambda! \leq \lambda^\lambda 2^{1-\lambda}$ and $(1+\frac{g}{\lambda})^\lambda \geq 2^g$ (with the proofs shown in the Appendix B and the Appendix C, respectively), we have: $\sum_{k=0}^{\lambda} \binom{n}{k} > 2^{\beta\lambda+\lambda-1+g}$. By combining Equation (5) and the above inequality, we can obtain $\underline{S(n,\lambda)} > (\beta+1)\lambda + g - 2$. Based upon **Theorem** 2, we can obtain $\overline{S(n,\lambda)} - \underline{S(n,\lambda)} \leq \lambda + 1$. □



(a) (b)

Fig. 4. Evaluation results of the ABSI algorithm. Note that the "bias" means the difference between the estimated upper bound λ and the real m. (a) $n = 200$, m varies. (b) n varies, $m = 10$.

VI. Simulation

In this section, we report simulation results. The simulations are conducted in Python 2.7.13 on an integrated development environment platform - PyCharm Community Edition 2017.1.3. The users' electricity consumption data are generated based on the dataset of individual household electric power consumption in [48], which are measurements of electric power consumption in individual household with a one-minute sampling rate over a period of almost four years.

The simulation setup is stated as follows. Honest users report their electricity consumptions as what they consumed. Reported readings of malicious users are between 10% and 50% of their actual electricity consumptions. According to the data of the World Bank [49], the ratio of the worldwide technical losses to the overall output is between 7% and 10%. Thus, in the simulations, we assume that the ratio of users' technical losses to electricity actually consumed is about 5% to 10%. Users' estimated technical losses are between 0.9 to 1.1 times of their own actual technical losses.

Since the goal of this paper is to identify malicious users within the shortest detection time, we apply the metric of the number of inspection steps conducted by the sub-inspectors to evaluate the performance of the proposed algorithms. Evidently, inspection algorithms with fewer inspection steps are better. Note that each piece of data in the following figures is averaged over 30 times of repeated experiments.

A. ABSI

We first define the term "bias" as the difference between the estimated upper bound λ and the real malicious user number m. In Fig. 4, we investigate how the "bias" influences the performance of the ABSI algorithm. In Fig. 4(a), we assume that there are a total number of $n = 200$ users in the NAN. Considering that in the real world, the ratio of malicious users to the total number of users is usually relatively low, we set this ratio from 0.01 to 0.30. As we can see, on the whole, for a given ratio of malicious users, a smaller bias implies fewer inspection steps. Besides, from the fact that the two curves

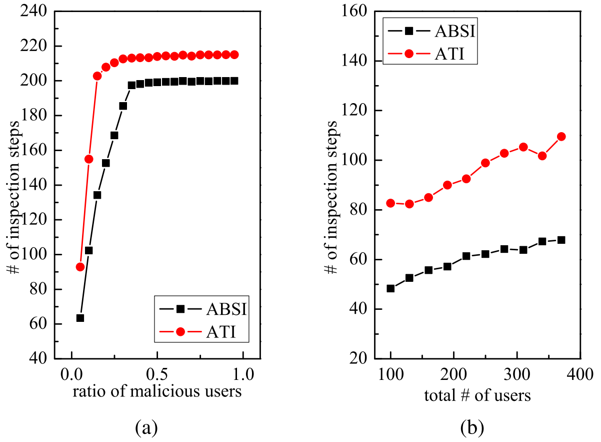


Fig. 5. Evaluation results: ABSI vs. ATI. Note that in Fig. 5(a) and Fig. 5(b), the biases between the estimated upper bound λ and the real malicious user number m are set as 5 and 3, respectively. (a) $n = 200$, m varies. (b) n varies, $m = 10$.

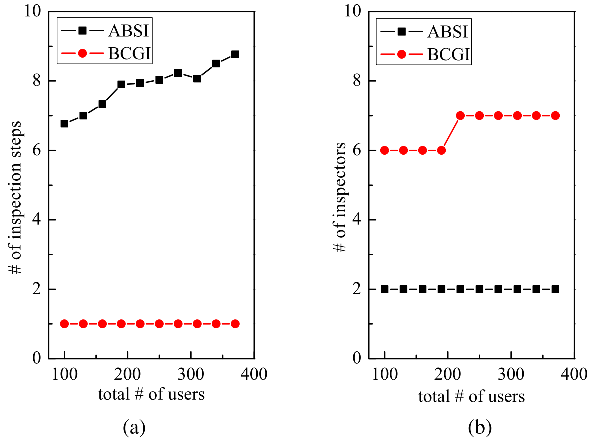


Fig. 6. Evaluation results: ABSI vs. BCGI. (a) # of inspection steps. (b) # of inspectors.

of bias 0 and bias 1 almost coincide with each other, we can conclude that a small bias (for example, bias 1) does not have a great impact on the number of inspection steps of the ABSI algorithm. Furthermore, for any given bias, when the ratio of malicious users is larger, the sub-inspectors need to conduct more inspections to find out all the malicious users.

In Fig. 4(b), the number of malicious users is settled as $m = 10$ and the total number of users n varies from 100 to 400. Fig. 4(b) shows that for a given number of users, the ABSI algorithm has better performance at a smaller bias. In other words, with a smaller bias, the sub-inspectors conduct fewer inspection steps to locate all malicious users in the NAN. In addition, for any given bias, with the increase of the total number of users, the number of inspection steps rises.

B. ABSI vs. ATI

In Fig. 5, we compare the performance of the ABSI algorithm with the ATI algorithm [17], in terms of the number of inspection steps. In Fig. 5(a), we set $n = 200$ and the ratio of malicious users ranges from 0.05 to 1. The bias between the estimated upper bound λ and the real value of m is set as 5. Fig. 5(a) shows that for any given ratio of malicious users, the sub-inspectors conduct fewer inspection steps using the ABSI algorithm than using the ATI algorithm. The maximum number of inspection steps of the ATI algorithm will be closely achieved when the ratio of malicious users is greater than 0.2, whereas that of the ABSI algorithm will be closely achieved when the ratio of malicious users is greater than 0.35. Furthermore, the maximum number of inspection steps of the ATI algorithm is larger than the total number of users in the NAN. In contrast, that of the ABSI algorithm is the same with it.

In Fig. 5(b), the total number of users ranges from 100 to 400 and the number of malicious users is set as $m = 10$. The bias between the upper bound λ and the real value of m is assumed as 3. Fig. 5(b) shows that for a given number of malicious users, the number of inspection steps of the ABSI algorithm is smaller than that of the ATI algorithm, regardless of the total number of users in the NAN.

C. ABSI vs. BCGI

In Fig. 6, we compare the performance of the ABSI algorithm with the BCGI algorithm. Since the BCGI algorithm works only when there is only one malicious user in the NAN, we set $m = 1$. The total number of users in the NAN ranges from 100 to 400. From Fig. 6(a), we can see that for any given n, the sub-inspectors will take more inspection steps using the ABSI algorithm than using the BCGI algorithm. However, from Fig. 6(b), we can observe that the BCGI algorithm utilizes more inspectors than the ABSI algorithm.

Compared to the BCGI algorithm, the greatest advantage of the ABSI algorithm is that it is more general approach. This is because the BCGI algorithm can be applied only when there is a unique malicious user in the NAN [36], [38]. In contrast, the ABSI algorithm can be applied regardless of the number of malicious users in the NAN.

D. Impacts of Threshold ω on Detection Accuracy

In this subsection, we study the impacts of the threshold ω on detection accuracy and false positive/negative rate, mainly through conducting experiments, where detection accuracy is defined as the ratio of the number of malicious and honest users who are identified correctly to the total number of users and false positive/negative rate is defined as the ratio of the number of honest/malicous users incorrectly identified as being malicious/honest to the total number of honest/malicious users.

As aforementioned, in practical applications, users' technical losses usually cannot be accurately estimated (note that the concrete mathematical models for estimating technical losses are out of the scope of this paper), and thus we introduce a threshold ω to guide the inspection process: if $x_i = r_i - \delta_i - \sum_{j \in G_i} q_j > \omega$, there exist malicious users among users in G_i. As defined earlier, r_i denotes the reading of the inspector i that is assumed to be secure, δ_i is the estimated technical loss, and q_j is the reported reading of user j. Let δ_i' denote the true technical loss of users in G_i. Let q_j' denote user j's true electricity consumption. Then, the above inequality can be equivalently rewritten as

$$\sum_{j\in G_i} q'_j - \sum_{j\in G_i} q_j > \delta_i - \delta_i' + \omega \qquad (6)$$

The left side of Inequality (6) is obviously the true amount of stolen electricity of users in G_i. We try to find a threshold which makes the right side of Inequality (6) as close to zero as possible. This is because if we happen to choose a threshold $\omega = \delta_i' - \delta_i$, all malicious users will be finally pinpointed, no matter how small amount of electricity that they steal. But in practice, since we do not know the exact value of δ_i', such a threshold is very difficult to be obtained. If the chosen threshold $\omega < \delta_i' - \delta_i$, Inequality (6) holds regardless whether there are malicious users in G_i or not. In this case, all users (including the actually honest users) will be always identified as being malicious. On the other hand, if the chosen threshold $\omega > \delta_i' - \delta_i$, some actually malicious users may not be located.

Before actually employing the proposed ABSI algorithm to locate malicious users in a specific NAN, we do trial experiments to choose an appropriate ω. The basic idea is simply stated as follows. Assume that there is at least one honest user in the NAN. We initialize to apply the ABSI algorithm to locate malicious users.

- If all the users (including both honest users and malicious users) are identified as being malicious, we increase the value of ω a little bit and then reconduct the inspection on the users in the NAN. The process repeats until not all the users are identified as being malicious. We choose ω as the last value that we have tried.
- On the other hand, if not all the users are identified as being malicious, we decrease the value of ω a little bit and reconduct the inspection until all the users are identified as being malicious. We choose ω as the second value from the last that we have tried.

Note that in both cases of the above trial experiments, we always choose an ω under which not all the users are identified as being malicious. According to the previous analysis, the ω that we choose must satisfy $\omega \geq \delta_i' - \delta_i$. From Inequality (6), we can derive $\sum_{j\in G_i} q'_j - \sum_{j\in G_i} q_j > \delta_i - \delta_i' + \omega \geqslant 0$. This means that users are identified as being malicious only when the difference between their actual and reported electricity consumptions is larger than zero. Apparently, in such cases, honest users will not be mistakenly identified as being malicious, since the difference between their actual and reported electricity consumptions is equal to zero. This implies that *the false positive rate is zero*. On the other hand, for the malicious users whose amount of stolen electricity is equal or less than $\delta_i - \delta_i' + \omega$, the inspectors cannot find out them. This means that we have a nonnegative false negative rate. In application, for thresholds satisfying $\omega \geq \delta_i' - \delta_i$, if we choose a larger ω, we will obviously have more malicious users whose amount of stolen electricity is equal or less than $\delta_i - \delta_i' + \omega$. This implies that with the increase of ω, there are more malicious users mistakenly identified as being honest, which means that the false negative rate rises. This also means that there are fewer malicious users identified correctly. On the other hand, as aforementioned, for thresholds satisfying $\omega \geq \delta_i' - \delta_i$, all honest users will be correctly identified. That is to say, the number of honest users identified correctly stays the same. Thus, we can infer that the detection accuracy declines with the increase of the threshold ω.

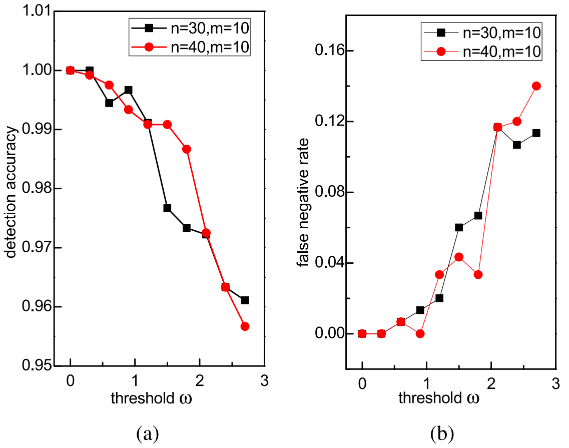


Fig. 7. The impacts of threshold ω on detection accuracy and false negative rate. Note that m and n denote the total number of users and the number of malicious users in the NAN, respectively. (a) Detection accuracy. (b) False negative rate.

In order to study the impacts of the threshold ω on the detection accuracy and the false negative rate, we conduct a simulation as follows. In the simulation, we assume that all the users experience the same technical losses and the utility companies estimate the total technical losses of users in an NAN. The technical losses each user experiences are assumed to be about 1 kWh. The total number of users is set as 30 or 40, respectively. Among these users, we assume that there are 10 malicious users whose amount of stolen electricity is $0.1, 0.2, \cdots, 0.9, 1.0$ kWh, respectively. Fig. 7 shows the detection accuracy and the false negative rate; when the threshold $\omega = 0$, the detection accuracy is 1; as ω increases, the detection accuracy declines gradually and the false negative rate increases gradually.

VII. Conclusion

In this paper, we investigate the MMI problem whose goal is to identify all malicious users with the minimum number of inspection steps. We propose to apply the group testing method to address the MMI problem and we call our method the ABSI algorithm. During the inspection process, the ABSI algorithm adaptively adjusts the inspection strategies. Specifically, among the users which need to be further inspected, if one user out of an average number of at least two users is malicious, the binary search method will be applied; otherwise, the scanning method will be applied. Furthermore, based upon some assumptions, we demonstrate how to estimate the minimum upper bound of the number of malicious users in the NAN, which is the prerequisite for applying the ABSI algorithm. Moreover, we give out the maximum number of inspection steps of the ABSI algorithm. After obtaining the theoretical minimum number of inspection steps, we analyze how much improvement can be made over the ABSI algorithm. Simulation results show that the ABSI algorithm outperforms existing methods in some aspects. Specifically, the ABSI algorithms surpasses the ATI algorithm in terms of the inspection speed. Compared to the BCGI algorithm, the ABSI algorithm is a more general approach.

Appendix A
Proof of Theorem 2

According to the ABSI algorithm, when the inspection process starts, we initiate $W = U$, $M = \emptyset$ and $H = \emptyset$. Therefore, at the beginning, we have $|W| = n$ and $2(\lambda - |M|) - 1 = 2\lambda - 1$. We assume that there are m malicious users in the NAN.

Case 1: We now prove the first part when $n < 2\lambda - 1$. According to the ABSI algorithm, the scanning method will be chosen at the first round of inspection. Let us consider one of the worst case scenarios in which all honest users will be inspected earlier than the malicious users. This scenario is one of the worst cases since no inspection steps can be saved from performing inspections on a group of honest users and the inspection process goes on until the last user to be probed is identified as being malicious. Then, during these rounds of inspections when the honest users are identified, we have $M = \emptyset$ and hence $2(\lambda - |M|) - 1 = 2\lambda - 1$. At the same time, we have $H \neq \emptyset$. Since any two sets among W, M, and H have no intersection and $U = W \cup H \cup M$, we have $|W| = |U - M - H| < n$ and $n < 2\lambda - 1$, Hence, for all these rounds of inspections, we always have $|W| < 2(\lambda - |M|) - 1$, and this means that the scanning method will be applied. Since there are $n - m$ honest users in the NAN, a total number of $n - m$ rounds of scanning will be executed. After these inspections, H will be determined and we have $|H| = n - m$.

After all honest users have been identified, the malicious users will be inspected subsequently. Next, we try to prove that the scanning method will be used instead of the binary method. At the beginning of the $(n - m + j)$-th ($j \geq 1$) round of inspections, we have $0 \leq |M| \leq m - 1$. Thus, $\lambda - |M| - 1 \geq \lambda - m$. Further, we have $|W| = |U - M - H| = n - |M| - (n - m) = m - |M|$. We have $2(\lambda - |M|) - 1 - |W| = (\lambda - m) + (\lambda - |M| - 1) \geq 2(\lambda - m)$. We also have $\lambda \geq m$. Therefore, we have $2(\lambda - |M|) - 1 - |W| \geq 0$. Thus, for all the rounds of inspections, we have $|W| \leq 2(\lambda - |M|) - 1$. If $|W| < 2(\lambda - |M|) - 1$, it means that the scanning strategy will be applied and only one inspection step will be conducted; otherwise we have $|W| = 2(\lambda - |M|) - 1$ and this means that the binary search method will be applied; after substituting $W = 2(\lambda - |M|) - 1$ into Equation (2), we obtain $\alpha = 0$ and this also means that only one user will be inspected, i.e., only one inspection step will be conducted. In general, m times for inspecting the malicious users are needed. To sump up, if $n < 2\lambda - 1$, in the worst case, a total number of n rounds of scanning will be adopted to locate the malicious users.

Case 2: We now focus on proving the second part, i.e., the cases where $n \geq 2\lambda - 1$.

Case 2a: When $2\lambda - 1 \leq n < 3\lambda - 1$. In this case, at the first round of inspection, there is only one user inspected using the binary search method. We prove it as follows: (1) at the beginning, we have $|W| = |U| = n, M = \emptyset$, from which, we can infer $|W| = n \geq 2\lambda - 1 = 2(\lambda - |M|) - 1$. This implies that the sub-inspector will apply the binary search method; (2) Substituting $|W| = n$ and $M = \emptyset$ into Equation (2), we can obtain $\alpha = \lfloor \log_2 \frac{n - \lambda + 1}{\lambda} \rfloor$. Due to $2\lambda - 1 \leq n < 3\lambda - 1$, we can derive $\alpha = 0$. Thus, only one user is inspected for this round of inspection.

We also prove that if at each round from the first to the j-th round of inspections, there is only one honest user identified, and then at the $(j + 1)$-th round of inspection, the sub-inspectors will inspect only one user using the binary search method, with the positive integer j satisfying $1 \leq j \leq n - 2\lambda + 1$. The proof is provided as follows: since only one honest user is identified from the first to the j-th round of inspection, at the beginning of the $(j + 1)$-th round of inspection, we have $|M| = \emptyset$, $|H| = j$ and $|W| = |U| - |M| - |H| = n - j$. Since $j \leq n - 2\lambda + 1$, we can derive $|W| = n - j \geq 2\lambda - 1 = 2(\lambda - |M|) - 1$. This implies that the binary search method will be used. Substituting$|W| = n - j$ and $M = \emptyset$ into Equation (2), we can obtain $\alpha = \lfloor \log_2 \frac{n - j - \lambda + 1}{\lambda} \rfloor$. Due to $1 \leq j \leq n - 2\lambda + 1$ and $2\lambda - 1 \leq n < 3\lambda - 1$, we can derive $0 \leq \alpha < 1$. Since α is an integer number, we have $\alpha = 0$. Thus, the number of users to be inspected is $2^0 = 1$.

Based upon the above analyses, we next prove $\overline{S(n, \lambda)} = (\beta + 2)\lambda + g - 1$ when $2\lambda - 1 \leq n < 3\lambda - 1$. As analyzed in **Case 1**, the maximum number of inspection steps is obtained when all honest users are inspected earlier than malicious users. In the NAN, there are at least $n - \lambda$ honest users. Due to $\lambda \geq 1$, we have $n - \lambda \geq n - 2\lambda + 1$. Thus, there exist the worst cases that at each round of inspection from the first to the $(n - 2\lambda + 1)$-th round, the sub-inspectors identify one honest user. Based upon our former analyses, we can infer that at the $(n - 2\lambda + 2)$-th round of inspection, the binary search method is used and only one user is inspected. No matter whether this user is malicious or honest, when the $(n - 2\lambda + 2)$-th round of inspection ends, we obviously have $|H| + |M| \geq n - 2\lambda + 2$. This means that $|W| = |U| - |M| - |H| \leq 2\lambda - 2 < 2\lambda - 1$. Thus, we can conclude that for the subsequent rounds of inspections, the scanning method is applied. Since only one user is inspected in each round of inspection, the sub-inspectors have to conduct n inspection steps for the worst cases where a malicious user is the last one to be inspected. In conclusion, if $2\lambda - 1 \leq n < 3\lambda - 1$, we have $\overline{S(n, \lambda)} = (\beta + 2)\lambda + g - 1 = n$, with $\beta = 0$ and $g = n - 2\lambda + 1$.

Case 2b: When $n = 2\lambda - 1$. We next prove $\overline{S(n, \lambda)} = (\beta + 2)\lambda + g - 1 = n$. As previously discussed, at the first round of inspection, only one user is inspected using the binary search method. In the worst cases where honest users are identified earlier than malicious users, after the first round of inspection, we have $|H| \geq 1$, $M = \emptyset$ and $|W| = |U| - |M| - |H| \leq n - 1 = 2\lambda - 2 < 2\lambda - 1$. This implies that in the subsequent rounds of inspections, the scanning method is applied to locate honest users. After all the honest users are identified, we have $|W| = |M| = m < \lambda < 2\lambda - 1$, which means that the scanning method is also used. In conclusion, in the case $n = 2\lambda - 1$, the binary search method is employed at the first round of inspection, following $n-1$ rounds of inspection using scanning method. In this case, we also have $\overline{S(n,\lambda)} = (\beta+2)\lambda + g - 1 = n$, with $\beta = 0$ and $g = n - 2\lambda + 1$.

Case 2c: When $n \geq 3\lambda - 1$. We now prove in this case, for any positive integer number $\lambda \geq 1$, the binary search strategy is applied at the first round of inspection as follows. At the beginning of the first round of inspection, since $|W| = n$ and $M = \emptyset$, we can infer $|W| = n \geq 3\lambda - 1 > 2(\lambda - |M|) - 1$, and this means that the binary search strategy is applied.

Case 2c-1: When $n \geq 3\lambda - 1$ and $\lambda = 1$. We next prove in this case, we have $\overline{S(n,\lambda)} = (\beta+2)\lambda+g-1 = n$. Since $\lambda = 1$, we have $\beta = \lfloor \log_2 \ n \rfloor$, which is equivalent to $n = 2^\beta + \gamma$ with $\beta \geq 0$ and $\gamma < 2^\beta$. Substituting $\lambda = 1$, $|W| = n$ and $M = \emptyset$ into Equation (2), we can obtain $\alpha = \lfloor \log_2 \ n \rfloor = \beta$. Thus, the number of users to be inspected is $2^\alpha = 2^\beta$. If the inspection result is "dirty", according to **Lemma** 1, we can locate the unique malicious user by at most β more inspection steps. Otherwise, the problem reduces to finding one malicious user among the remaining γ users. In this situation, according to **Corollary** 1, the unique malicious user will be found out with at most β more inspection step. In conclusion, when $\lambda = 1$, we have $n_s(n,1) = (\beta+2)\lambda + g - 1 = \beta + 1$ with $\lambda = 1$ and $g = 0$.

Case 2c-2: When $n \geq 3\lambda - 1$ and $\lambda \geq 2$. Due to $\alpha = \lfloor \log_2 \frac{n-\lambda+1}{\lambda} \rfloor = \beta$, there will be 2^β users inspected at the first round of inspection.

$$\overline{S(n,\lambda)} = \max(1 + n_s(n - 2^\beta, \lambda), 1 + \beta + n_s(n-1, \lambda-1)). \tag{7}$$

In the max(·) function, the first item corresponds to the case where all the 2^β users that are inspected at the first round of binary search are honest (i.e., the inspection result is "clean"); and the second item corresponds to the case where the result of the first round of inspection is "dirty".

We now consider the case in the first item. Let $n' = n - 2^\beta$ and $\lambda' = \lambda$. Since the equation $g = \lfloor \frac{n+1-(1+2^\beta)\lambda}{2^\beta} \rfloor$ can be rewritten as $n - \lambda + 1 = 2^\beta\lambda + 2^\beta g + \theta$ with $g < \lambda$ and $0 < \theta < 2^\beta$, we have

$$\begin{aligned} n' - \lambda' + 1 &= n - \lambda + 1 - 2^\beta \\ &= \begin{cases} 2^\beta\lambda + 2^\beta(g-1) + \theta, & \text{if } g \geq 1 \\ 2^{\beta-1}\lambda + 2^{\beta-1}(\lambda-2) + \theta, & \text{if } g = 0,\ \theta < 2^{\beta-1} \\ 2^{\beta-1}\lambda + 2^{\beta-1}(\lambda-1) + (\theta - 2^{\beta-1}), & \text{if } g = 0,\ \theta \geq 2^{\beta-1}. \end{cases} \end{aligned}$$

By conclusions in [50] and [51], we have

$$\begin{aligned} n_s&(n - 2^\beta, \lambda) \\ &= \begin{cases} (\beta+2)\lambda + (g-1) - 1, & \text{if } g \geq 1 \\ (\beta+1)\lambda + (\lambda-2) - 1, & \text{if } g = 0,\ \theta < 2^{\beta-1} \\ (\beta+1)\lambda + (\lambda-1) - 1, & \text{if } g = 0,\ \theta \geq 2^{\beta-1}. \end{cases} \end{aligned}$$

Thus, the first item in Equation (7) can be derived as follows:

$$\begin{aligned} 1 + n_s&(n - 2^\beta, \lambda) \\ &= \begin{cases} (\beta+2)\lambda + g - 3, & \text{if } g = 0,\ \theta < 2^{\beta-1} \\ (\beta+2)\lambda + g - 2, & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

On the other hand, for the second item, we let $n'' = n - 1$ and $\lambda'' = \lambda - 1$, we have

$$\begin{aligned} n'' - \lambda'' + 1 &= n - \lambda + 1 \\ &= \begin{cases} 2^\beta(\lambda-1) + 2^\beta(g+1) + \theta, & \text{if } g < \lambda - 2 \\ 2^\beta(\lambda-1) + \theta, & \text{if } g = \lambda - 2 \\ 2^{\beta+1}(\lambda-1) + 2^\beta + \theta, & \text{if } g = \lambda - 2, \end{cases} \end{aligned}$$

Also, by conclusions in [50] and [51], we have

$$\begin{aligned} 1 + n_s&(n-1, \lambda-1) \\ &= \begin{cases} (\beta+2)(\lambda-1) + (g+1) - 1, & \text{if } g \leq \lambda - 3 \\ (\beta+3)(\lambda-1) - 1, & \text{if } \lambda - 2 \leq g \leq \lambda - 1. \end{cases} \end{aligned}$$

Thus, the second item in Equation (7) can be rewritten as

$$\begin{aligned} 1 + \beta + n_s&(n-1, \lambda-1) \\ &= \begin{cases} (\beta+2)\lambda + g - 2, & \text{if } g = \lambda - 1 \\ (\beta+2)\lambda + g - 1, & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

Substituting Equations (8) and (9) to Equation (7), we can obtain $\overline{S(n,\lambda)} = (\beta+2)\lambda + g - 1$.

Appendix B
Proof of Inequality: $\lambda! \leq \lambda^\lambda 2^{1-\lambda}$

Proof: λ is a natural number and $0 \leq \lambda \leq n$. The proof is conducted with mathematical induction. When $\lambda = 0$, The inequality obviously holds. Assume that when $\lambda = k$, with k as an arbitrarily natural number, the inequality holds: i.e., $k! \leq k^k 2^{1-k}$. Since we have $(k+1)^k = \binom{k}{0}k^k + \binom{k}{1}k^{k-1} + \cdots = k^k + k \cdot k^{k-1} + \cdots = 2k^k + \cdots \geq 2k^k$, we can derive $(k+1)! = (k+1)k! \leq (k+1)k^k 2^{1-k} \leq (k+1)\frac{(k+1)^k}{2}2^{1-k} = (k+1)^{k+1}2^{1-(k+1)}$. Therefore, when $\lambda = k+1$, the inequality also holds. □

Appendix C
Proof of Inequality: $(1 + \frac{g}{\lambda})^\lambda \geq 2^g$

Proof: g, λ are both natural numbers and $0 \leq g < \lambda$. As defined in **Theorem** 2, we have $g = \lfloor \frac{n+1-(1+2^\beta)\lambda}{2^\beta} \rfloor$, from which we can derive $g = \lfloor \frac{n+1-\lambda}{2^\beta} - \lambda \rfloor$. According to the definition in **Theorem** 2, we also have $\beta = \lfloor \log_2 \frac{n-\lambda+1}{\lambda} \rfloor$, from which we can derive $\beta \leq \log_2 \frac{n-\lambda+1}{\lambda} < \beta + 1$. This is equivalent to $2^\beta \leq \frac{n-\lambda+1}{\lambda} < 2^{\beta+1}$, which means $\lambda \leq \frac{n-\lambda+1}{2^\beta} < 2\lambda$. Hence, we can derive $0 \leq g \leq \lambda$.

The inequality is equivalent to $2^{\frac{g}{\lambda}} - \frac{g}{\lambda} - 1 \leq 0$. Let $t = \frac{g}{\lambda}$, then we have $0 \leq t \leq 1$. Let $h(t) = 2^t - t - 1, \forall 0 \leq t \leq 1$. Then to prove the inequality is equivalent to prove $h(t) \leq 0$. Let $h'(t)$ and $h''(t)$ denote the first derivative and the second derivative of the function $h(t)$, respectively. Then, we have $h'(t) = 2^t \ln 2 - 1$. Since $h''(t) > 0$, $h(t)$ is a strict and lower convex function with $0 \leq t \leq 1$. Thus, we have $h(t) \leq \max\{h(0), h(1)\} = 0$. □

References

[1] I. Hosni and N. Hamdi, "Distributed cooperative spectrum sensing with wireless sensor network cluster architecture for smart grid communications," *Int. J. Sensor Netw.*, vol. 24, no. 2, pp. 118–124, 2017.

[2] M. Faisal and A. A. Cardenas, "Incomplete clustering of electricity consumption: An empirical analysis with industrial and residential datasets," *Cyber-Phys. Syst.*, vol. 3, nos. 1–4, pp. 42–65, 2017.

[3] S. Ma, Y. Yang, Y. Qian, H. Sharif, and M. Alahmad, "Energy harvesting for wireless sensor networks: Applications and challenges in smart grid," *Int. J. Sensor Netw.*, vol. 21, no. 4, pp. 226–241, 2016.

[4] G. Xu, W. Yu, D. Griffith, N. Golmie, and P. Moulema, "Toward integrating distributed energy resources and storage devices in smart grid," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 192–204, Feb. 2017.

[5] J. Gao, Y. Xiao, J. Liu, W. Liang, and C. L. P. Chen, "A survey of communication/networking in smart grids," *Future Generat. Comput. Syst.*, vol. 28, no. 2, pp. 391–404, 2012.

[6] Z. Ling, K. Liu, Y. Xu, Y. Jin, and X. Fu, "An end-to-end view of IoT security and privacy," in *Proc. IEEE Global Commun. Conf. (Globecom)*, Singapore, Dec. 2017, pp. 1–7.

[7] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, "Security vulnerabilities of Internet of Things: A case study of the smart plug system," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1899–1909, Dec. 2017.

[8] B. Krebs. (2012) *FBI: Smart Meter Hacks Likely to Spread*. [Online]. Available: https://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/

[9] J. Liu, Y. Xiao, S. Li, W. Liang, and C. L. P. Chen, "Cyber security and privacy issues in smart grids," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 981–997, 4th Quart., 2012.

[10] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 717–729, Mar. 2014.

[11] Northeast Group, LLC. (2014). *World Loses $89.3 Billion to Electricity Theft Annually, $58.7 Billion in Emerging Markets*. [Online]. Available: http://www.prnewswire.com/news-releases/world-loses-893-billion-to-electricity-theft-annually-587-billion-in-emerging-markets-300006515.html

[12] J. Mu, W. Song, W. Wang, and B. Zhang, "Self-healing hierarchical architecture for ZigBee network in smart grid application," *Int. J. Sensor Netw.*, vol. 17, no. 2, pp. 130–137, 2015.

[13] Q. Yang, D. An, R. Min, W. Yu, X. Yang, and W. Zhao, "On optimal PMU placement-based defense against data integrity attacks in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1735–1750, Jul. 2017.

[14] T. Yucelen, W. M. Haddad, and E. M. Feron, "Adaptive control architectures for mitigating sensor attacks in cyber-physical systems," *Cyber-Phys. Syst.*, vol. 2, nos. 1–4, pp. 24–52, 2016.

[15] C. Liu, S. Ghosal, Z. Jiang, and S. Sarkar, "An unsupervised anomaly detection approach using energy-based spatiotemporal graphical modeling," *Cyber-Phys. Syst.*, vol. 3, nos. 1–4, pp. 66–102, 2017.

[16] Z. Xiao, Y. Xiao, and D. H. C. Du, "Non-repudiation in neighborhood area networks for smart grid," *IEEE Commun. Mag.*, vol. 51, no. 1, pp. 18–26, Jan. 2013.

[17] Z. Xiao, Y. Xiao, and D. H. C. Du, "Exploring malicious meter inspection in neighborhood area smart grids," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 214–226, Mar. 2013.

[18] X. Xia, W. Liang, Y. Xiao, M. Zheng, and Z. Xiao, "A difference-comparison-based approach for malicious meter inspection in neighborhood area smart grids," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 802–807.

[19] X. Xia, W. Liang, Y. Xiao, and M. Zheng, "Difference-comparison-based malicious meter inspection in neighborhood area networks in smart grid," *Comput. J.*, vol. 60, no. 12, pp. 1852–1870, 2017.

[20] A. H. Nizar, Z. Y. Dong, and Y. Wang, "Power utility nontechnical loss analysis with extreme learning machine method," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 946–955, Aug. 2008.

[21] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and A. M. Mohammad, "Detection of abnormalities and electricity theft using genetic support vector machines," in *Proc. IEEE Region Conf. TENCON*, Nov. 2008, pp. 1–6.

[22] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and M. Mohamad, "Nontechnical loss detection for metered customers in power utility using support vector machines," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 1162–1171, Apr. 2010.

[23] C. C. O. Ramos, A. N. de Sousa, J. P. Papa, and A. X. Falcao, "A new approach for nontechnical losses detection based on optimum-path forest," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 181–189, Feb. 2011.

[24] 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, May 2016.

[25] R. D. Trevizan *et al.*, "Non-technical losses identification using optimum-path forest and state estimation," in *Proc. IEEE Eindhoven PowerTech*, Eindhoven, The Netherlands, Jun./Jul. 2015, pp. 1–6.

[26] J. Jow, Y. Xiao, and W. Han, "A survey of intrusion detection systems in smart grid," *Int. J. Sensor Netw.*, vol. 23, no. 3, pp. 170–186, 2017.

[27] R. Jiang, R. Lu, Y. Wang, J. Luo, C. Shen, and X. Shen, "Energy-theft detection issues for advanced metering infrastructure in smart grid," *Tsinghua Sci. Technol.*, vol. 19, no. 2, pp. 105–120, Apr. 2014.

[28] C.-H. Lo and N. Ansari, "CONSUMER: A novel hybrid intrusion detection system for distribution networks in smart grid," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 33–44, Jun. 2013.

[29] C. J. Bandim *et al.*, "Identification of energy theft and tampered meters using a central observer meter: A mathematical approach," in *Proc. IEEE PES Transmiss. Distrib. Conf. Expo.*, Dallas, TX, USA, Sep. 2003, pp. 163–168.

[30] W. Han and Y. Xiao, "NFD: A practical scheme to detect non-technical loss fraud in smart grid," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 605–609.

[31] W. Han and Y. Xiao, "NFD: Non-technical loss fraud detection in smart grid," *Comput. Secur.*, vol. 65, pp. 187–201, Mar. 2017.

[32] W. Han and Y. Xiao, "CNFD: A novel scheme to detect colluded non-technical loss fraud in smart grid," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl. (WASA)*, Bozeman, MA, USA, Aug. 2016, pp. 47–55.

[33] W. Han and Y. Xiao, "A novel detector to detect colluded non-technical loss frauds in smart grid," *Comput. Netw.*, vol. 117, pp. 19–31, Apr. 2017.

[34] W. Han and Y. Xiao, "FNFD: A fast scheme to detect and verify non-technical loss fraud in smart grid," in *Proc. ACM Int. Workshop Traffic Meas. Cybersecur.*, Xi'an, China, May 2016, pp. 24–34.

[35] W. Han and Y. Xiao, "Design a fast non-technical loss fraud detector for smart grid," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5116–5132, 2016.

[36] X. Xia, W. Liang, Y. Xiao, and M. Zheng, "BCGI: A fast approach to detect malicious meters in neighborhood area smart grid," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 7228–7233.

[37] E. McCary and Y. Xiao, "Malicious device inspection home area network in smart grids," *Int. J. Sensor Netw.*, vol. 25, no. 1, pp. 45–62, 2017.

[38] X. Xia, W. Xiao, Y. Liang, and M. Zheng, "Coded grouping-based inspection algorithms to detect malicious meters in neighborhood area smart grid," *Comput. Secur.*, vol. 77, pp. 547–564, Aug. 2018.

[39] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Statist.*, vol. 14, no. 4, pp. 436–440, 1943, doi: 10:1214/aoms/1177731363.

[40] P. S. N. Rao and R. Deekshit, "Energy loss estimation in distribution feeders," *IEEE Trans. Power Del.*, vol. 21, no. 3, pp. 1092–1100, Jul. 2006.

[41] K. A. Seger and D. J. Icove, "Power theft: The silent crime," *FBI Law Enforcement Bull.*, vol. 57, no. 3, pp. 20–25, Mar. 1988.

[42] (2003). *The Electricity Act*. [Online]. Available: https://www:vakilno1:com/bareacts/theelectricityact/theelectricityact:html#135TheftofElectricity

[43] CarolinaCountry. (2013). *Stealing Electricity—Another Way to Get Electrocuted or Land in Jail*. [Online]. Available: https://www:carolinacountry:com/your-energy/between-thelines/departments/between-the-lines/stealing-electricity

[44] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, 3rd ed. Hoboken, NJ, USA: Wiley, 1968.

[45] W. King. (1984) *Utilities Say 1% of Users are Stealing Power*. [Online]. Available: http://www.nytimes.com/1984/03/26/us/utilities-say-1-of-users-are-stealing-power.html

[46] V. Krishnan, *Probability and Random Processes*. Hoboken, NJ, USA: Wiley, 2006.

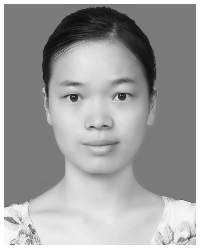
[47] Wikipedia. (2017). *Self-Information*. [Online]. Available: https://en.wikipedia.org/wiki/Self-information

[48] (2012). *UCI Machine Learning Repository, Individual Household Electric Power Consumption Data Set*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Individual+house\\hold+electric+power+consumption/

[49] The World Bank. (2018). *Electric Power Transmission and Distribution Losses (% of Output)*. [Online]. Available: https://data.worldbank.org/indicator/EG.ELC.LOSS.ZS

[50] F. K. Hwang, "A method for detecting all defective members in a population by group testing," *J. Amer. Stat. Assoc.*, vol. 67, no. 339, pp. 605–608, 1972.

[51] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications* (Applied Mathematics) vol. 12, 2nd ed. Singapore: World Scientific, 2000.



Xiaofang Xia received the B.E. degree from Xiangtan University, China, in 2012. She is currently pursuing the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences, China. She was a Visiting Scholar with the Department of Computer Science, The University of Alabama, USA, from 2016 to 2018. Her research interests are mainly in cyber security and smart grid security.

Yang Xiao currently is a Professor with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA. He has published over 200 journal papers and over 200 conference papers. His current research interests include cyber physical systems, internet of things, security, wired/wireless networks, smart grid, and telemedicine. He was a Voting Member of the IEEE 802.11 Working Group from 2001 to 2004, involving the IEEE 802.11 (Wi-Fi) Standardization Work.

Wei Liang received the Ph.D. degree in mechatronic engineering from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2002. She is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences. As a Primary Participant/a Project Leader, she developed the WIA-PA and WIA-FA standards for industrial wireless networks, which are specified by IEC 62601 and IEC 62948, respectively. Her research interests include industrial wireless sensor networks and wireless body area networks. She received the International Electrotechnical Commission 1906 Award in 2015 as a Distinguished Expert of industrial wireless network technology and standard.