Electronic Preprint for Journal of Information Processing Vol.28

Regular Paper

De-identification for Transaction Data Secure against Re-identification Risk Based on Payment Records

Satoshi Ito[1,a)] Reo Harada[1] Hiroaki Kikuchi[1,b)]

Received: November 25, 2019, Accepted: June 1, 2020

Abstract: De-identification is a process to prevent revealing the identity of a person based on personal data of that individual including personal identification information. In conventional de-identification studies, re-identification is a process used to identify individuals from *static* data where there is one record specified for each individual. In contrast, in this paper, we employ *dynamic* data, for example, trajectory data and online payment records. In particular, we consider the open competition data from the 2016 Privacy Workshop Cup (PWS Cup 2016) held in Japan consisting of purchasing history data. Throughout the analysis, we find that attackers can re-identify individuals with a high degree of accuracy from their de-identified purchase history data based on a feature of the set of goods. To address this re-identification risk, we propose a new method to de-identify history data by adding dummy records under certain restrictions. In our method, we use the Jaccard coefficient and the TF-IDF to form user clusters. We evaluate the performance of our proposed method and compare it with the performance of the PWS Cup 2016 participants as an experiment in data privacy. Even in the best de-identified data in PWS Cup 2016, 22.25% of customers were re-identified by our re-identification algorithm based on the Jaccard coefficient. However, only about 12% of customers are re-identified by random re-identification method and about 17% of customers are re-identified by re-identification method based on the Jaccard coefficient in the data that are de-identified by our de-identification method.

Keywords: de-identification, re-identification, k-anonymity, personal identifiable Information

1. Introduction

De-identification is a process to prevent individuals from being identified from datasets containing personally identifiable information (PII). De-identification methods should be carefully selected to efficiently reduce re-identification risks in specific de-identified data. Companies should confirm that the re-identification risks have been reduced sufficiently before transferring big data to their business partners. In Japan, the Act on the Protection of Personal Information fully came into effect in 2015, in which a new concept called "*Anonymously Processed Information* [*1]" was introduced [1]. Due to this revision, a data controller is allowed to provide various services using a data containing PII, free from the risk of re-identification.

However, most conventional de-identification techniques assume the datasets are well *structured*, i.e., data is represented logically in the form of a table. Hence, the dataset for applying the de-identification techniques is limited to within just a small fraction of larger data. For example, ISO/IEC 20889 does not apply to complex datasets, e.g., free-form text, images, audio, or video [2]. Nevertheless, the diversity of datasets dealt with industries increases year by year.

In order to enrich the range of datasets for de-identification, in this paper, we study de-identification of transaction data consisting of multiple records per individuals at the time of certain events. Payment history data for example is widely stored in many business applications so related individual face the risk of being identified as targets for promotions or advertisements. A data competition called, The Privacy Workshop Cup (PWS Cup 2015) [3] was held in 2015 to support development of secure de-identification techniques to apply to more complex data. Even if de-identification of the purchasing history was fully performed to maintain confidentiality, a motivated adversary might be able to acquire the entire dataset and successfully re-identify an individual based on certain features of the payments. We note that the customers in data must have some characteristics on purchasing goods and the sets of purchasing goods are significant enough to allow sufficiently re-identifying an individual. In this paper, we provide re-identification algorithms exploiting the payment features and evaluate the risk of record linkage in these algorithms [*2].

Providing de-identification that is fully resilient against re-identification threats is not easy. The simplest method for de-identification is suppression of records so that no two individuals are distinguishable from one another. Record suppression can cause extreme loss of data utility. A second method is adding some dummy records so as to hide the purchasing characteristics of customers. However unplanned adding of dummy records may cause a loss in data utility if too may dummy records are added. To balance the trade-off between security and util-

[1] Meiji University Graduate School of Advanced Mathematical Sciences, Nakano, Tokyo 164–8525, Japan
[a)] mmhm@meiji.ac.jp
[b)] kikn@meiji.ac.jp

[*1] Japanese version of de-identified information with slightly changes to common anonymized data.
[*2] The primary version of this paper was published in MDAI 2019.

ity, we shall carefully classify the set of customers in a dataset, into some smaller clusters in which customers share common purchasing characteristics so that fewer dummy records are required. However, the conventional clustering algorithms such as the k-means method suffers from the following problems and we shows schematic views of customers (black points) and clusters in **Fig. 1**; (1) (*Monopoly of cluster*) A few huge clusters occupy most of the records. For instance of PWS Cup 2016, the purchase history data contains 38,087 records of unique 2,781 goods suffers in the large cluster with common payment pattern of the most frequent goods. Excessively large cluster such as the orange cluster in Fig. 1 require the addition of many dummy records. (2) (*Too-many Minorities*) Many small (mostly, size of 1) clusters like blue ones in Fig. 1 are formed and most of them are free of dummy record changes. However, the singletons are easy to re-identify. Typical transaction data has similar property, i.e., with many records of small diversity. Hence, it could be skewed and and suffer a loss of utility due to many noisy dummy records. Therefore, we need to develop a new clustering algorithm customized so that all cluster sizes are well balanced.

In this paper, we address the unbalanced issue of clustering in the following ways: (1) A clustering method replacing Term (*good*) Frequency–Inverse Document (*individual*) Frequency (TF-IDF) weights by the frequencies of purchasing goods. With TF-IDF weight, the rare items are weighted higher than common items and hence the monopoly cluster can be weakened. (2) A new algorithm that clusters in limited sizes by applying a minimum clustering size. As far as the threshold size, every cluster grows to a certain size that prevents uniquely identifying it. We evaluate the utility of this de-identification method by the number of dummy records for making clusters.

Only about 12% of customers are re-identifiable by random re-identification method and about 17% of customers are re-identified by a re-identification method based on the Jaccard coefficient in the data that are de-identified by our de-identification method. Even in the best de-identified data in PWS Cup 2016, 22.25% of customers were re-identified by the re-identification algorithm based on the Jaccard coefficient. We assume an attacker who knows a set of purchased goods of all customers as basic background knowledge and tries to re-identify customers by the re-identification method based on the characteristics of the purchased goods set and the Jaccard coefficient.

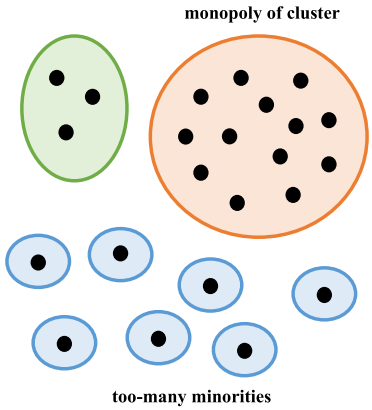

Fig. 1 Schematic views of customers and clusters.

Our study makes three contributions: (1) we propose a new method to de-identify a transaction data making clusters by the Jaccard coefficient and the TF-IDF; (2) we show risks that individuals in a transaction data like Online Retail Data are easily re-identified; (3) we evaluate our de-identification method by re-identification ratio and the number of dummy records to make clusters.

The remainder of the paper is organized as follows. In Section 2, we show the characteristics of the purchase history data and the re-identification risks that are revealed in PWS Cup 2016. In Section 3, we propose a method to de-identify data. In Section 4, we describe some experimental results. In Section 6, we provide a conclusion to this paper.

2. Characteristics of Purchase History Data and Re-identification Risks

2.1 Purchase History Dataset

The Online Retail Data Set [5] comprises the actual purchase history data observed in one year for an online retail shop in the UK and is published at the UCI Machine Leaning Repository [4]. This dataset has been used in PWS Cup 2016–2018.

In this paper, we define the fundamental quantities of the dataset as follows.

Definition 1 Let n, m, and ℓ be the number of customers of the dataset, the number of records, and the number of kind of goods, respectively. Let $U = \{u_1, \ldots, u_n\}$ be a set of customers in the dataset. Let $U' = \{u'_1, \ldots, u'_n\}$ be the set of customers in the de-identified data. Let $I(U) = \{g_1, \ldots, g_\ell\}$ be a set of goods purchased by all customers. Let $I(u_i)$ be a subset of $I(U)$ purchased by customer u_i. Let b be the mean number of goods that a customer purchases in a year. Let $J(u_i, u_j)$ be the Jaccard coefficient between u_i and u_j that is defined by $J(u_i, u_j) = |I(u_i) \cap I(u_j)|/|I(u_i) \cup I(u_j)|$.

We show the summary, the sample records, and the statistics of a subset of the Online Retail Dataset that was made for PWS Cup 2016 in **Tables 1**, **2**, and **3**, respectively. The transaction data contains 7 attributes, 400 users ($n = 400$), 38,087 transactions ($m = 38{,}087$), and 2,781 goods. From observation of these data,

Table 1 Summary of dataset that was used in the competition.

Attribute	Detail
User ID	ID of user (5 digit number)
Receipt ID	ID of receipt (6 digit number)
Date	Purchase date (yyyy/mm/dd)
Time	Purchase time (hh:mm)
Goods	ID of purchased goods (number and character)
Price	Price of purchased goods (Pound sterling)
Number	Quantity of purchased goods (number)

Table 2 Example of dataset that was used in the competition.

User ID	Receipt ID	Date	Time	Goods	Price	Number
12583	536370	2010/12/1	8:45	22728	3.75	24
12583	536370	2010/12/1	8:45	22727	3.75	24
12431	536389	2010/12/1	10:03	22941	8.5	6
12431	536389	2010/12/1	10:03	21622	4.95	8
12431	536389	2010/12/1	10:03	21791	1.25	12
12838	536415	2010/12/1	11:57	22952	0.55	10
12567	537065	2010/12/5	11:57	22837	4.65	8
12567	537065	2010/12/5	11:57	22846	16.95	1
12748	537429	2010/12/6	15:54	84970S	0.85	12
12748	537429	2010/12/6	15:54	22549	1.45	8

Table 3 Statistics of dataset that was used in the competition.

	Parameter	Value
#Customer	n	400
#Transaction	m	38,087
#Receipt		1,763
#Goods	ℓ	2,781
Price of goods (£)		0.04 – 4161
Quantity of goods		1 – 74215
Date of purchasing		2010/12/1 – 2011/12/9
#Mean of purchasing goods per customer	b	65
Mean of Jaccard Coefficient	μ	0.03

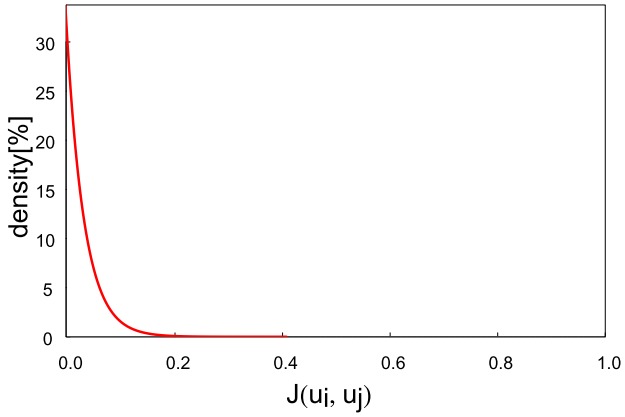

Fig. 2 Distribution of the Jaccard coefficient of the data used in the competition.

we found that a customer purchases $b = 65$ goods on average and the mean Jaccard coefficient is 0.03. **Figure 2** plots a histogram of the Jaccard coefficient between two customers. The maximum value of the Jaccard coefficient between two customers is 0.41 and the mean value is 0.03. This means that the most similar pair of customers has a similarity of only 41%. In other words, the sets of purchased goods are quite distinct and there is great diversity in customers. Note that the primary attributes are Use ID, Receipt ID, and Goods in the attributes. We use a simplified tables with these attributes in Section 3.1.

2.2 Record Linkage Risk from the Jaccard Coefficient

The Jaccard coefficient is a critical quantity for records due to the threat of relinking it with the de-identified data. This is because a motivated attacker who happens to observe the set of goods that the target customer purchased on the retail site can easily identify the target customer's records by examining the Jaccard coefficients of all candidate customers.

To prevent the attacker from identifying customers, we need to somehow modify the dataset so that the attacker cannot single out any one individual set. For example, the participants de-identify data by adding noise, deleting records, and adding dummy records. We define the quantities related to this process.

Definition 2 Let m and Δm be the total number of records in the dataset and the difference in the number of records through de-identification, respectively. The resulting number of records through de-identifying is $m' = m + \Delta m$.

The purchase history data is dynamic data consisting of some transactions records over time. We argue that dynamic data is more vulnerable than static data with a very high re-identification risk because of its observation over the long term. For example, the Online Retail Dataset has a re-identification risk via the purchased goods set for one year. In our past research [15], we found that an attacker having only one background knowledge element can identify individuals of Online Retail Dataset with about a 10% probability.

Algorithm 1 Re-identification Using the Jaccard Coefficient

Input: M, T, M', T'

Step 1.
Let M, T be the data and M', T' be the de-identified data. Let $I(u_i), I(u'_i)$ $(i = 1, \ldots, n)$ be a set of purchased goods of customer u_i in T and u'_i in T'.

Step 2.
Let $i^*_j = \arg\max_{i \in \{1,\ldots,n\}} J(I(u'_j), I(u_i))$ $(j = 1, \ldots, n')$ be the index of the customer in T' who is the nearest to u_i.

Output: $Q = (i^*_1, i^*_2, \ldots, i^*_n)$

To model the malicious behavior of the attacker, we propose a re-identification method using the characteristics of the purchased goods set of customers in Algorithm 1. In this method, we assume that an attacker has access to all of the transaction records in the original data. Given the de-identified data, the attacker will then attempt to re-identify the victim customer who has the data pair most similar to the target customer using the Jaccard coefficient. Note that the calculation amount of our algorithm is $O(n^2)$. In this paper, we define the attacker as follows.

Definition 3 The attacker knows a set of purchased goods $I(u_i)$ of all customers as background knowledge and tries to re-identify by the re-identification method using the characteristics of the purchased goods set of customers showed in Algorithm 1.

Note that the attacker knows *the kinds* of purchased goods, but does not know *the amount* of purchased goods in this paper. The amount of purchased goods is also valid background knowledge for an attacker re-identifying customers.

2.3 Re-identification Risk in PWS Cup 2016

We evaluate Algorithm 1 using the de-identified data submitted in PWS Cup 2016. Let $D_1, \ldots, D_{10}$ denote the de-identified data submitted by the top-nine teams in the competition. In this paper, we use 9 data excluding D_7 because this is de-identified by our team [*3]. **Table 4** provides the evaluation results for these datasets. Column (a) shows the maximum rate for successfully re-identified records by the participating teams and column (b) the rate of successfully re-identified records by Algorithm 1. The

[*3] We exclude D_7 from Table 4, because listing D_7 equally with other teams' data is not fair in evaluating the performance of Algolithm 1. D_7 (de-identified by our team) was processed by using special countermeasure to Algorithm 1, but other data are not processed in consideration of the Algolithm 1.

Table 4 Re-identification risk of data using characteristics of purchased goods (PWS Cup 2016).

Data	Max Re-identification Rate (a)	Our Method (b)
D_1	0.2225	*0.2225
D_2	0.2375	*0.2375
D_3	0.2550	*0.2550
D_4	0.2750	*0.2750
D_5	0.3025	*0.3025
D_6	0.3175	*0.3175
D_8	0.3725	0.2750
D_9	0.3850	*0.3850
D_{10}	0.5500	*0.5500

values marked * indicate that our algorithm outperforms any of the other participants. Even in the best de-identified data D_1, 22.25% of customers were re-identified using our algorithm. In this paper, the *re-identification ratio* is defined as follows.

Definition 4 Let *Reid* be a re-identification ratio defined by a fraction of re-identified customers.

3. Our Proposal on De-identification

3.1 How to Prevent Data Being Distinguished by the Jaccard Coefficient

The challenge is to prevent data from being identified by the Jaccard coefficient. We pursued this by mixing the records of purchased goods so that no customer could be re-identified with the Jaccard coefficient using three methods. (1) Altering some existing records ($m' = m$). (2) Deleting some existing records ($m' < m$). (3) Adding some dummy records ($m' > m$). Methods 1 and 2 (altering and deleting records) may lose their data accuracy. In contrast, Method 3 (adding some dummy records) preserves the existing purchase histories. **Table 5** shows pros and cons of these three methods.

In this paper, we add some fake records that do not spoil the utility of the data. In **Fig. 4**, we illustrate how our algorithm works. Table (a) is the original transaction data T for three attributes, user IDs, record IDs, and the good IDs of m records. We detail the list of purchased goods for each customer in Table (b). In this case, we mix up three customers u_1, u_2, and u_3 by adding some dummy records randomly chosen from the set of goods. Finally, we provide the de-identified data in Table (d), shown as $I(u'_1) = I(u'_2) = I(u'_3) = I(u_1) \cup I(u_2) \cup I(u_3) = \{g_1, g_2, g_3, g_4, g_5\}$.

Figure 5 shows an example of how to add dummy records to Online Retail Dataset. Table (a) is an example of original data that contains 5 records and 2 users and Table (b) is an example of processed data that contains 6 records and 2 users. Attacker knows $I(u_1) = \{A, B, C\}$ and $I(u_2) = \{A, B\}$ as background knowledge and we add dummy records to prevent his re-identification using the Jaccard coefficient. In this case, we have to make $I(u_1)$ and $I(u_2)$ to be same set $\{A, B, C\}$ and add a dummy record (6th record) in Table (b) for u_2. This record contains same values in 4 attributes (User ID, Receipt ID, Date, Time) as the latest record of u_2 and values about additional goods in 2 attributes (Goods, Price) and "1" in the attribute of Number. Note that attacker knows only background knowledge about the attribute of Goods in this paper and the values of other attributes do not affect re-identification risk.

As shown, there is a trade-off between the number of dummy records Δm and the utility of the de-identified data. If we attempt to unify all customers, the number of dummy records will be huge and the data useless. Therefore, we need to minimize the number of dummy records by carefully classifying the set of customers into some small clusters that preserver similar purchasing characteristics.

Table 5 Pros and cons of three de-identification methods.

Method	Pros	Cons
Altering	Processing is independent by each record.	Satisfying k-anonymity might be impossible.
Deleting	The de-identified data does not contain any incorrect information.	The amount of data decreases.
Adding	The de-identified data contains all information contained in original data.	The de-identified data contains incorrect information.

Algorithm 2 Algorithm to Add Dummy Records

Input: $M, T, X = \{x_1, x_2, \ldots, x_c\}$

Let u be a user and x be a cluster in X.

(1) Add dummy records that contain goods $I(x) - I(u)$ as a transaction of u in each cluster x. Set T' be a transaction data that contains some dummy records.

(2) Unify the set of purchased goods by each customer as $I(x) = \bigcup_{u \in x} I(u)$ in each cluster $x \in X$. Set M' be a customer data that contains some unified sets.

Output: T', M'

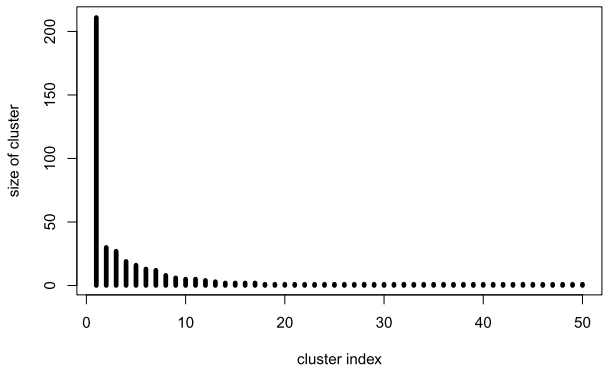

Fig. 3 Distribution of cluster size via the Jaccard coefficient.

The simplest way to cluster similar customers is to begin with representative c customers, and then extend them by assigning other customers to the closest cluster, letting c be the number of clusters, $X = \{x_1, \ldots, x_c\}$, the set of clusters, and $s_i = |x_i|$ the size of a cluster. Note that cluster x_i is that set of customers partitioning the whole set of customers U, i.e., $\bigcup_{i=1}^{c} x_i = U$ and we define the number of clusters as c because the notation k is confusing with the k-anonymity. Algorithm 2 details the method to add some dummy records to each cluster.

3.2 TF-IDF Distances between Records

Algorithm 2 is too simple to deal with unbalanced transaction records. Generally, purchase history data contains many goods that are distributed "long-tailed," whereby a few customers occupy most records and so a simple clustering method involves a large number of dummy records. For instance, **Fig. 3** depicts the distribution of the cluster sizes resulting in the simple clustering method (k-means method) with the Jaccard coefficient as the distance between two customers. When we perform the simple

Algorithm 3 Weighting of Purchased Goods via TF-IDF

Input: $u_i \in U, I(u_i), c$

Step 1. Let $\boldsymbol{v}_i = (f_{i1}, f_{i2}, \ldots, f_{i\ell})$ be a characteristics vector of dimension ℓ of u_i where

$$f_{ij} = \begin{cases} 1 & \text{if} \quad g_j \in I(u_i) \\ 0 & \text{otherwise.} \end{cases}$$

Step 2. Let $D_j = \{u_i \in U | g_j \in I(u_i)\}$ be a set of customers who purchased a good g_j. Let $f'_{ij} = (f_{ij} / \sum_{k=1}^{\ell} f_{ik})(\log \frac{n}{|D_j|} + 1)$ be a weight of f_{ij} via TF-IDF and $\boldsymbol{v}'_i = (f'_{i1}, f'_{i2}, \ldots, f'_{i\ell})$ be a characteristics vector of u_i.

Step 3. Classify the customers U into clusters via k-means and the cosine similarity between the characteristics vectors $\boldsymbol{v}'$.

Output: $X = \{x_1, x_2, \ldots, x_c\}$

(a) T

U	Receipt ID	Goods
u_1	100	g_1
u_1	300	g_2
u_2	500	g_1
u_2	500	g_3
u_2	600	g_5
u_3	...	...

(d) T' ∗ : Dummy Record

U	Receipt ID	Goods	
u'_1	100	g_1	
u'_1	300	g_2	
u'_1	300	g_3	∗
u'_1	300	g_4	∗
u'_1	300	g_5	∗
u'_2	...	...	

(b)

	U	$I(U)$
1	u_1	$\{g_1, g_2\}$
2	u_2	$\{g_1, g_3, g_5\}$
3	u_3	$\{g_4, g_5\}$

(c)

	U'	$I(U')$
1	u'_1	$\{g_1, g_2, g_3, g_4, g_5\}$
2	u'_2	$\{g_1, g_2, g_3, g_4, g_5\}$
3	u'_3	$\{g_1, g_2, g_3, g_4, g_5\}$

Fig. 4 How to add Dummy Records to data.

(a) Original data

User ID	Receipt ID	Date	Time	Goods	Price	Number
u_1	1	2010/12/1	8:00	A	5	10
u_1	1	2010/12/1	8:00	B	10	20
u_1	2	2010/12/2	12:00	C	100	30
u_2	3	2010/12/1	15:00	A	5	5
u_2	4	2010/12/5	20:00	B	10	15

(b) Processed data ∗ : Dummy Record

User ID	Receipt ID	Date	Time	Goods	Price	Number	
u'_1	1	2010/12/1	8:00	A	5	10	
u'_1	1	2010/12/1	8:00	B	10	20	
u'_1	2	2010/12/2	12:00	**C**	**100**	30	
u'_2	3	2010/12/1	15:00	A	5	5	
u'_2	**4**	**2010/12/5**	**20:00**	B	10	15	
u'_2	**4**	**2010/12/5**	**20:00**	**C**	**100**	1	∗

Fig. 5 How to add Dummy Records to online retail dataset.

clustering method (k-means method) with transaction data used in PWS Cup 2016, the largest cluster size is 211, which is excessively large, while the remaining 33 clusters have just one element. This suggests the cluster sizes are greatly biased.

To address the *monopoly behavior* of clusters, we propose a method to form clusters, where we simply replace the Jaccard coefficient by the TF-IDF value of the set of goods for measuring similarity between customers. In this paper, we use TF-IDF defined in Algorithm 3. Namely, we use the boolean matrix of the elements of terms (good) times by the array of the inverse number of documents (customers), that is, that contains arrays of the term (good) a weight of goodness of cluster for goods. Consequently, we obtain an improved clustering method using the TF-IDF weight in Algorithm 3.

(a)

U	$I(U)$
u_1	$\{g_1, g_2\}$
u_2	$\{g_1, g_2\}$
u_3	$\{g_1, g_2, g_3, g_4\}$
u_4	$\{g_1, g_3, g_4\}$

(b)

	g_1	g_2	g_3	g_4	
u_1	1	1	0	0	$\boldsymbol{v_1}$
u_2	1	1	0	0	$\boldsymbol{v_2}$
u_3	1	1	1	1	$\boldsymbol{v_3}$
u_4	1	0	1	1	$\boldsymbol{v_4}$

(d)

U	cluster
u_1	0
u_2	0
u_3	1
u_4	1

(c)

g_1	g_2	g_3	g_4	
0.5	0.56	0	0	$\boldsymbol{v'_1}$
0.5	0.56	0	0	$\boldsymbol{v'_2}$
0.25	0.28	0.32	0.32	$\boldsymbol{v'_3}$
0.33	0	0.42	0.42	$\boldsymbol{v'_4}$

Fig. 6 Example of clustering of customers via TF-IDF.

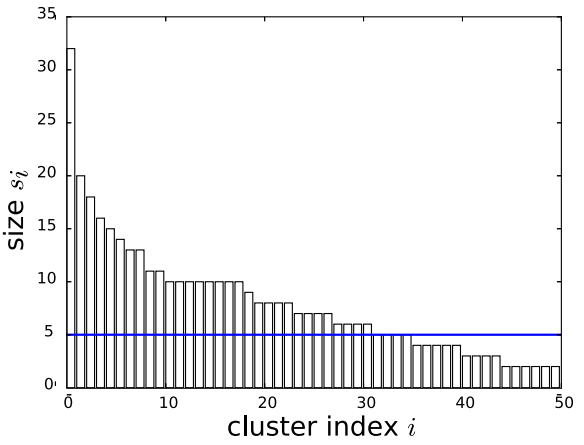

Fig. 7 Distribution of cluster size via Method 1 ($c = 50$).

Figure 6 depicts how the algorithm works for an example of four customers. Suppose we classify customers $U = \{u_1, u_2, u_3, u_4\}$ into two clusters $X = \{x_1, x_2\}$. Table (a) details the list of the purchased good sets for the four customers, characterized by a binary matrix of purchased goods in (b). We replace the binary matrix by the matrix of TF-IDF weights of goods shown in (c). For example, the characteristics value of goods g_1 of u_1 is 0.5 because TF = 1/2 and IDF = 1. Finally, we have the resulting clusters $x_1 = \{u_1, u_2\}$ and $x_2 = \{u_3, u_4\}$ based on the cosine similarity between the two customers, as shown in (d). Note that the size of the clusters is evenly distributed and the clusters are well balanced because of the similarities in the TF-IDF values.

3.3 Method 1: De-identification Method Based on k-means Clustering

Method 1 using weighting goods in the TF-IDF performs a clustering of the k-means method via cosine similarity, and adds some dummy records so that the sets of purchased goods in the cluster are indistinguishable from one another.

Figure 7 plots the distribution of cluster sizes when the number of clusters is specified as $c = 50$. Compared with the clustering result in Fig. 3, the deviation in cluster sizes is smaller because

Algorithm 4 Algorithm to Balance Method 1 (Method 2)

Input: s_{min}, c, M, T
  Clustering via Method 1
  Set of clusters: $X = \{x_1, x_2, \ldots, x_c\}$
  for x **in** $\{x_i \in X \big| |x_i| < s_{min}\}$ **do**
    Maximum cluster: $x_{max} \in X$
    while $|x'| < s_{min}$ **do**
      $u_j = \underset{u_j \in x_{max}, u_i \in X}{\arg\max}\ J(I(u_i), I(u_j))$, $x'_{max} \leftarrow x_{max} - \{u_j\}$, $x' \leftarrow x \cup \{u_j\}$
    end while
  end for
  Add some dummy records in a way like Section 3.1.
Output: M', T', P

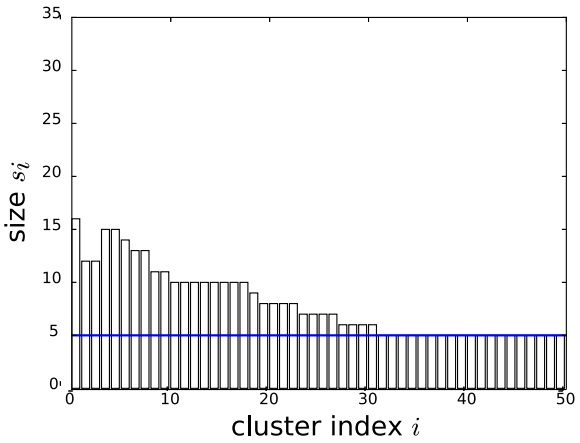

Fig. 8 Distribution of cluster size via Method 2 ($s_{min} = 5, c = 50$).

of the TF-IDF weights. Letting x_{max} and x_{min} be the largest and smallest clusters, respectively, we observe that $|x_{max}|$ is 32 and $|x_{min}|$ is 1. Obviously, there is still skew in the distribution and it also suffers from a loss of utility due to adding too many dummy records for customers belonging to a large cluster.

3.4 Method 2: Balanced De-identification Method

To address the unbalanced issue, we propose a second de-identification method with the restriction of the smallest cluster size. In Method 2, we restrict the cluster sizes so that these are not below the lower limit of s_{min}, which corresponds to quantity k of k-anonymity.

Algorithm 4 shows the modified Method 2. We move a customer belonging to the largest cluster x_{max} to the cluster with a size less than s_{min}. We repeat the movement operation until all cluster sizes are larger than s_{min}. The minimum threshold value s_{min} is specified depending on the number of clusters c and will be in the range of $\{2, 3, \ldots, \lfloor n/c \rfloor\}$.

Figure 8 illustrates the distribution of cluster sizes in Method 2 when the minimum threshold is $s_{min} = 5$ and the number of clusters is $c = 50$. Compared with the clustering result in Fig. 8, the maximum cluster size falls from 32 (Fig. 7) to 16 (Fig. 8) and the sizes of all clusters are satisfied as they are all more than s_{min}.

Finally, we show results from comparing the simple method (described in Section 3.2) with our proposal methods (described in Sections 3.3, 3.4) in **Table 6**.

Table 6 A comparison of the simple method and our proposal methods.

Item	Algorithm 2 (simple k-means)	Algorithm 3 (TF-IDF)	Algorithm 4 (Balance)
Methodologies	k-means	TF-IDF	Regulation of maximum size of cluster
Monopoly issue	Yes	No	No
Unbalanced issue	Yes	Yes	No
Size of max cluster	211	32	16
# singleton (one-element cluster)	33	0	0

4. Evaluation of Our Method

4.1 The Relationship between the Utility and the Number of Dummy Records

We evaluate the utility of de-identified data by the three metrics, U1-cMAE, U2-cMAE, and U3-RFM that were used in PWS Cup 2016 [16], [17], [18]. Here, U1-cMAE and U2-cMAE are metrics that evaluate utility of de-identified data with mean absolute error (MAE) between cross tabulations of the original data and the de-identified data restricted with two categories (sex, nationality) of customers. The cross tabulation contains 72 cells because the customers in dataset for PWS Cup 2016 are of two sexes and 36 nationalities and the values (e.g., mean of prices) in this cross tabulation will change when data are de-identified. Here, U3-RFM is a metric that evaluates the utility of de-identified data with RFM (Recently, Frequency, Monetary) analysis that is a method to analyze customers. The customers are divided into 1,000 ranks (combination of 10 most-recent-purchase rank, 10 Frequency rank, and 10 Monetary rank) and frequency of this rank will change when data are de-identified. The utility of de-identified data decreases when the evaluation value of these utility metrics increases because the evaluation value signifies an error size between the original and de-identified data.

The utility of the de-identified data greatly depends on the number of dummy records Δm. We shows a correlation between Δm and utility evaluation values to indicate that the number of dummy records that we showed how to add in Section 3.1 is one of the utility metrics. **Table 7** provides the relationship between some known utility metrics used in PWS Cup 2016 and Δm. The Jaccard Reid signifies the re-identification ratio of Algorithm 1 and the random Reid signifies the mean re-identification ratio when an attacker re-identifies a customer randomly chosen within a given cluster. We identify a strong negative correlation between Δm and utility metrics (U1-cMAE1, U2-cMAE, and U3-RFM) that are used in PWS Cup 2016. This implies that the utility of the de-identified data decreases as Δm increases. When the cluster size c increases, Δm decreases, and accordingly, the ratio of re-identification increases because the correlation coefficient between Δm and c is –0.8454.

Figure 9 shows the scatter plot between Δm and the utility metric U1-cMAE. When we add dummy records as far as $0 < \Delta m \leq$ 300,000, the utility metrics are $0.0 \leq U_1 \leq 1.02$ and the utility of the data decreases with an increase in Δm. In the experiment, we evaluate our de-identification methods ten times for Δm when using the Jaccard re-identification method.

Table 7 Correlation Coefficients between Δm and Utility metrics.

	Δm	$U1$-cMAE	$U2$-cMAE	$U3$-RFM	Jaccard *Reid*	random *Reid*	c
Δm	1.0000						
$U1$-cMAE	0.9798	1.0000					
$U2$-cMAE	0.9798	1.0000	1.0000				
$U3$-RFM	0.9547	0.9876	0.9876	1.0000			
Jaccard *Reid*	−0.8586	−0.9327	−0.9327	−0.9494	1.0000		
random *Reid*	−0.8489	−0.9247	−0.9247	−0.9432	0.9996	1.0000	
c	−0.8454	−0.9220	−0.9220	−0.9406	0.9994	0.9999	1.0000

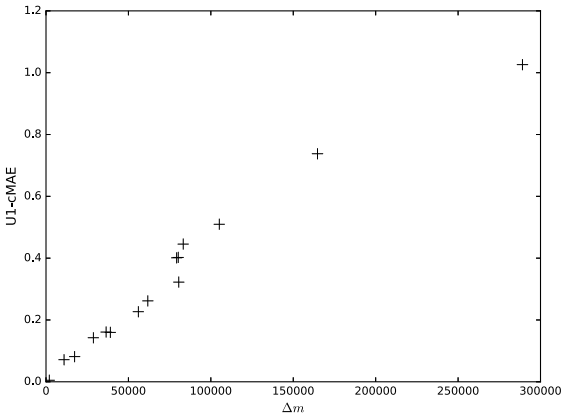

Fig. 9 Relationship between Δm and metrics U1.

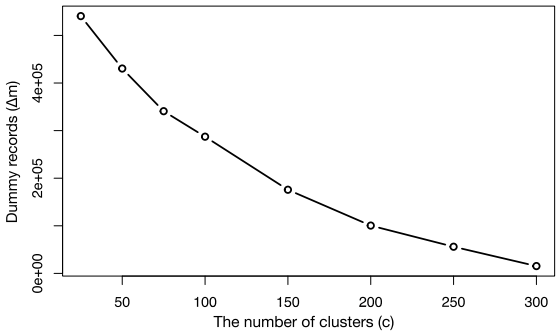

Fig. 10 Δm of simple kmeans clustering method.

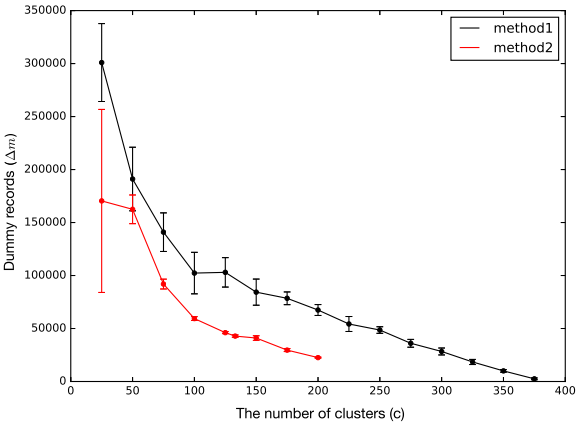

Fig. 11 Comparison of Δm between Method 1 and Method 2.

4.2 Theoretical Value of Δm and the Jaccard Coefficient

Figure 10 shows the number of dummy records Δm of the simple k-means clustering method introduced in Section 3.2. We have to add 540,583 dummy records when there are 25 clusters made based on the simple clustering method. **Figure 11** shows the distribution of Δm of our proposal methods with respect to c. The black line shows Δm of Method 1 and the red line shows Δm of Method 2. In comparison between Figs. 10 and 11, the number of dummy records for our proposed methods are obviously less than the number for the simple clustering method. In the experiment, we investigate the purchase history data of 400 customers with the threshold value s_{min} specified as $\lfloor \frac{n}{c} \rfloor$. In a comparison of Methods 1 and 2, Method 2 has only about 53% of the Δm of Method 1.

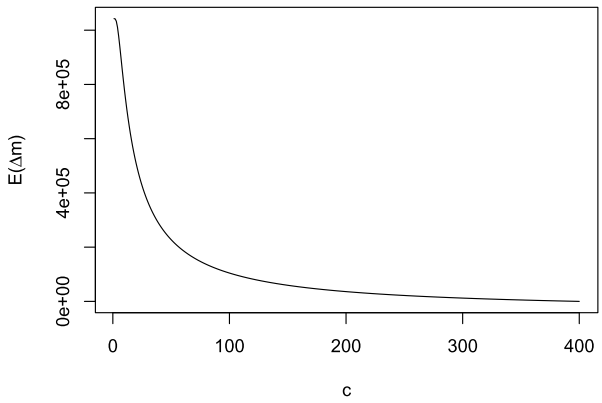

Fig. 12 Relationship between c and $E(\Delta m)$.

We are interested in estimating the theoretical value of Δm in the method to add dummy records and find that the expected value of Δm given m, n, ℓ, and c as follows and **Fig. 12** shows the distribution of $E(\Delta m)$ with respect to c.

$$E(\Delta m) = n\ell \left\{ \left(1 - \frac{1}{\ell}\right)^{\frac{m}{n}} - \left(1 - \frac{1}{\ell}\right)^{\frac{m}{c}} \right\}$$

We quantify a degree of similarity between customer u_i and u_j in terms of the sets of purchased goods as the Jaccard coefficient as follows.

Definition 5 Let μ be the mean of the Jaccard coefficients between every two customers defined as $\mu = 1/\binom{n}{2} \sum_{i \neq j \in U} J(u_i, u_j)$, where $J()$ is defined by $J(u_i, u_j) = |I(u_i) \cap I(u_j)|/|I(u_i) \cup I(u_j)|$. Let μ be the mean size of the intersection of the two sets of goods purchased by distinct customers.

Given the dataset statistics, we estimate the mean Jaccard coefficient in the following way.

Proposition 1 Let b and h be the mean number of goods that a customer purchases in a year and the mean size of the intersection of the two sets of goods purchased by distinct customers, i.e., $h = |I(u_i) \cap I(u_j)|$. Then, the mean size of the intersection is $h = 2b\mu/1 + \mu$.

Proof: We are able to transform μ

$$\mu = \frac{E(|I(u_i) \cap I(u_j)|)}{E(|I(u_i)|) + E(|I(u_j)|) - E(|I(u_i) \cap I(u_j)|)} = \frac{h}{2b - h}.$$

By solving for h, we obtain the proposition. □

Table 8 Relationship between s_{min} and Δm.

	c = 50			c = 100			c = 125		
	Δm	Jaccard *Reid*	random *Reid*	Δm	Jaccard *Reid*	random *Reid*	Δm	Jaccard *Reid*	random *Reid*
Method 1	182897	0.1728	0.1235	128568	0.3060	0.2488	97581	0.3692	0.3120
Method 2									
s_{min} = 2	183902	0.1729	0.1223	99228	0.3061	0.2475	60492	0.3687	0.3105
s_{min} = 3	175449	0.1726	0.1222	68357	0.3041	0.2480	46101	0.3667	0.3102
s_{min} = 4	162474	0.1723	0.1218	59374	0.3044	0.2465			
s_{min} = 8	125798	0.1681	0.1218						

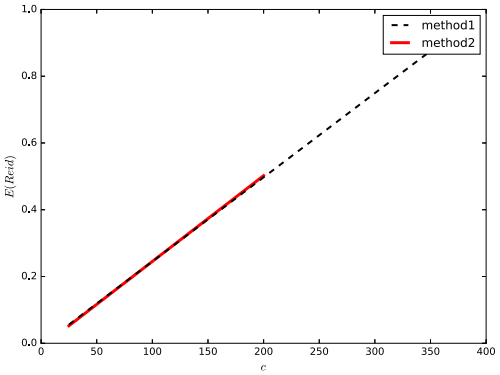

Fig. 13 Comparison of security of Method 1 and Method 2.

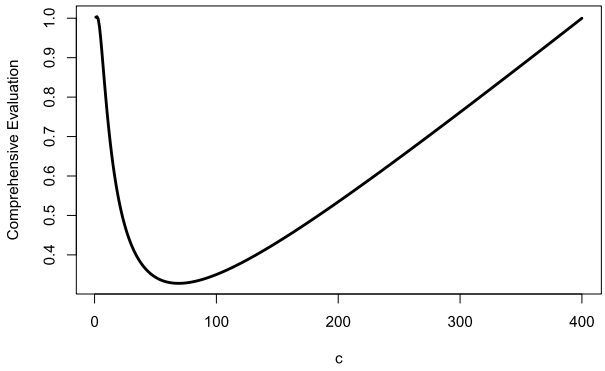

Fig. 14 Relationship between c and comprehensive evaluation of de-identified data.

4.3 Utility and Security

Table 8 shows the relationship between Δm and s_{min}. Note that Δm is minimized when s_{min} is $\lfloor \frac{n}{c} \rfloor$ in each c. We observe that the Jaccard coefficients are distributed across a small range and the standard deviation of the Jaccard coefficient is smaller than 0.01.

We show the actual rate of re-identified records of each c in the column labeled Reid in Table 8. **Figure 13** shows the rates to be re-identified of de-identified data in the methods. For each of the de-identified data, we apply Algorithm 1, being the Jaccard re-identification method described in Section 2 for a certain c. The Jaccard re-identification method successfully identifies at least one customer in each cluster who purchased goods most frequently. We obtain the simplest result that the expected rate of de-identified data to be re-identified using either Method 1 or 2 is calculated as $E(Reid) = c/n$.

In this paper, we evaluate the de-identified data comprehensively based on the metrics $\alpha E(\Delta m) + E(Reid)$ referring to the metrics (Utility + Security)/2 used in PWS Cup 2016 (Let α be a coefficient to normalize $E(\Delta m)$ to the range of $0 \leq E(\Delta m) \leq 1$). We show the relationship between c and the comprehensive evaluation of de-identified data in **Fig. 14** when n = 400, m = 38,000, l = 2,700, α = 1/1,042,653. In this case, the comprehensive evaluation value is the smallest when c = 69, or in other words, the de-identified data processed based on our method will be the best when c = 69.

5. Related Works

Technical Specification ISO/TS 25237 [6] defines anonymization as "a process that removes the association between the identifying data and the data subject." The ISO definition classifies anonymization techniques into masking and de-identification. Many anonymization algorithms have been proposed to preserve privacy while retaining the utility of the data that have been *anonymized*. In other words, the data are made less specific so that a particular individual cannot be identified. Anonymization algorithms employ various operations, including *suppression* of attributes or records, *generalization* of values, replacing values with *pseudonyms*, *perturbation* with random noise, sampling, rounding, swapping, top/bottom coding, and microaggregation [7], [8].

Koot et al. proposed a method to quantify anonymity via an approximation of the uniqueness probability using a measure of the Kullback–Leibler distance [9]. Monreale et al. proposed a framework for the anonymization of semantic trajectory data called c-safety [10]. Based on this framework, Basu et al. presented an empirical risk model for privacy based on k-anonymous data release [11]. Their experiment using car trajectory data gathered in the Italian cities of Pisa and Florence allowed the empirical evaluation of the protection of anonymization of real-world data. Stokes et al. defined n-confusion [12], which is a generalization of k-anonymity. In 2017, Torra presented a general introduction to data privacy [13]. Li and Lai proposed a definition of a new δ-privacy model which requires that no adversary could improve his own background knowledge more than δ in privacy degree [14]. Xiao et al. proposed a new generalization principle m-invariance that effectively limits the risk of privacy disclosure in re-publication. This method consists of generalization and adding dummy records that resemble those of other customers in other datasets.

6. Conclusions

In this paper we reveal the risk of data to be re-identified via the characteristics of purchasing goods of customers and propose a de-identification method by minimizing additional dummy records to be add the datasets. In our method, we classify the set of customers into some clusters based on the characteristics of purchasing goods weighted as the TF-IDF. We demonstrate that our proposed algorithm reduces the number of dummy records as

far as restricted size of clusters.

Topics for future study include evaluating the accuracy of clustering and effectiveness in case of other datasets. We will try to use other de-identification methods like deleting and adding noise to improve our de-identification method.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number JP18H04099.

References

[1] Personal Information Protection Commission Secretariat: Report by the Personal Information Protection Commission Secretariat: Anonymously Processed Information –Towords Balanced Promotion of Personal Data Utilization and Consumer Trust (2017).
[2] ISO/IEC 20889: Privacy enhancing data de-identification terminology and classification of techniques (2018).
[3] Kikuchi, H., Yamaguchi, T., Hamada, K., Yamaoka, Y., Oguri, H. and Sakuma, J.: What is the Best Anonymization Method? – A Study from the Data Anonymization Competition Pwscup 2015, *Data Privacy Management Security Assurance* (*DPM2016*), LNCS 9963, pp.230–237 (2016).
[4] UCI Machine Learning Repository: A WEB Page, available from ⟨http://archive.ics.uci.edu/ml/index.php⟩ (accessed 2018-12-17).
[5] UCI Machine Learning Repository: Online Retail Data Set, available from ⟨https://archive.ics.uci.edu/ml/datasets/online+retail⟩ (accessed 2018-12-17).
[6] ISO Technical Specification ISO/TS 25237: Health informatics – Pseudonymization (2008).
[7] Information Commissioner's Office (ICO): Anonymisation: managing data protection risk code of practice (2012).
[8] Aggarwal, C.C. and Yu, P.S.: A General Survey of Privacy-Preserving Data Mining, Models and Algorithms, *Privacy-preserving Data Mining*, pp.11–52, Springer (2008).
[9] Koot, M.R., Mandjes, M., van't Noordende, G. and de Laat, C.: Efficient probabilistic estimation of quasi-identifier uniqueness, *Proc. ICT OPEN 2011*, pp.119–126 (2011).
[10] Monreale, A., Trasarti, R., Pedreschi, D., Renso, C. and Bogorny, V.: *C*-safety: A framework for the anonymization of semantic trajectories, *Trans. Data Privacy*, Vol.4, No.2, pp.73–101 (2011).
[11] Basu, A., Monreale, A., Trasarti, R., Corena, J.C., Giannotti, F., Pedreschi, D., Kiyomoto, S., Miyake, Y. and Yanagihara, T.: A risk model for privacy in trajectory data, *Journal of Trust Management*, pp.2–9 (2015).
[12] Stokes, K. and Torra, V.: *n*-confusion: A generalization of *k*-anonymity, *EDBT/ ICDT Workshops 2012*, pp.211–215 (2012).
[13] Torra, V.: Data Privacy: *Foundations, New Developments and the Big Data Challenge, Studies in Big Data 28*, Springer (2017).
[14] Li, Z., Lai, T.H.: δ-privacy: Bounding Privacy Leaks in Privacy, *Preserving Data Mining*, DPM/CBT 2017, LNCS 10436, pp.124–142, Springer (2017).
[15] Ito, S., Kikuchi, H. and Nakagawa, H.: Attacker models with a variety of background knowledge to de-identified date, *J. Ambient Intell. Human. Comput.*, pp.1–11 (2019).
[16] Nojima, R. et al.: How to Handle Excessively Anonymized Datasets, *Journal of Information Processing*, Vol.26, pp.477–485 (2018).
[17] Kikuchi, H., Oguri, H., Nojima, R., Hamada, K., Murakami, T., Yamaoka, Y., Yamaguchi, T. and Watanabe, C.: PWS CUP Competition: De-identify Transaction Data Securely, *Computer Security Symposium*, 2A1-2, pp.271–278, in Japanese (2016).
[18] Kikuchi, H.: Data Anonymization and Quantifying Risk Competition (online), available from ⟨https://project.inria.fr/FranceJapanICST/files/2017/05/HKikuchi_presentation_2017.pdf⟩ (accessed 2020-3-26).
[19] Xiao, X. and Tao, Y.: m-invariance: Toward privacy preserving republication of dynamic datasets, *Proc. SIGMOD'07*, pp.689–700 (2007).

Satoshi Ito received Bachelor's degree in science and Master's degree in mathematical science from Meiji University in 2017 and 2019, respectively. He is currently a graduate-student at Graduate School of Advanced Mathematical Sciences of Meiji University since 2017.

Reo Harada received Bachelor's and Master's degrees from Meiji University in 2017 and 2019, respectively. He is currently engaged in development related to ADAS at SUBARU CORPORATION Engineering Div. 1 Advanced Safety Design Dept.

Hiroaki Kikuchi received B.E., M.E. and Ph.D. degrees from Meiji University in 1988, 1990 and 1994. After he working in Fujitsu Laboratories Ltd. in 1990, he had worked in Tokai university from 1994 through 2013. He is currently a professor in at Department of Frontier Media Science, School of Interdisciplinary Mathematical Sciences, Meiji University. He was a visiting researcher of the school of computer science, Carnegie Mellon University in 1997. He has been board chairman, Japan Computer Emergency Response Team Coordination Center (JPCERT/CC) since 2018. His main research interests are network security, cyrptographical protocl, privacy-preserving data mining, and fuzzy logic. He received the Best Paper Award for Young Researcher of Japan Society for Fuzzy Theory and Intelligent Informatics in 1990, the Best Paper Award for Young Researcher of IPSJ National Convention in 1993, the Best Paper Award of Symposium on Cryptography and Information Security in 1996, the IPSJ Research and Development Award Award in 2003, the Journal of Information Processing (JIP) Outstanding paper Award in 2010 and 2017 and the IEEE AINA Best Paper Award in 2013. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE), the Japan Society for Fuzzy Theory and Systems (SOFT), IEEE and ACM. He receives Information Processing Society of Japan (IPSJ) Fellow.