IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 9, 2016; accepted December 14, 2016, date of publication December 21, 2016, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2642978

Novel Unsupervised SPITters Detection Scheme by Automatically Solving Unbalanced Situation

KENTAROH TOYODA[1,2], (MEMBER, IEEE), MIRANG PARK[1], NAONOBU OKAZAKI[3], AND TOMOAKI OHTSUKI[2], (SENIOR MEMBER, IEEE)

[1]Kanagawa Institute of Technology, Atsugi 243-0203, Japan
[2]Keio University, Yokohama 223-8522, Japan
[3]University of Miyazaki, Miyazaki 889-2192, Japan

Corresponding author: Kentaroh Toyoda (toyoda@ohtsuki.ics.keio.ac.jp)

ABSTRACT Spam over Internet telephony (SPIT) is recognized as a new threat for voice communication services such as voice over Internet protocol (VoIP). Due to the privacy reason, it is desired to detect SPITters (SPIT callers) in a VoIP service without training data. Although a clustering-based unsupervised SPITters detection scheme has been proposed, it does not work well when the SPITters account for a small fraction of the entire caller. In this paper, we propose an unsupervised SPITters detection scheme by adding artificial SPITters data to solve the unbalanced situation. The key contribution is to propose a novel way to automatically decide how much artificial data should be added. We show that classification performance is improved by means of computer simulation with real and artificial call log data sets.

INDEX TERMS SPIT (spam over internet telephony), unsupervised learning, VoIP (voice over internet protocol), security.

I. INTRODUCTION

SPIT (Spam over Internet Telephony) is recognized as a new threat for voice communication services such as VoIP (Voice over Internet Protocol) [1]. In general, the aim of SPIT is to merchandise products, to make phishing calls, and to take survey by automatically playing recorded voice or speaking by real persons. In the background, there is a fact that inexpensive (even free-of-charge) IP-based telephony services, e.g., Skype, Google Hangouts and Facebook, are getting much popular in recent years and are expected to continue growing until 2020 [2]. Therefore, it is an urgent demand to detect SPIT and/or SPITters (SPIT callers) in the VoIP services. However, there are many challenges in SPIT/SPITters detection. One of the major challenges is that the legitimacy of call contents cannot be judged before a callee takes it. This means that any spam detection schemes in E-mail are typically infeasible to be applied. Hence, one of the feasible detection approaches is that a service provider detects SPITters by inspecting each caller's CDR (Call Detail Records) [3]. By inspecting CDR, several calling features (e.g., call frequency and average call duration) can be calculated for each caller and they are considered to be useful for SPITters detection. For example, because SPITters typically make a large number of calls, it may result in high call frequency. Although such "tendency" can be found, the problem is how to use it for SPITters detection. One possible solution is to set thresholds for each calling feature to judge whether a caller is a SPITter or not (e.g., [3], [4]). However, it is infeasible to do so due to the privacy issue. More specifically, although a service provider must check the calling content to train thresholds, this obviously violates caller's privacy. Not only threshold-based approach but also any supervised techniques (e.g., [5]) that combine multiple features for detecting SPITters cannot still avoid a training phase.

To solve this problem, a clustering-based unsupervised SPITters detection scheme has been proposed [6], [7]. The aim of this scheme is to separate the inspected callers into two clusters, one is the legitimate cluster and the other is the SPITters one by clustering with multiple features. Although these clusters are not labeled, the SPITters' cluster can be identified by comparing the average of any single calling feature (e.g., calls per day) between two clusters. Here, if we let A and B denote two clusters, and the callers in a cluster A call more frequently than ones in the other cluster B, all callers in cluster A are identified as SPITters. Since it only leverages the "tendency" to judge which cluster is SPITters or not, no training data is required.

However, there is a big issue in this scheme that the classification performance degrades when the SPITters account for a small fraction of the entire caller. The root cause of the issue is that it is no longer meaningful to cluster callers into two clusters if most of the inspected callers are legitimate. As described in [7] and [8], the ratio of SPITters could be ranging from 1% to 50% to the entire caller. Hence, a new unsupervised SPITters detection scheme is required so that SPITters can be still identified under such an unbalanced situation. For this purpose, a tentative scheme was proposed to resolve the unbalanced situation by adding artificial SPITters data into the inspected callers dataset [9]. It has been shown that if the ratio of SPITters can be perfectly estimated, the classification performance is improved by adding sufficient artificial SPITters' data. However, as easily guessed, it is a difficult task to estimate the ratio of SPITters. Therefore, it is crucial to propose a novel way to automatically decide the number of added artificial SPITters data without knowing the exact ratio of SPITters in the entire caller.

We propose a novel unsupervised SPITters detection that automatically decides the number of added artificial SPITters data without knowing the ratio of SPITters. The key contribution is that our scheme does not need any estimation of the ratio of SPITters and automatically finds the appropriate number of artificial data, which we denote as N_{added}. We argue that if the appropriate N_{added} is chosen, most of the legitimate callers can be successfully separated from the SPITters. In contrast, if N_{added} is improper, it will result that some of legitimate callers are identified as SPITters. To leverage this fact, we define a scoring function that reflects the goodness of choosing N_{added}. Let us consider an example that a clustering-based classification scheme [6], [7] is repeated ten times. If any caller is identified as legitimate (or a SPITter) ten out of ten, it indicates that there is no need to add artificial data. On the other hand, if any caller is identified as legitimate (or a SPITter) five out of ten, it indicates that the clustering fails mainly because it is in the unbalanced situation. In this case, if we add some artificial data and repeat the above procedure again, any caller's 'classification accuracy' might be improved, say, seven out of ten. Hence, our idea is to quantify the 'goodness of clustering' as a score and to solve as an optimization problem by defining a scoring function. As will be shown later, the appropriate N_{added} can be achieved by our scoring technique in the sense that any caller's classification accuracy is the most improved.

We show the validity of the proposed scheme by means of computer simulation with two real call logs, which are RealityMining [10] and Nodobo [11], and artificial datasets. We show that three classification performance metrics, which are (i) accuracy, (ii) TPR (True Positive Rate), and (iii) FPR (False Positive Rate), are significantly improved irrespective of the ratio of SPITters and outperform the previous schemes [4], [7]. The drawback of our scheme is that computation complexity gets increased. Hence, we also measure the computation time to clarify that the complexity of our scheme is not a big issue in practical use.

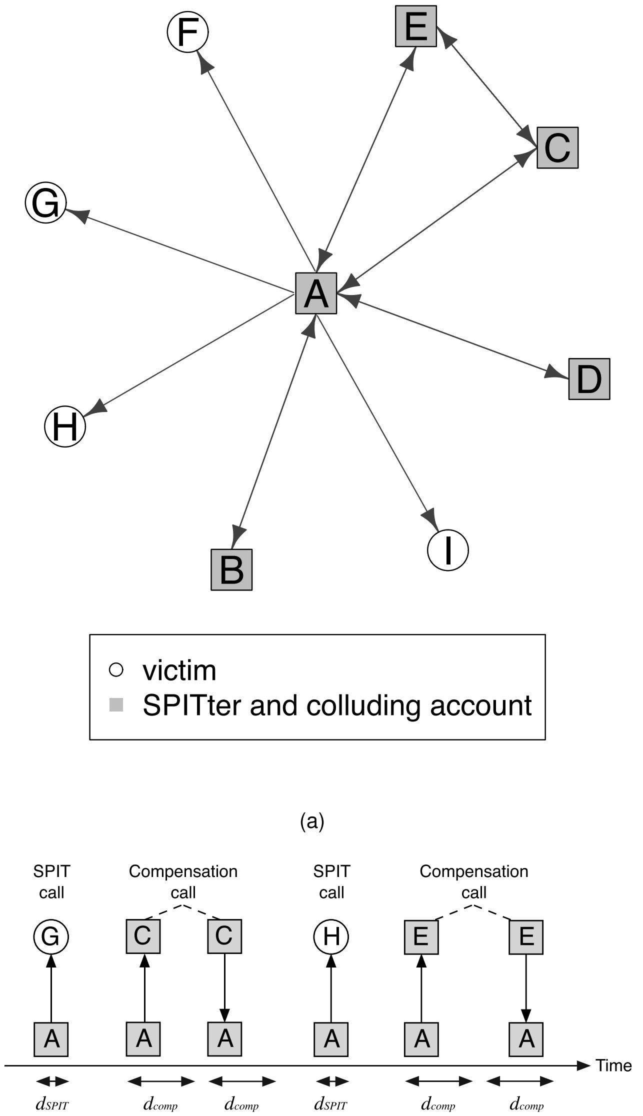
The rest of this paper is structured as follows. Section II describes the preliminaries including the model of SPITters and the system model assumed in this paper. Related work is summarised in Section III. Section IV deals with the procedures of the previous scheme and its drawback. The detailed proposed scheme is described in Section V. Performance evaluation is shown in Section VI. Finally the conclusions are shown in Section VII.

II. PRELIMINARIES

In this section, the SPITters and the system model assumed in this paper are rigorously defined. We first describe the model of SPITters and then the system model. In terms of models, we refer the same models defined in [7].

A. MODEL OF SPITters

We model two types of SPITters which are traditional and sophisticated ones. The traditional SPITters disperse a large number of SPIT calls and call only victims. However, this model can be easily detected by a SPIT detection scheme that identifies high frequent callers as SPITters (e.g., [3]). Hence, the sophisticated SPITters, whose calling behavior is much more like legitimate callers, are also considered. To model the sophisticated SPITters, the previous works assume that the SPITters collude with multiple Sybil accounts [6], [7]. Such SPITters try to reduce the call frequency, compensate for short average call duration, and make more human-like relationships by calling with their colluding Sybil accounts. Figures 1(a) and 1(b) illustrate the model and calling pattern of a SPITter with colluding accounts, respectively. In Fig. 1(a), node A is a sophisticated SPITter and A has four colluding Sybil accounts, namely B, C, D, and E, whereas F, G, H, and I are victims (legitimate users). The arrows indicate the call direction, i.e., a SPITter A and the colluding Sybil accounts call each other whereas the victims may hardly call back to A. From Figures 1(a) and 1(b), it can be seen that a sophisticated SPITter with colluding Sybil accounts can compensate for short average call duration by occasionally calling back with colluding accounts for a certain duration. In addition, this sophisticated model breaks the indication that most of legitimate callers typically call to top five friends [4]. By preparing more than five colluding accounts, a SPITter can imitate the call behavior of legitimate callers and this is obviously an easy task for SPITters. In addition, low frequent SPITters, say 10 calls/day, are also modeled. This is because there could be a case that a real human makes SPIT calls. We model five call frequency patterns (i.e., 10, 50, 100, 500, and 1,000 calls/day) for both traditional and sophisticated SPITters. To summarize, totally ten models of SPITters are assumed throughout this paper. Table 1 shows the model parameters between traditional and sophisticated SPITters. In this table, d_{SPIT}, d_{comp}, and call back rate denote the average call duration of a SPIT, that of compensation call with colluding accounts, and the rate that a legitimate caller calls back to SPITters, respectively. Although we omit the way to calculate d_{comp}, an interested reader may refer [7] for more detail.



(b)

Fig. 1. Model of a SPITter with colluding accounts. (a) Relationships between a SPITter and colluding accounts. Fig. 1: Model of a SPITter with colluding accounts.

We assume the entities include (i) legitimate callers, (ii) SPITters, and (iii) a voice call service provider. A service provider manages the CDR of their callers (i.e., legitimate callers and SPITters). TABLE 2 shows an example of CDR of a caller obtained at a service provider. The task of the service provider is to identify SPITters from their N_{callers} customers by using CDR. To detect the SPITters, a SPITters detection system is deployed in the service provider. Since SPITters must contract with service providers, the service provider can detect SPITters with their CDR and halt their services to the SPITters. The SPITters detection scheme is executed at regular intervals, say once a day, and any calls are rejected until the next SPIT detection phase if the caller is judged as a SPITter. This simple construction avoids any complicated procedures during the call establishment and thus it hardly cause delay.

TABLE 1. Parameters of the SPITter models.

(a)

PARAMETER	VALUE
# of SPIT calls per day	10, 50, 100, 500, and 1,000
SPIT call duration	$d_{\text{SPIT}} \sim Exponential(\mu_{\text{SPIT}} = 15sec)$
Callees (Victims)	Uniformly chosen from the legitimate callers
Call back rate	0.01

(b)

PARAMETER	VALUE
# of SPIT calls per day	10, 50, 100, 500, and 1,000
SPIT call duration d_{SPIT}	$d_{\text{SPIT}} \sim Exponential(\mu_{\text{SPIT}} = 15sec)$
Callees (Victims)	Uniformly chosen from the legitimate callers
Call back rate	0.01
# of colluding accounts	5
Compensation call duration d_{comp}	$d_{comp} \sim Exponential(\mu_{\text{comp}})$

TABLE 2. An example of CDR of a caller.

DATE[DD/MM/YYYY H:M:S]	CALLER/CALLEE	DIRECTION	DURATION [S]
01/May/2016 21:02:19	sip:eve@bar.com	Outgoing	49
01/May/2016 08:12:56	sip:dave@foo.com	Incoming	55
...	...	...	...
07/May/2016 20:17:52	sip:dave@foo.com	Outgoing	192

B. SYSTEM MODEL

III. RELATED WORK

Many researches have proposed SPITters detection schemes. The research topics can be categorized into four fields, (i) features-based SPITters detection schemes (e.g., [3], [4], [12]–[15]), (ii) SPITters detection schemes based on social network trustworthiness (e.g., [8], [16]–[18]), (iii) content-based SPIT detection schemes (e.g., [19]–[22]), and (iv) the proposal of framework (e.g., [23]–[25]). However, we only summarize the first one and do not deal with the latter three fields, because our work is classified as a feature-based SPITters detection scheme.

To extract distinguishable features for SPITters detection, most of the feature-based schemes use CDR (e.g., [3], [4], [13]–[15]), or the message fields of SIP (Session Initiation Protocol) (e.g., [12], [26]), which is a de-facto standard signalling protocol of VoIP. For example, PMG (Progressive Multi Gray-leveling) is a call frequency based SPIT caller detection scheme [3]. Since SPITters are assumed to make a large number of calls, call frequency could be used to distinguish the SPIT callers from legitimate ones.

Yang *et al.* proposed a supervised decision tree-based SPITters detection scheme [5]. Totally six features are used for the detection, namely (i) the number of callees, (ii) total calls, (iii) failed calls, (iv) canceled calls, (v) completed calls, and (vi) the ratio of number of calls outgoing and incoming. These features are trained with a decision tree based machine learning classifier with labeled training data and the SPITters are detected with the trained classifier.

Several works leverage the fact that a legitimate caller typically makes and receives calls, while a SPITter makes a large number of calls but seldom receives calls, e.g., [12], [13].

Based on this notion, callers are identified as the SPITters if their ratio of answered calls and dialed calls is low [13].

Bokharaei *et al.* suggested that two features, namely ST (Strong Ties property) and WT (Weak Ties property), could detect possible SPITters based on the analysis of a real phone call dataset in North America [4]. On the one hand, ST is defined as the ratio of the total call duration of the top 5 callees to the total call time. Legitimate callers' ST values are typically much higher than SPITters' ones, because legitimate callers may spend most of their talk time with only 4-5 people. On the other hand, WT is defined as the fraction of callees that talk for more than 60 sec. The WT value must be very small for SPIT callers since the content of SPIT is annoying for most people and the call duration of SPITters results in shorter than that of legitimate ones. By using ST and WT, they propose a SPITters detection scheme called LTD (Loose Ties Detection). In this scheme, callers are identified as SPITters when both of their WT and FT values are less than predefined value F.

Sengar *et al.* proposed a SPIT detection scheme based on the multiple calling features [14]. In this scheme, frequent but low call duration callers are detected as SPITters. More specifically, if a caller calls five calls within 15 min, the Mahalanobis distance between the average call duration of the caller and the legitimate callers' model is calculated. Then, if its distance deviates from the pre-trained threshold, the caller is identified as a SPITter. Regarding the legitimate caller's model, it is assumed that its call arrival and call duration obey *Poisson*(180 sec) and *Exponential*(60 sec), respectively.

Wang *et al.* proposed call/receive ratio and normalized call frequency based features CI and F_{CD} which are input into the k-means clustering algorithm [15]. The scheme finds the center mass of a legitimate callers and classifies each caller by comparing the distance between the caller and a common reference model with the trained threshold.

Although many schemes have been proposed, all of the aforementioned schemes require supervised training data to decide thresholds and to train machine learning classifiers. That is, both SPITters' and legitimate callers' features sets must be known and a service provider must check the content of calls for labeling the training data. However, it is infeasible to accomplish this task due to the privacy reason. To solve this issue, an unsupervised SPITters detection scheme that does not need any training data was proposed [6], [7]. In the following section, this scheme will be described.

IV. PREVIOUS SCHEME

The idea of the scheme [6], [7] is, with calling features, to separate the callers into two clusters, i.e., one is legitimate callers' cluster and the other is SPITters' one. In other words, the calling features are used not to directly trap SPITters but to find the dissimilarity among callers. This way avoids the complex threshold tuning and training phase. Although clustering itself does not give us the SPITter cluster, "SPITters cluster" may be able to be identified by comparing the average of a feature, e.g., calls per day, calculated within each cluster. That is, it is good enough to leverage the fact that the call duration of SPITters is relatively short compared to legitimate ones and the call frequency of a SPITter is relatively higher than legitimate ones.

TABLE 3. An example of three callers' feature vectors.

CALLER	f_{ACD}	f_{CPD}	f_{ST}	f_{WT}	f_{IOR}
Alice	119.65	2.86	0.69	0.61	0.72
Bob	104	6.25	0.66	0.52	0.43
Carol	61.17	507.12	0.85	0.02	0.1

A. PROCEDURES

The procedures of this scheme consists of the following three steps, namely (i) calculating calling features, (ii) clustering callers based on calling features, and (iii) identifying the SPITters cluster.

1) *Calculating calling features*
At the first step, multiple calling features are calculated from CDR for each caller. The following features are calculated, namely ACD (Average Call Duration), CPD (Call frequency Per Day), ST, WT, and IOR (Incoming/Outgoing Ratio). A set of these calling features are denoted as a feature vector and represents each caller's call pattern. TABLE 3 shows an example of three callers' feature vectors.
2) *Clustering callers based on calling features*
The feature vectors calculated at the first step are used to cluster callers. By inputting them into a clustering algorithm, each caller is grouped into two clusters. There are two choices for clustering. The first one is RF+PAM (the dissimilarity of RF (Random Forests) [27] + PAM (Partitioning Around Medoids) [28]) that finds the dissimilarities between each caller by RF and inputs it to PAM clustering algorithm. The other one is k-means [29] that finds the dissimilarity by scaled Euclidean distance and inputs it into k-means clustering algorithm.
3) *Identifying the SPITters cluster*
Two clusters are obtained in the second step, which are the SPITters and legitimate callers clusters while we do not know which cluster is SPITters' one. Therefore, it is necessary to identify which cluster is the SPITter cluster. For this, the following simple idea is leveraged: If the callers are successfully clustered, the tendency that SPITters call more frequently than legitimate callers may be observed even though some SPITters are low frequent SPITters. For this reason, the higher f_{CPD} cluster is labelled as the SPITter cluster and all callers in this cluster are identified as SPITters while the callers in the other cluster is identified as legitimate callers. Here, although f_{CPD} is used as the judgement

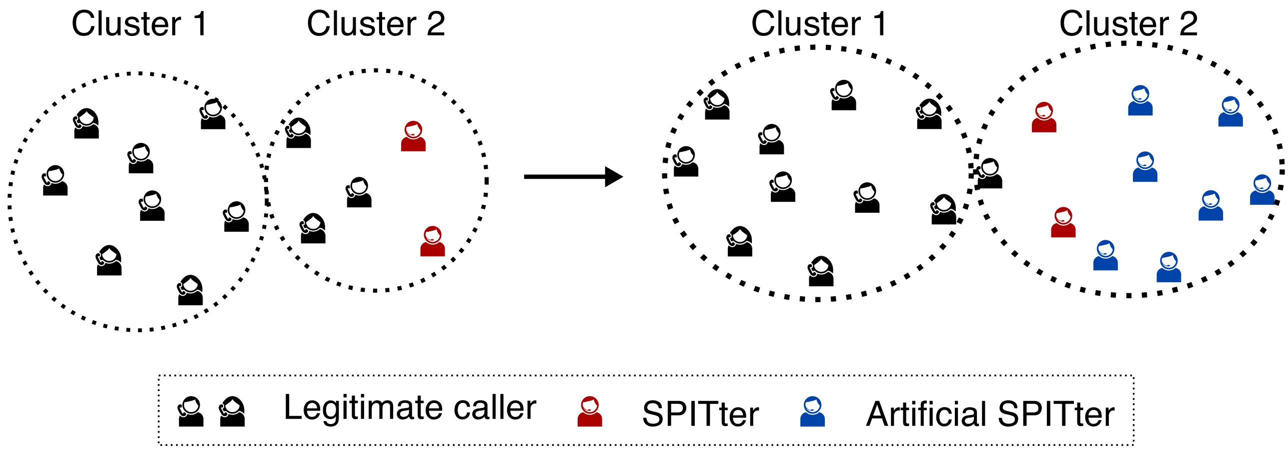


Fig. 2. An example of two-dimensional map of callers. The closeness of each caller denotes the similarity of call patterns. The left figure depicts that if the ratio of SPITters is low, some legitimate callers are wrongly clustered. The right figure represents how this situation is avoided by adding artificial SPITters.

for identifying the SPITter's cluster, other features, i.e., ACD, ST, WT, and IOR, can be used as well [7].

B. UNSOLVED ISSUE AND THE DIRECTION FOR ITS SOLUTION

This scheme can successfully make the SPITters detection scheme unsupervised. However, as also seen in [7], when the ratio of SPITters is low, say 1% or 5%, the scheme does not work well. The root cause of this issue is that it is no longer meaningful to cluster callers into two clusters if most of the inspected callers are legitimate. Based on this observation, we previously proposed an idea to solve this issue [9]. Fig. 2 depicts an example of two dimensional map of callers. In this figure, the closeness of each caller denotes the similarity of call patterns. The idea is that a certain number of artificial SPITters feature vectors are added to solve the unbalanced situation before clustering callers in the second step of the aforementioned procedures. Such a rebalancing technique has been shown to be useful for improving classification performance [30]. More specifically, if the detection system knows (i) the ratio of SPITters in the inspected callers, $\hat{r}_{\text{SPITters}}$, and (ii) the optimal ratio of SPITters in the inspected callers, r_{optimal}, the number of artificial SPITters to be added, N_{added}, can be denoted as the following equation.[1]

$$N_{\text{added}} = N_{\text{callers}} \times (r_{\text{optimal}} - \hat{r}_{\text{SPITters}}). \tag{1}$$

Although the effectiveness of this idea has been verified in [9], there is an unsolved issue. Obviously, both the optimal ratio of SPITters r_{optimal} and the ratio of SPITters $\hat{r}_{\text{SPITters}}$ must be known. However, it is difficult to find the appropriate r_{optimal} for any set of inspected callers. Furthermore, it is also infeasible to know $\hat{r}_{\text{SPITters}}$ in advance since the ratio of SPITters could be ranging from 1% to 50% depending on the case [7], [8]. Therefore, a novel scheme is required to find the optimal N_{added} without knowing both r_{optimal} and $\hat{r}_{\text{SPITters}}$.

[1] Here, 'optimal' ratio denotes the ratio of SPITters where the clustering works best.

V. PROPOSED SCHEME

Here, we propose an unsupervised SPITters detection scheme for the unbalanced case by automatically finding the optimal N_{added}. We argue that if the sufficient number of artificial SPITters' feature vectors are added to the original ones, most of the legitimate callers can be separated from the SPITters. In contrast, if the number of added artificial SPITters' feature vectors is improper, it will result that many legitimate callers are identified as SPITters. To leverage this fact, we define a scoring function that reflects the goodness of choosing N_{added}. Since the ratio of SPITters r_{SPITters} can be very small, e.g., 0.01, in this case, N_{added} might be as big as N_{callers}. Similarly, if r_{SPITters} is very high, e.g., 0.5, no additional SPITters' features is necessary. Hence, the optimal N_{added} should be in $0 \leq N_{\text{added}} \leq N_{\text{callers}}$. To find the optimal N_{added} that maximizes the scoring function, a novel iterative method called bisection method is used [31].

In what follows, we describe (i) the proposed scoring function, (ii) the procedure, and (iii) pros and cons of our scheme.

A. SCORING FUNCTION

We describe the proposed scoring function that reflects the goodness of choosing N_{added}. Alg. 1 describes the proposed scoring function. Notations used in the proposed scheme are listed in TABLE 4. To define the scoring function, we argue that the optimal N_{added} must be obtained when the result of classification is the most stable. From this observation, given N_{added}, the previous scheme [6], [7] is repeated by N_{repeat} times. Then, for each repetition, the classified labels $\mathcal{L}$ are calculated for all inspected callers $i \leq k \leq N_{\text{callers}}$.

$$\mathcal{L}[k] = \begin{cases} +1, & \text{If a caller } k \text{ in the original } \mathcal{F} \text{ is identified as a legitimate caller,} \\ -1, & \text{If a caller } k \text{ in the original } \mathcal{F} \text{ is identified as a SPITter.} \end{cases} \tag{2}$$

$\mathcal{L}[k]$ is accumulated for each caller k as $L[k]$ and the overall score S is calculated by normalizing the summation of $|L[k]|$ by N_{repeat}, where $|x|$ denotes the absolute value of x.

Algorithm 1 s: a Scoring Function That Reflects the Goodness of n

Input: $\mathcal{F}, n$
Output: a score value that reflects the goodness of n
Initialize the accumulated score $S = 0$
Initialize N_{callers} label array L
for i in $[1, N_{\text{repeat}}]$ **do**
  for j in $[1, n]$ **do**
    Randomly choose a SPITter model from ten models of SPITters described in Section II
    Generate a CDR of an artificial SPITter j based on the chosen model
    Calculate artificial SPITter j's feature vector with the generated CDR
  end for
  $\mathcal{F}' \leftarrow$ Merge original feature vectors with the artificial ones
  $\mathcal{L}' \leftarrow$ Input $\mathcal{F}'$ into the previous scheme and obtain $N_{\text{callers}} + n$ classified labels
  $\mathcal{L} \leftarrow$ Omit the classified labels of n artificial feature vectors from $\mathcal{L}'$
  for k in $[1, N_{\text{callers}}]$ **do**
    $L[k] = L[k] + \mathcal{L}[k]$
  end for
end for
for k in $[1, N_{\text{callers}}]$ **do**
  $S = S + |L[k]|$
end for
return S/N_{repeat}

TABLE 4. Notation table.

ITEM	DESCRIPTION
N_{added}	The number of added artificial feature vectors
N_{callers}	The number of total inspected callers (without including artificial SPITters)
N_{repeat}	The number of repetition to calculate the score in $s(\mathcal{F}, N_{\text{added}})$
$\mathcal{F}$	The set of feature vectors of all inspected callers
$\mathcal{F}'$	The set of feature vectors of all inspected callers and artificial SPITters

To understand that this definition is rational, let us consider the two extreme cases: The first one is the most unstable case, i.e., when almost no SPITters exist in the original feature vectors and no artificial SPITters' ones are added ($N_{\text{added}} = 0$). In this case, if we repeat the previous scheme multiple times, say $N_{\text{repeat}} = 10$ times, the clustering does not work well and any caller might be identified as legitimate callers and SPITters for five times each. Hence, the values of $\mathcal{L}[k]$ will take $+1$ and -1 equally and $s(\mathcal{F}, N_{\text{added}})$ will eventually approach 0. In contrast, when we consider the situation where the sufficient number of SPITters exist, if we repeat the previous scheme multiple times, say again 10 times, most of SPITters (or legitimate callers) can be only identified as SPITters (or legitimate callers) by almost 10 times, respectively. Hence, regardless of caller's classes, i.e., SPITters or legitimate callers, the result of $|L[k]|$ will ideally approach 1 and $s(\mathcal{F}, N_{\text{added}})$ will also approach 1.

Since our scoring function returns a value from 0 to 1 according to chosen N_{added}, the remaining task is to quickly find the optimal $n = N_{\text{added}}$ that maximizes $s(\mathcal{F}, n)$, where $0 \leq n \leq N_{\text{callers}}$. Hence, the task can be written as the following optimization problem of $s(\mathcal{F}, n)$.

$$N_{\text{added}} = \underset{0 \leq n \leq N_{\text{callers}}}{\operatorname{argmax}} \; s(\mathcal{F}, n). \tag{3}$$

To solve Eq. (3), we leverage BM (Bisection Method) [31]. BM is used to find the maximum (or minimum) point of a given function of $s(\mathcal{F}, n)$ by iteratively narrowing down the range of n. Alg. 2 represents the algorithm to find the optimal N_{added} based on BM. In this algorithm, $round(x)$ denotes a function that returns the integer of x by rounding. However, strictly speaking, we cannot guarantee that Eq. (3) always returns the optimal N_{added}. This is because it has not been proven that $s(\mathcal{F}, n)$ has only one maximum within $0 \leq n \leq N_{\text{callers}}$. In addition, since our scheme randomly selects the model of SPITters in each repetition, $s(\mathcal{F}, n)$ does not always return the same value against the given feature vector. That is, Eq. 3 may lead to a local maximum and affect classification performance. Although we cannot fully solve this problem, it can be mitigated by specifying appropriate N_{repeat}. We will clarify the relationships between N_{repeat} and classification performance in Section VI.

B. PROCEDURES

By combining the aforementioned idea with the previous scheme, the whole procedure of the proposed scheme can be constructed as following three steps.

1) *Calculating calling features*
The five features, namely ACD, CPD, ST, WT, and IOR, are calculated for each caller in the given CDR and $\mathcal{F}$ is obtained.
2) *Adding optimal number of artificial feature vectors*
The optimal number of artificial SPITters data, N_{added}, is calculated by Alg. 1 and Alg. 2.
3) *Identifying the SPITters with the previous scheme*
N_{added} artificial SPITters' feature vectors are generated and merged with the original $\mathcal{F}$. The merged feature vectors are input into the previous scheme and obtain classified labels for each inspected caller.

C. PROS AND CONS

We discuss the pros and cons of the proposed scheme. The key advantage of our scheme is to realize the unsupervised SPITters detection irrespective of the ratio of SPITters. This is important in practical use since the ratio of SPITters is typically unknown in VoIP services.

The disadvantage is that our scheme requires more calculation than the previous scheme. This is because our scheme iteratively calculates $s(\mathcal{F}, n)$ by $N_{\text{repeat}} \times N_{\text{iter}}$ times to find the optimal N_{added}, where N_{iter} denotes the number of iterations to find N_{added} in Alg. 2. Therefore, a low complexity clustering

Algorithm 2 An Algorithm to Find the Optimal N_{added} Based on BM

```
Input: F, N_callers
Output: N_added
a = 0
b = N_callers
while |a − b| > 1 do
    s_a = s(F, a)
    s_b = s(F, b)
    if s_a > s_b then
        b = round((a + b)/2)
    else
        a = round((a + b)/2)
    end if
end while
return N_added = a
```

TABLE 5. Simulation parameters.

PARAMETER	VALUE
N_{callers}	100
r_{SPITters}	0.01, 0.05, 0.1, 0.2, ..., and 0.5
N_{repeat}	2, 5, 10, 20, 50, and 100
# of trials	100 for each measurement

algorithm is preferred in step 12 of Alg. 1. For this reason, we use k-means as a clustering algorithm since k-means requires only $\mathcal{O}(N_{\text{callers}})$ computation complexity [29]. We will later evaluate the calculation time for the proposed and previous schemes.

VI. PERFORMANCE EVALUATION

In order to show the efficiency of our scheme, we evaluate the classification performance and entire calculation time by means of computer simulation. We evaluate three measures of classification performance, namely accuracy, TPR, and FPR [32]. Accuracy is defined as Eq. (4) and represents the ratio of correctly identified SPITters and legitimate ones to others.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{4}$$

where TP, TN, FP, and FN denote (i) the number of correctly identified SPITters, (ii) correctly identified legitimate callers, (iii) incorrectly identified legitimate callers, (iv) incorrectly identified SPITters, respectively. Similarly, TPR is the ratio of correctly identified SPITters to the total SPITters.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5}$$

FPR is the ratio of legitimate callers mistakenly identified as SPITters.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{6}$$

We have implemented each scheme on a workstation equipped with 2.6 GHz quad-core CPU and 32 GB RAM. The operating system is Ubuntu 16.04 and the codes are written in R language [33]. We compare the proposed scheme with the previous one [7] and LTD [4]. Regarding the proposed and previous schemes, RF+PAM and k-means are evaluated. We merge and use two real call logs datasets called RealityMining [10] and Nodobo [11], which involve 94 and 27 callers' anonymized call logs, as legitimate ones. In contrast, to our knowledge, no real SPITter's call log is available. Hence we generate the SPITters' call logs based on the ten models described in Section II-A. TABLE 5 shows parameters used in the simulation. The total number of inspected callers N_{callers} is fixed to 100. r_{SPITters} and N_{repeat} are varied as specified in the table. Each scheme is repeated by 100 times and the average value is plotted for each performance metric.

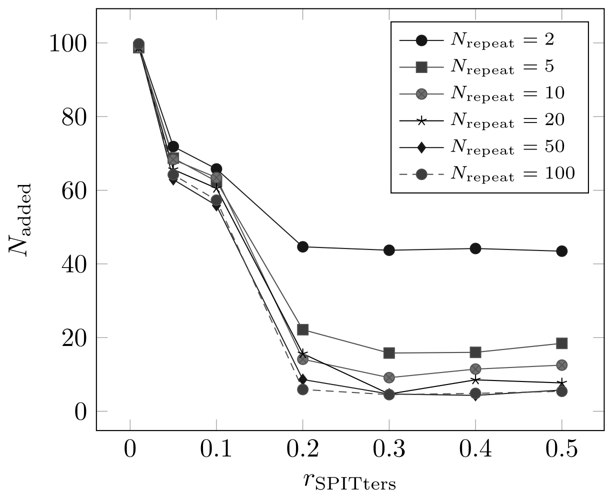
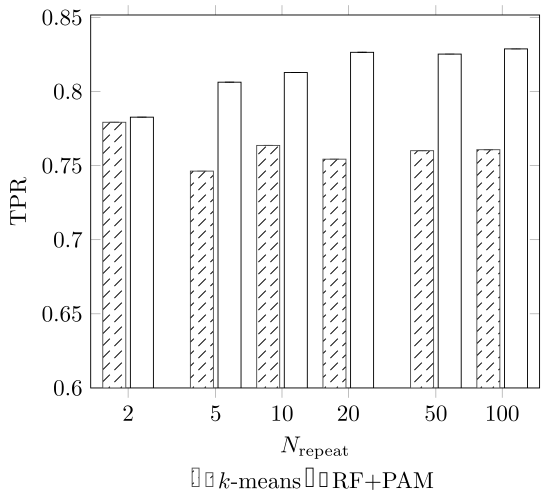


FIGURE 3. N_{added} versus r_{SPITters} by varying N_{repeat}.

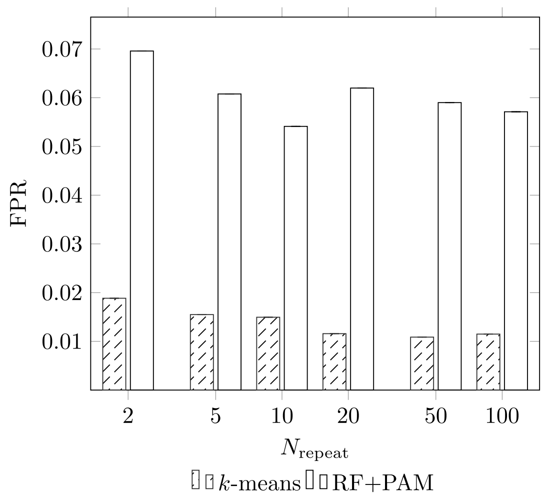
A. CLASSIFICATION PERFORMANCE VERSUS N_{repeat}

Before comparing our scheme with the previous schemes, we show how N_{repeat} affects classification performance and calculation time. Fig. 3 shows N_{added} by varying r_{SPITters}. As can be seen from this figure, when r_{SPITters} is low (i.e., when $r_{\text{SPITters}} < 0.2$), large N_{added} is obtained. For example, when $r_{\text{SPITters}} = 0.01$ which means that only one SPITter exists in the entire caller, $N_{\text{added}} \approx 99$ is obtained. In this case, the number of SPITters and that of legitimate callers become almost same in the clustering phase. In contrast, when $r_{\text{SPITters}} \geq 0.2$, obtained N_{added} is gradually decreased and almost no artificial feature vector are added when $r_{\text{SPITters}} \geq 0.3$ and $N_{\text{repeat}} \geq 50$. This result matches the expectation since when r_{SPITters} is relatively high (i.e., $r_{\text{SPITters}} \geq 0.3$), there is no need to add artificial feature vectors. From this result, it can be said that our automatic selection algorithm of N_{added} works well.

We then discuss how N_{repeat} affects N_{added}. From Fig. 3, when low N_{repeat} is chosen, (i.e., $N_{\text{repeat}} = 2$), the obtained N_{added} is relatively high for r_{SPITters}. This means that $N_{\text{repeat}} = 2$ is not enough to quantify the stability of clustering.



(a)



(b)

Fig. 4. Classification performance versus N_{repeat}. (a) TPR. (b) FPR.

By increasing N_{repeat}, more accurate N_{added} can be obtained and $N_{\text{repeat}} \geq 50$, N_{added} seems to be accurate enough. However, at this point, we cannot conclude that choosing $N_{\text{repeat}} = 50$ is the best. This is because N_{repeat} must be chosen by taking into account classification performance and its computation time. Hence, we then discuss the classification performance. Fig. 4 shows TPR and FPR by varying N_{repeat}. In this evaluation, the mean values of TPR and FPR are obtained by averaging the results when $r_{\text{SPITters}} = 0.01, 0.05, 0.1, \cdots, 0.5$. As can be seen from Fig. 4(a), when k-means is used, TPR is almost unchanged even if large N_{repeat} is chosen. In contrast, when RF+PAM is used, its TPR increases with N_{repeat} and is almost saturated at $N_{\text{repeat}} = 20$. TPR is at best 0.83 and its reason is that the sophisticated SPITters whose call frequency is less than or equal to 10 calls/day are very difficult to be identified. We then discuss the result of FPR. From Fig. 4(b), when $N_{\text{repeat}} = 2$, FPR is much higher than the other choices (i.e., $N_{\text{repeat}} = 5, 10, \cdots, 100$) in both k-means and RF+PAM. This means that $N_{\text{repeat}} > 2$ should be chosen. However, there is no significant difference for $N_{\text{repeat}} \geq 5$ in terms of TPR and FPR. Hence, to choose N_{repeat}, its calculation time may be an important factor. We then see the calculation time by varying N_{repeat}.

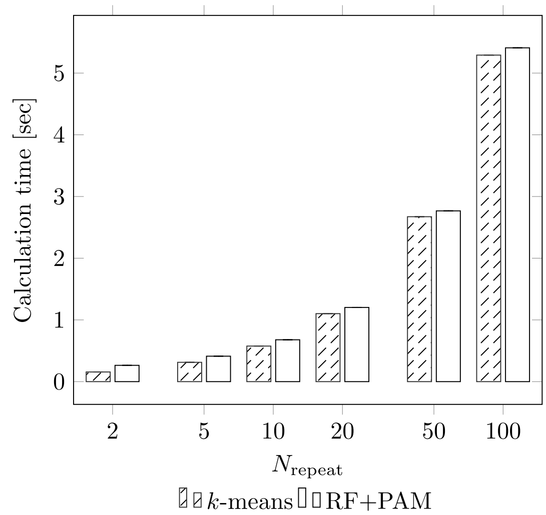


Fig. 5. Calculation time versus N_{repeat}. Note that, as described in Section V-A, k-means algorithm is used to obtain N_{added} even when RF+PAM is used for the classification in the final step.

TABLE 6. Calculation time versus N_{callers}. '-' denotes that the calculation cannot be executed due to the out-of-memory error.

(a)

N_{callers}	ELAPSED TIME [SEC]					
	$N_{\text{repeat}} = 2$	5	10	20	50	100
100	0.22	0.37	0.68	1.1	2.5	5.4
1,000	1.3	1.5	1.9	2.7	5.3	9.4
10,000	78	86	86	88	100	120
100,000	-	-	-	-	-	-
1,000,000	-	-	-	-	-	-

(b)

N_{callers}	ELAPSED TIME [SEC]					
	$N_{\text{repeat}} = 2$	5	10	20	50	100
100	0.18	0.31	0.60	1.0	2.6	5.4
1,000	0.24	0.51	0.92	1.7	4.2	8.7
10,000	0.67	1.5	3.3	6.3	17	35
100,000	5.7	14	30	69	160	320
1,000,000	62	170	320	650	1,600	3,100

(a) RF+PAM
(b) k-means

Fig. 5 shows the calculation time versus N_{repeat}. We measured the elapsed time of the entire procedure described in Section V-B. As can be seen from this figure, the calculation time linearly increases with N_{repeat}. This is because the most time consuming part of our scheme is to calculate k-means clustering algorithm by $N_{\text{repeat}} \times N_{\text{iter}}$ times. When $N_{\text{repeat}} = 5$, its calculation time is less than a second for $N_{\text{callers}} = 100$. Even if we choose $N_{\text{repeat}} = 100$, it takes almost five seconds for $N_{\text{callers}} = 100$. We also evaluate the scalability when N_{callers} increases. TABLE 6 shows the calculation time by varying N_{callers} from 100 to 100,000. In this evaluation, the legitimate callers' call data are randomly duplicated to obtain $N_{\text{callers}} > 100$. From this table, RF+PAM is not scale against N_{callers}. This is because the calculation complexity of PAM clustering algorithm is $\mathcal{O}(N_{\text{callers}}^2)$. Moreover, this also means that PAM clustering requires huge memory and cannot cluster $N_{\text{callers}} = 100,000$ callers. In contrast, k-means is preferable in terms of scalability. For example, when k-means and $N_{\text{repeat}} = 10$ are chosen, it takes about five minutes to classify $N_{\text{callers}} = 1,000,000$ callers. As can be seen from the results when $N_{\text{callers}} = 1,000,000$ and k-means is chosen, the impact of choosing larger N_{repeat} gets bigger. Note that since our scheme is assumed to be executed in an offline manner once a day and thus minute-order (even an hour) calculation time is still acceptable in practical case. To summarize, when a large number of callers exist in a service provider, i.e., $N_{\text{callers}} \geq 1,000,000$, k-means is the only choice for a clustering algorithm and the calculation time linearly increases with chosen N_{callers}.

B. CLASSIFICATION PERFORMANCE VERSUS r_{SPITters}

We finally compare our scheme with the previous scheme [6], [7] and a novel SPITters detection scheme LTD as well [4]. In this evaluation, $N_{\text{repeat}} = 10$ is used for the proposed scheme. For LTD, a threshold parameter $F = 0.9$ is suggested in their paper though it is too tight for our dataset. Hence, we also tested $F = 0.7$.

Fig. 6 shows accuracy, TPR, and FPR versus r_{SPITters}. We first discuss the result of accuracy. From Fig. 6(a), both of the proposed schemes (RF+PAM and k-means) significantly improve accuracy against the previous schemes when $r_{\text{SPITters}} < 0.3$ and achieve almost the same accuracy when $r_{\text{SPITters}} \geq 0.3$. We can also see that the accuracy of the proposed schemes gradually decreases with r_{SPITters}. This is because, as r_{SPITters} gets larger, TP takes larger part in accuracy in the definition in Eq. (4). As seen in Figures 6(b) and 6(c), our scheme achieves very low FPR while TPR is not so much significantly high. Therefore, the low FPR contributes to high accuracy in low r_{SPITters} while the accuracy gradually decreases with r_{SPITters} by TPR. However, our schemes still significantly outperform the previous schemes in terms of accuracy.

From Figures 6(b) and 6(c), both of TPR and FPR of the proposed schemes are relatively consistent against r_{SPITters}. Especially, our scheme with k-means achieves very low FPR which is about 0.01. In contrast, TPR and FPR of the previous schemes get worse when $r_{\text{SPITters}} \leq 0.3$. Our schemes also outperform LTD. This is because LTD does not deal with sophisticated SPITters and suffers from parameter tuning. From this result, we can say that it is difficult to detect sophisticated SPITters only with WT and ST in LTD. In contrast, our scheme does not require any threshold parameters to be tuned. This is a large advantage against the conventional SPITters detection schemes.

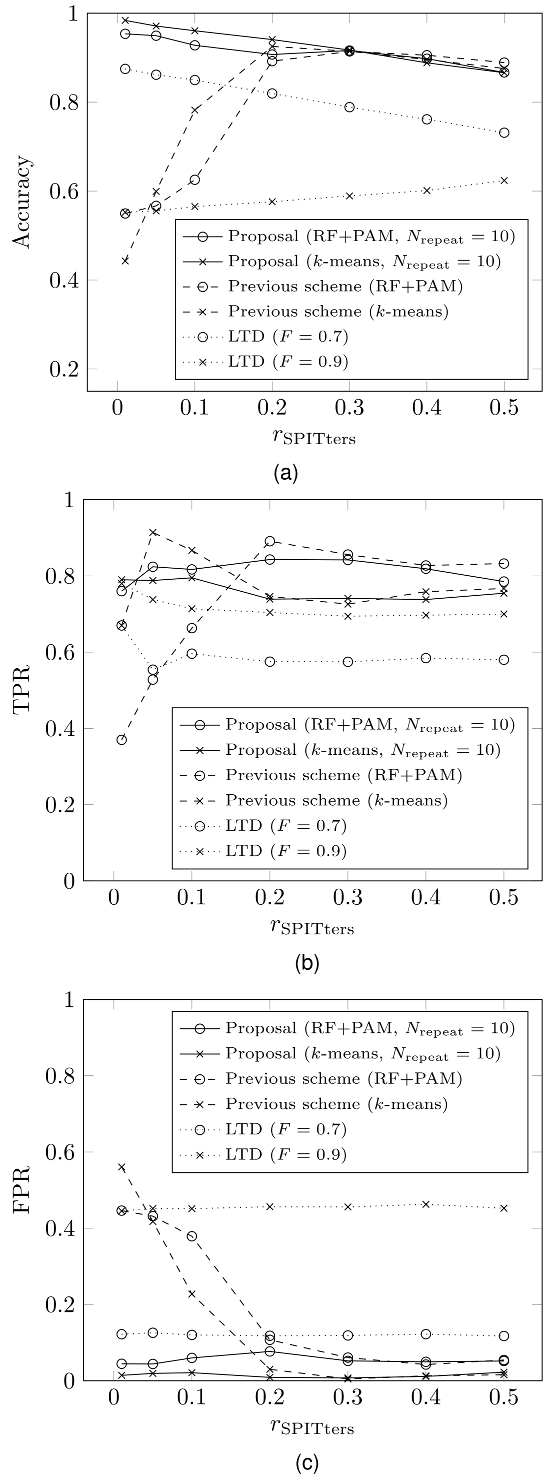


Fig. 6. Classification performance versus r_{SPITters}. (a) Accuracy. (b) TPR. (c) FPR.

VII. CONCLUSIONS

We have proposed an unsupervised SPITters detection scheme that deals with the situation when the SPITters are significantly less than legitimate callers. The idea of our scheme is to add artificial callers' data into inspected ones before clustering callers. By doing this, the unbalanced situation can be solved and the classification performance improves even when r_{SPITters} is low. The novelty of this paper is to automatically decide how much artificial SPITters feature vectors are added. We have proposed a scoring function that quantifies the stability of clustering results and the appropriate number of artificial SPITters feature vectors is calculated by solving the optimization problem against the proposed scoring function. By means of computer simulation, we have shown that our scheme achieves the good classification performance against any possible r_{SPITters} and outperforms the previous schemes. In addition, the number of repetition is not a significant factor for classification performance. Hence, our scheme does not suffer from any parameter tuning issues. Although the drawback of our scheme is to require an additional step to find the optimal number of artificial SPITters feature vectors, our scheme with k-means algorithm can inspect 1, 000, 000 callers within 330 sec when the number of repetition is ten.

REFERENCES

[1] A. D. Keromytis, "A comprehensive survey of voice over IP security research," *IEEE Commun. Surveys Tut.*, vol. 14, no. 2, pp. 514–537, 2nd Quart., 2012.
[2] S. Sudip. *Global VoIP Services Market—Transparency Market Research*. Feb. 2016. [Online]. Available: http://www.transparencymarketresearch.com/pressrelease/voip-services-market.htm
[3] D. Shin, J. Ahn, and C. Shim, "Progressive multi gray-leveling: A voice spam protection algorithm," *IEEE Netw.*, vol. 20, no. 5, pp. 18–24, Sep. 2006.
[4] H. Bokharaei, A. Sahraei, Y. Ganjali, R. Keralapura, and A. Nucci, "You can SPIT, but you can't hide: Spammer identification in telephony networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 41–45.
[5] W. Yang and P. Judge, "VISOR: VoIP security using reputation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2008, pp. 1489–1493.
[6] K. Toyoda and I. Sasase, "SPIT callers detection with unsupervised random forests classifier," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 2068–2072.
[7] K. Toyoda and I. Sasase, "Unsupervised clustering-based SPITters detection scheme," *J. Inf. Process.*, vol. 23, no. 1, pp. 81–92, 2015.
[8] M. A. Azad and R. Morla, "Caller-REP: Detecting unwanted calls with caller social strength," *Comput. Secur.*, vol. 39, pp. 219–236, Nov. 2013.
[9] K. Toyoda, M. Park, and N. Okazaki, "Unsupervised spitters detection scheme for unbalanced callers," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl. Workshops (AINAW)*, Mar. 2016, pp. 64–68.
[10] N. Eagle and A. Pentland, "Reality mining: Sensing complex social systems," *Pers. Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, 2006.
[11] S. Bell, A. McDiarmid, and J. Irvine, "Nodobo: Mobile phone as a software sensor for social network research," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, May 2011, pp. 1–5.
[12] R. MacIntosh and D. Vinokurov, "Detection and mitigation of spam in IP telephony networks using signaling protocol analysis," in *Proc. IEEE/Sarnoff Symp. Adv. Wired Wireless Commun.*, Apr. 2005, pp. 49–52.
[13] Y. Bai, X. Su, and B. Bhargava, "Adaptive voice spam control with user behavior analysis," in *Proc. IEEE Int. Conf. High Perform. Comput. Commun. (HPCC)*, Jun. 2009, pp. 354–361.
[14] H. Sengar, X. Wang, and A. Nichols, "Call behavioral analysis to thwart SPIT attacks on VoIP networks," in *Security and Privacy in Communication Networks*, vol. 96. Berlin, Germany: Springer, 2012, pp. 501–510.
[15] F. Wang, M. Feng, and K. Yan, "Voice Spam detecting technique based on user behavior pattern model," in *Proc. IEEE Int. Conf. Wireless Commun. Netw. Mobile Comput. (WiCOM)*, Sep. 2012, pp. 1–5.
[16] V. A. Balasubramaniyan, A. Mustaque, and P. Haesun, "Callrank: Combating SPIT using call duration, social networks and global reputation," in *Proc. Conf. Email Anti-Spam (CEAS)*, 2007, pp. 1–8.
[17] T. Kusumoto, E. Y. Chen, and M. Itoh, "Using call patterns to detect unwanted communication callers," in *Proc. IEEE/IPSJ Int. Symp. Appl. Internet (SAINT)*, Jul. 2009, pp. 64–70.
[18] N. Chaisamran, T. Okuda, G. Blanc, and S. Yamaguchi, "Trust-based VoIP spam detection based on call duration and human relationships," in *Proc. IEEE/IPSJ Int. Symp. Appl. Internet (SAINT)*, Jul. 2011, pp. 451–456.
[19] J. Quittek, S. Niccolini, S. Tartarelli, M. Stiemerling, M. Brunner, and T. Ewald, "Detecting SPIT calls by checking human communication patterns," in *Proc. IEEE Int. Conf. Commun.*, 2007, pp. 1979–1984.
[20] H. Huang, H.-T. Yu, and X.-L. Feng, "A SPIT detection method using voice activity analysis," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur. (MINES)*, vol. 2. Nov. 2009, pp. 370–373.
[21] D. Lentzen, G. Grutzek, H. Knospe, and C. Porschmann, "Content-based detection and prevention of spam over IP telephony—System design, prototype and first results," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.
[22] J. Strobl, B. Mainka, G. Grutzek, and H. Knospe, "An efficient search method for the content-based identification of telephone-SPAM," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 2623–2627.
[23] M. Falomi, R. Garroppo, and S. Niccolini, "Simulation and optimization of SPIT detection frameworks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2007, pp. 2156–2161.
[24] Y. Soupionis *et al.* (2008). *SPAM Over Internet Telephony Detection Service*. [Online]. Available: http://projectspider.org/documents/Spider_D4.2_public.pdf
[25] J. Quittek, S. Niccolini, S. Tartarelli, and R. Schlegel, "On spam over Internet telephony (SPIT) prevention," *IEEE Commun. Mag.*, vol. 46, no. 8, pp. 80–86, Aug. 2008.
[26] A. Gazdar, Z. Langar, and A. Belghith, "A distributed cooperative detection scheme for SPIT attacks in SIP based systems," in *Proc. Int. Conf. Netw. Future (NOF)*, 2012, pp. 1–5.
[27] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[28] L. Kaufman and P. J. Rousseeuw, "Finding groups in data," in *An Introduction to Cluster Analysis*. vol. 344, Hoboken, NJ, USA: Wiley, 2009.
[29] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1. 1967, pp. 281–297.
[30] J.-H. Xue and P. Hall, "Why does rebalancing class-unbalanced data improve AUC for linear discriminant analysis?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 1109–1112, May 2015.
[31] R. L. Burden and J. D. Faires, "Numerical analysis," Ninth, Eds. Brooks/Cole Publishing Company, 2011.
[32] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *Int. J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011, Bioinfo Publications.
[33] R Core Team, Vienna, Austria. (2016). *R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing*. [Online]. Available: https://www.R-project.org/

KENTAROH TOYODA (M'13) was born in Tokyo, Japan, in 1988. He received the B.E., M.E., and D.E. degrees from Keio University in 2011, 2013, and 2016, respectively. He is currently an Assistant Professor with Keio University. His research interest is security & privacy for systems and services with Internet of Things devices and cryptocurrency. He received the Fujiwara Foundation Award in 2016, the Telecom System Technology Encouragement Award in 2015, and the IEICE communication society encouragement awards in 2012 and 2015, respectively. He is a member of IEICE and IPSJ.

MIRANG PARK received the B.S. and M.S. degrees in electrical engineering from Hanyang University, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree in information science and technology from Tohoku University, Japan, in 1993. He is currently a Professor with the Faculty of Information Technology, Kanagawa Institute of Technology, Japan. She joined the Information Technology Research and Development Center, Mitsubishi Electric Corporation, in 1994, where she has been involved in research and development of network security. Her research interests include security for large deployment of sensor networks, peer-to-peer networks, and user authentication. She is a member of IPSJ, IEICE, and JSSM.

NAONOBU OKAZAKI received the B.E., M.E., and D.E. degrees in electrical and communication engineering from Tohoku University, Japan, in 1986, 1988, and 1992, respectively. He is currently a Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include mobile network and network security. He is a member of IPSJ, IEICE, and IEEJ.

TOMOAKI OHTSUKI (M'91–SM'01) received the B.E., M.E., and Ph.D. degrees from Keio University, Yokohama, Japan, in 1990, 1992, and 1994, respectively, all in electrical engineering. From 1994 to 1995, he was a Post-Doctoral Fellow and a Visiting Researcher in electrical engineering with Keio University. From 1995 to 2005, he was with the Science University of Tokyo. From 1998 to 1999, he was with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley. In 2005, he joined Keio University. He is currently a Professor with Keio University. He has authored over 140 journal papers and 340 international conference papers. He is involved in research on wireless communications, optical communications, signal processing, and information theory. He is a fellow of the IEICE. From 1993 to 1995, he was a Special Researcher of Fellowships of the Japan Society for the Promotion of Science for Japanese Junior Scientists. He was a recipient of the 1997 Inoue Research Award for Young Scientist, the 1997 Hiroshi Ando Memorial Young Engineering Award, the Ericsson Young Scientist Award 2000, the 2002 Funai Information and Science Award for Young Scientist, the IEEE the 1st Asia-Pacific Young Researcher Award 2001, the 5th International Communication Foundation (ICF) Research Award, the 2011 IEEE SPCE Outstanding Service Award, the 28th TELECOM System Technology Award, the ETRI Journals 2012 Best Reviewer Award, and the 9th International Conference on Communications and Networking in China 2014 (CHINACOM '14) Best Paper Award. He served as the Chair of the IEEE Communications Society, and the Signal Processing for Communications and Electronics Technical Committee. He has served as the General-Co-Chair and the Symposium Co-Chair of many conferences, including IEEE GLOBECOM 2008, SPC, IEEE ICC 2011, CTS, and IEEE GLOBECOM 2012, SPC. He gave tutorials and keynote speech at many international conferences, including IEEE VTC and IEEE PIMRC. He is currently serving as a Vice President of Communications Society of the IEICE. He served as a Technical Editor of the *IEEE Wireless Communications Magazine*. He is currently serving an Editor of the IEEE COMMUNICATIONS SURVEYS and TUTORIALS and the Elsevier *Physical Communications*.