To appear in the Proceedings of 2nd IEEE Int. Conference on Smart Cloud (SmartCloud 2017), Nov, 2017, New York, USA.

# Learning Data Privacy and Terms of Services from Different Cloud Service Providers

Mehdi Bahrami[1], Mukesh Singhal[2] and Wei-Peng Chen[3]

[1,3] Fujitsu Laboratories of America, Inc.
Sunnyvale, California, United States
{mbahrami;wchen}@us.fujitsu.com

[2] University of California, Merced
msinghal@ucmerced.edu

***Abstract*—People are using on daily basis websites, mobile applications, and online software nowadays. Major mobile and desktop software applications have been moved to cloud computing environment that allows users to interact with a variety of applications on the move and pay only for additional usage of services on-demand. Each online application has its own term of service, data privacy agreement that needs to be signed by users even if users are not able to read all documents. In addition, the online service providers collect a variety of information from online users depending on their agreement. This variety of agreements, terms and policies might be challenging for users who use several online applications every day. Data privacy plays a key role in the age of online information centric. It might be more challenging for the users who subscribed to several accounts from different online service providers. In this study, we proposed a machine learning-based method that generates a vector representation of term of services. Then, it generates a matrix of term of services for different cloud service providers. Finally, it employs a clustering algorithm to analyze the collected data and it monitors data privacy of users in real-time according to written terms of services from cloud service providers.**

***Keywords—data privacy; machine learning; natural language processing.***

## I. INTRODUCTION

Todays, major online software-providers offer free of charge, a long term trial, or a low cost subscription for using different online mobile and web services. The main advantage of the using cloud services is the cost of a service for the users. However, users must sign or accept the term of services for data collection and data sharing policy according to unique terms of each service provider. Signing up for a service means the user is able to take within provider's social platform [1]. The term of service describes a large numbers of terms, information collection from the provider, how the information has been collected, how the information will be shared with other third-parties as well as type of information collection. Therefore, the term of service for each service might be different and lengthy, and it might come with detail that needs to be explained in several pages.

In the virtual world, we can sign up/subscribe to a variety of cloud service even without reading any terms of services. This advantage allows users to subscribe to a large number of services even if they do not use the service. It also allows users to use services without paying any fee. However, this advantages generally comes at the loss of data privacy [2, 3].

According to "*Terms of Service; Didn't Read*"[1]: most of the users never read any term of service because the terms of service regularly are too long and most of the people cannot read the whole documents in 10 minutes. However, in order to get access or take advantage of the service, users must accept the terms of service for signing up or subscribing for a service. The challenge of data privacy might be started at this point when the users are not able to read the whole agreement quickly.

The challenge of data privacy might be increased by signing up for a mobile service or web service when a user start sharing information with others (actually with the service provider as the first recipient) by adding messages, photos, friends, family members (i.e., when using a social network service), emails, and agreements with other web services and mobile based services (i.e., by using a free email service). When a service becomes a major player in the user's daily life (such as email), then it might consist of not only daily no confidential information, such as regular messages or photos, but it might also consist of confidential information, such as social security numbers, passwords, and actual users' identifications. At this point, the users are not able to avoid sharing this information with the service providers.

The information collected from users by service providers might be used for data mining purpose for each user, or each group of users. It might be shared, and sold to other third-parties, such as internal or external entities. The shared data also can be

[1] Website is available at https://tosdr.org/

leveraged by multiple sources in order to understand users' behaviors, such as shopping plans.

In addition, in the age of online mobile applications when a user uses several online free applications per day, it might be difficult to monitor own data privacy violation.

Our contribution in this paper are as follows:

*1)* we develop a method that collects information from different service providers;
*2)* we describe step-by-step method of producing a vector represenation of Terms of Services that allows users to quickly understand the trems and do a comparison between different terms of services;
*3)* we use the vector representations of different Terms of Services to train an unsupervised machine learning method, k-means model, to undestand legible and elligible data privacy violations, overlap between different terms of services from different prespectives;
*4)* we use the model to provide data privacy recommandations to the users.

This paper is organized as follows: the next section describes the motivation of this study and our goals. *Section III* explains a method that collects information from *Terms of Services*, and generating a vector representation of different Terms of Services from different cloud service providers. *Section IV* explains the proposed method that uses a well-known machine learning-based algorithm, *k-means*, to generate an unsupervised model of Terms *of Services*. We provide experimental results, visualization of different perspectives of input variables as a proof of the concept for the proposed method as well as the evaluation results in *Section V*. Finally, we summarize this study in *Section VI*.

## II. Motivation

The ultimate goal of our study is preserving users' data privacy when a user is subscribing to different cloud services. As we presented a light-weight data privacy method (DPM) previously [4], it enables users to protect their data privacy while they are using different cloud services. It is also capable to run on parallel GPU cores [5]. However, the method requires some initialization steps and if the user initializes the method for two service providers who collect data similarly, then the service providers might be able to violate users' data privacy. This issue leads us to introduce this study when it helps users to understand similarity between different service providers based on different criteria.

The goal of this study is to transfer written terms of services to a matrix of integer value (our target in this study) or float values. The generated matrix lets us to process the actual documentation in different data representations when a machine is able to process some algorithm on this data. It enables us to perform some computation on this dataset in order to understand the whole agreement of cloud services. By accomplishing this study, DPM is then able to detect similar cloud service providers based on their criteria which have been described in terms of services. Therefore, this extension enables DPM to run different initialization to scramble data by using the generated model from this study. It enables DPM to perform its own tasks efficiently and more importantly secure.

In the real-world, the proposed method allows users to understand different data privacy quickly and efficiently by using generated machine learning models. Therefore, the users do not need to read the whole documentation of the terms of services for each subscription service. In addition, if the generated machine learning model leverages with DPM, it allows users to protect their data on client side based on written the terms of services.

## III. A Vector Representation of Terms of Services

This section aims to generate a list of data privacy terms for each service provider. Then, we will be able to compare different terms against each other as well as perform data mining on different terms from different service providers.

Each service provider describes her own data privacy, the term of services through a website. Each service provider describes what type of information has been collected from users, which information might be shared with third-parties. For instance, Fig 1. shows a snapshot of data privacy right for users of Fujitsu North America's website[2]. In this page, there are different sections including data collection, security, Opt Ins and etc. Reading all these pages for a user is time consuming and even difficult to remember all the policies which the user has been agreed during sign up process.

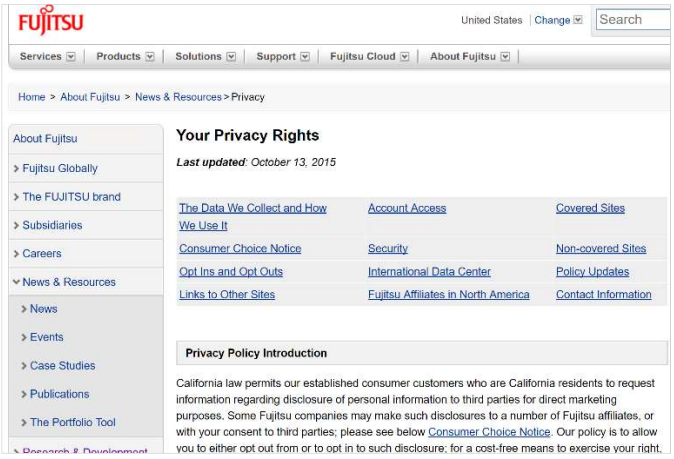


Fig. 1. A snapshot of data privacy right for users of Fujitsu North America's website

We use a web crawler to collect all related pages of each service provider's website. A distributed web crawler based on cloud computing environment that uses parallel agents may be implemented as described in [6]. In order to collect a large number of pages in a short time, we process the web crawler as

[2] Available at: http://www.fujitsu.com/us/about/resources/privacy/index.html

several independent distributed agents. Each agent such that collects a list assigned pages. Each agent may collect full or partial pages of terms of services of one provider or multiple providers.

Since each defined link in a web page might point to other pages and even external websites, we define several criteria as shown in Fig 2 and the following parameters. This criteria limits the parallel web crawler agents to crawl limited pages which more likely described our target pages (terms of services).

i. *MaxDepth* that indicates to maximum depth of the link parser. This condition allows the web crawler to collect a limited pages related to the target pages.

ii. *MaxPagePerDomain* that limits the number of pages for each service provider.

iii. *MaxPagePerAgent* that limits the number of page collections from source for each independent agent.

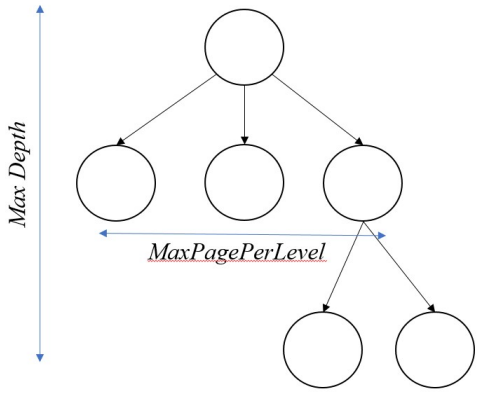


**Fig 2.** Parameters of crawling the pages

The parameters enable the web crawler to collect limited web pages that relate to the target pages of privacy, which is described as terms of use. It also avoid over collection of web pages which are not related to the target pages.

The next step is processing the content of webpage collections in order to extract information that constructs our data structure of privacy for each service provider. In order to extract information, we consider rule based information extraction [7]. The rule-based algorithm has been used in different application, e.g., Mykowiecka et al. [8] describe a rule based information extraction for Polish medical texts. It allows users to extract information based on defined rules. Although rule based information extraction has been widely used in industry, it is not in the interest of academia [9]. There are several studies that show that the rule-based information extraction is widely used in industry [9] since the rules may provide more accurate and reliable results. In this paper, we used rule-based algorithm to extract information from web pages. Each rule aims to collect information based on its criteria.

We generate a set rules as follows:

$$\mathcal{R}_i = \bigcup_{j=1}(\lambda_{i,j} \bigcup_{j=1} TermPolicy_{i,j}\ \omega_{i,j}) \tag{1}$$

where $\mathcal{R}_i$ denotes $i$th rule of rule dataset.

Each rule consists of $j$ items of disjunction logical OR elements (Boolean OR) that consist of a prefix ($\lambda$) and a postfix of a target term ($TermPolicy_{i,j}$) that requires to be extracted.

Each $TermPolicy$ also has a set of $j$ elements that describe the target rule. For instance, in order to extract information on data collection, we may consider any prefix and postfix without any limitations; we may need several terms to be considered for each rule, therefore we have a set of $j$ items of $TermPolicy$s. For example, we may consider the following $TermPolicy$ for data collection on terms of services or data privacy as shown in Table 1.

**Table 1.** $Term$Policy for "*collect*" term

| |
|---|
| … we collect … |
| … we record … |
| … we archive … |
| … we gather … |

Therefore, considering human knowledge to generate a set of disjunction logical terms may not be efficient. Since we have a large dataset of content of webpages, we can generate a language model of all contents. The language model allows us to understand different relation between words in a large number of documents. For instance, it enables the algorithm to understand synonyms and antonyms of a word by processing a large number of sentences. For example, by giving "*collect*" as an input to the language model, it is able to detect all other synonyms which have been used in different web pages. Therefore, it is capable to detect all possibilities of usage of "collect" in Table 1.

In addition, the algorithm of generating the language model does not required training dataset that allows us to generate an unsupervised language model.

Adding each $PolicyTerm$ manually may be time consuming and error prone due to manual understanding of expert to add each $Term$ to the rule. This also may require to define a large number of $PolicyTerm$. The language model uses a vector representation of each sentence and it allows us to find similar words that have been used in the content with the same position in the sentence [10]. When we have a vector representation of each $Term$, then we can understand two different terms by comparing their cosine similarity which is defined as follows. The similarity rate allows us to understand if a word in all sentences has been used widely similar to other word. The following definition shows how we can understand the similarity between $Term_1$ and $Term_2$.

$$TermPolicy_1 . TermPolicy_2 = \|Term_1\|_2 \|Term_2\|_2 \cos(\theta) \tag{2}$$

$$Similarity\ of\ TermPolicy = \cos(\theta) = \frac{\sum_{i=1}^{m} PolicyTerm_1 PolicyTerm_2}{\sqrt{\sum_{i=1}^{n} PolicyTerm_{1,i}{}^2} \sqrt{\sum_{i=1}^{n} PolicyTerm_{1,i}{}^2}} \tag{3}$$

where $TermPolicy_{1,i}$ $1 \le i \le n$ denotes $n$ components of vector representation of $TermPolicy_1$.

Therefore, we consider the following equation that describes all possible $TermPolicy$:

$$TermPolicy = \bigcup_{j=1} Cosine(Vect(TermPolicy_j)) \tag{4}$$

In this equation, $Vect$ represents a vector representation of each $TermPolicy$ and $Cosine$ represents all similar words of the target $TermPolicy$.

We improve Equation (1) by applying a new definition of the $TermPolicy$ as follows:

$$\mathcal{R}_i = \bigcup_{j=1} (\lambda_{i,j} \left( \bigcup_{j=1} Cosine \left( Vect\left(TermPolicy_{i,j}\right)\right)\right) \omega_{i,j}) \tag{5}$$

In this equation, $\lambda_{i,j}$ denotes a prefix of $j$th $TermPolicy$ of $i$th rule. $\mathcal{R}_i$ generates all possible usages of $TermPolicy_{i,j}$ in different terms of services which have been defined by different service providers.

Since a vector representation of each $TermPolicy$ consists of a large numbers of other similar words, we have considered a threshold to limit the number of similar words.

$$\mathcal{R}_i = \bigcup_{j=1} (\lambda_{i,j} \left( \bigcup_{j=1} Cosine_{\vartheta} \left( Vect\left(TermPolicy_{i,j}\right)\right)\right) \omega_{i,j}) \tag{6}$$

where $\vartheta$ denotes a threshold of $Cosine$ similarity of $TermPolicy_{i,j}$.

Each terms of service for $k$th service provider which is denoted as a $Policty_k$, can be defined as follows:

$$Policy_k = \bigcup_{i=1}^{n} R_i \tag{7}$$

where each $Policy_k$ consists of $n$ set of rules ($R$s).

For instance, when we want to compose a policy of "*data collection*" for each service provider, we may follow the following steps to generate policies.

1) use the web crawler to collect all related pages for "terms of services" from different service providers.
2) generate $n$ set of rules for "data collection" when each rule may extract partial information from the website which is describing the terms of services.
3) each rule composes of prefix, postfix and the main condistions. For example, the following rule may extract partial information of "data collection".

$$\mathrm{R}_0 = (*.\cup \ \ Cosine_{\vartheta}(Vect("data\ collection")).*)$$

$$\mathrm{R}_1 = ("we".*.\cup \ \ Cosine_{\vartheta}(Vect("collect")).*\ ")$$

In this example, $\mathrm{Cosine}_{\vartheta}(\mathrm{Vect}(\text{"collect"}))$ will be replaced with a list of similar terms in our corpus as described in Eq. (3). Therefore, the final results might be as follows, if we assume that $\mathrm{Cosine}_{\vartheta}(\mathrm{Vect}(\text{"collect"}))$ returns Table 1 as output.

$$\mathrm{R}_1 = (\text{“}we\text{”}.\ *\ \text{“collect” | “record” | "archive" | "gather" .* ")}$$

Each $i$th $Policy$ of $s$th service provider can be represented as follows.

$$ExistingPolicy_{s,i} = \begin{cases} 1\ if\ \exists\ str\ in\ Policy_{s,i} \\ 0\ if\ \nexists\ str\ in\ Policy_{s,i} \end{cases}$$

where $str$ denotes any output from each $Policy$.

Since repetition of each $TermPolicy$ indicates the importance of the content, we consider the frequency of this repetition by improving previous equation for $ExistingPolicy$ as follows.

$$ExistingPolicy_{s,i} = \begin{cases} TermFrequency(Policy_{s,i})\ if\ \exists\ str\ in\ Policy_{s,i} \\ 0\ if\ \nexists\ str\ in\ Policy_{s,i} \end{cases} \tag{8}$$

Therefore, a vector representation of all defined $Policy$ of $s$th service provider can be shown as follows.

$$ServicePolicy_s = \begin{bmatrix} Policy_{s,1} \\ Policy_{s,2} \\ \vdots \\ Policy_{s,i} \end{bmatrix} \tag{9}$$

The Terms of services for all service providers can be defined as shown in the following equation.

$$TermsOfService_s = \begin{bmatrix} ServicePolicy_1 \\ ServicePolicy_2 \\ \vdots \\ ServicePolicy_s \end{bmatrix}$$

In another word, the terms of service can be defined based on Eq. (9) and Eq. (10) as follows.

$$TermsOfService = \begin{bmatrix} Policy_{1,1} & Policy_{2,1} & \cdots & Policy_{s,1} \\ Policy_{1,2} & Policy_{2,2} & \cdots & Policy_{s,2} \\ & \vdots & & \\ Policy_{1,p} & Policy_{2,p+1} & \cdots & Policy_{s,p+k} \end{bmatrix} \quad (10)$$

where $s$ represents the total number of service providers, $p$ indicates the number of policies for each service provider, and $k$ represents total number of existing policies for each service provider which are defined in Eq. (8).

## IV. Processing Vector Representations of Terms of Services

The vector representations of terms of services for each service provider allows a user to efficiently learn, understand the criteria and simply compare each term against other service providers. Although the current representation of terms of policy might be useful for a user, it is complex when a user subscribes to a large number of services from different service providers.

In this section, we introduce an unsupervised machine learning method that classifies all vector representations of terms of services into different perspectives. Classifying each terms of policy allows a user to compare different criteria based on different perspectives (e.g., data privacy perspective, and data collection perspective).

### A. Understanding Different Prespectives of Terms of Services

We define the following algorithm that considers different perspectives of $TermsOfService$ which is defined in (10). In order to understand what each perspective of $TermsOfService$ means, we use $Policy$ definition in (6).

Given a set of policies which have been composed from different service providers, our goal is to understand different perspectives of terms of services from variety of service providers. For example, it is possible that two service providers have the same definition of "*data collection*" or "sharing *information*".

**Table 2.** Understanding different perspectives of terms of services

| **Different Perspectives of $TermsOfService$** |
|---|
| **Input:** $TermsOfService$ |
| **Output:** Different perspectives of $TermsOfService$ |
| 1: For each policy in $Policy_k$: |
| 2: $SinglePolicyName \leftarrow R_i \; where \; i = 1 \dots n$ |
| 3: $NewPolicyView \leftarrow \text{Permutation(R)}_\delta$ |
| 4: $PolicyName \leftarrow \text{Permutation}(SinglePolicyName)_\delta$ |
| 5: For each policy in $NewPolicyView$: |
| 6: $FinalPolicy \leftarrow K - Mean\ Clustering(NewPolicyView)$ |

As shown in Table 2, the algorithm first, names each single rule in Line 2. For instance, it may name one policy as "*Data Collection*" and name another policy as "*sharing information*".

It combines each rule in Line 3 with a seed of $\delta$ and generate the name of the combination of rules with the same seed ($\delta$) which allows the permutation algorithm generates all possible combinations with the same order. For example, in a simple case of combining only two subsets of "*sharing information*" and *"data collection"*, it might generate a matrix of two columns (features) with $n$ rows that indicate the number of service providers. The algorithm names the combination of the columns by combining the name of each column. The algorithm allows a user to learn which service providers have the same view, or policy in respect to both "*sharing information"* and *"data collection"* terms. The algorithm in a complex case might generate a combination of 3 features, i.e., *"sharing information"*, *"data collection*" and "*warranty*" or in a complex form, a large number of all policies. The total number of $NewPolicyView$ can be calculated as follows:

$$n + P_n = n + n! \quad (11)$$

where $P_n$ denotes all permutation of combination of names and $n$ represents the number of features.

The permutation of all possible combinations may cause an issue. For example, for a large number of features when it may greater than 9, it generates a large data set, 3,628,810 subsets for $n = 10$. Therefore, the clustering algorithm in the next step might have some challenges to compute the clustering for a high dimensions of features. In this case for $n > 10$, we consider a Principal Components Analysis (PCA) [11] that reduce the high dimensions into lower number of dimensions. There are several applications for this method, which have been described in several technical papers, such as [12, 13].

Finally, the algorithm performs an unsupervised machine learning *k-mean clustering* on each subsets of $NewPolicyView$ in Line 6. It returns a number of clusters that has the same definition in each set. In a simple form, the algorithm generates a leverage view of "*data privacy"* for different service providers while it shows how the "data privacy" definitions are the same in the terms of services from different service providers. In another example, for a set of 3 features, the algorithm generates a clustering on a 3-dimension features which allows a user to understand the leverage of three terms, i.e., *"sharing information"*, *"data collection*" and "*warranty*".

The final model also is capable of performing on a mobile cloud computing environment because the client application requires light-weight computation [14, 15, 16]. For instance, a query can be submitted to the model to return a real-time response on a mobile device. The model also can be used in Dynamic Data Encryption Strategy (D2ES) [17] when it encrypts partial data. However, it is required some modification on D2ES that allows the method understand different rules for different classes of the terms of services.

### B. Clustering Algorithm

K-Means [18, 19] algorithm clusters data when it divides $X$ samples into $n$ groups and each group has an equal variance. Each group minimizes error sum of squares (SSE) [20] and it can be defined as follows.

$$SSE = \sum_{i=1}^{n} \left(x_i - \bar{X}\right)^2 \tag{12}$$

where $n$ denotes the number of observations, $x_i$ represents the value of $i$th observation. The mean of all observations is 0. The algorithm requires the number of groups ($k$) in order to find $k$ centric nodes that minimize $SSE$. Therefore, *k-means* can be defined as follows.

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (\|x_j - \mu_j\|^2) \tag{13}$$

where $\mu_j$ denotes the $j$th sample of data set, $n$ denotes the number of observations. The algorithm divides $X$ elements into $K$ disjoint clusters of $C$ when the average of each cluster ($C$) denotes as $\mu_j$[4].

## V. Evaluation

We define a proof-of-concept for the proposed method by generating a $TermsOfService$ with 6 features. In this experiment, we considered a large number of data providers in order to consider a complex case. In addition, one of the feature has been developed as fine grain data that describes different data collection criteria. This complex environment allows us to simulate a real-world challenge for users when they need to understand different criteria as well as different overlap between a large numbers of service providers. Although we apply the proposed method to understand a variety of terms of service from a large number of cloud service providers, it is extensible to any type of problem that needs to understand texts and find any subset of overlap, leverage and pros and cons of different features from different perspectives.

### A. Configuration Setup

We produced a set of features for $TermsOfService$ which is defined in Eq. 10. By considering a high-dimension of input, we showed how the proposed method is capable to process a large dataset. The features that we consider in this experiments are listed as follows:

*1)* Data collection that consist of differents rules $R_{0,0}$ ="Data collection" and a set of collected data where each of them generate an indivisual rule as follows.

*a)* $R_{0,1}$ = IP Address,

*b)* $R_{0,2}$ = Mailing address,

*c)* $R_{0,3}$ = Purchase details,

*d)* $R_{0,4}$ = Company name,

*e)* $R_{0,5}$ = position,

*f)* $R_{0,6}$ = inquiry details,

*g)* $R_{0,7}$ = telephone numbers,

*h)* $R_{0,8}$ = "Credit card/payment information"

*2)* $R_{1,0}$ = "Data sharing"

*3)* $R_{2,0}$ = "Comply with laws and law enforcement"

*4)* $R_{3,0}$ = "Participation in surveys or contests"

*5)* $R_{4,0}$ = "$Restriction\ on\ third\ party\ service\ providers$"

*6)* $R_{5,0}$ = "$Referral\ Service$"

Since this is an ongoing project, we assume that we have extracted all values from website and the method generates $TermsOfService$. We produced a random value for this matrix to evaluate the results. We also used SciKit Learning library[3] for *K-Means* clustering algorithm which is a well-knows library for machine learning. We considered a large number of service providers that consisted of 150 different service providers.

### B. Evaluation Results

Since we do not have ground truth of clustering data from different perspectives, we evaluated the quality of clustering by explaining some sample cases to show how the proposed method enables users to understand different data privacy and terms of services without reading the whole documents from different cloud service providers.

In this experiment, we have considered a variety of permutations of $TermsOfService$ and different perspectives of overlaps between different clusters. Although the clustering algorithm works on high-dimension, presenting higher dimension than 3D might be not clear in the paper because our goal is to evaluate the quality of this clustering.

In this section, we focus on a results which were performed on a 3-dimension[4] and it is also capable to show on a 2D view of different input variables (features).

Fig. 3 shows an experimental results for 150 different service providers when each one generates an array of 3 features as $TermsOfService$ (input variables). In this figure, the size of each shape indicates the frequency of a triple of <"Data sharing, Data Collection', "Third-party restriction">. For example, if $s_1 = [2,3,0]$ and $s_2 = [1,3,0]$, then, $\sum_{i=1}^{3} s_{1,i} = 5$ and $\sum_{i=1}^{3} s_{2,i} = 4$ which means $s_1$ represent a larger size than $s_2$ in Figure 3.

Each data point of a triplet consist of different shapes of "o" or "^" or "*" when it represents the clustering of each feature (input variable) in a 2-dimension, e.g., "o" represents service providers who have a similar policy of "*data sharing*".

Each data point also has been shown with different colors. Each color indicates the output results of clustering algorithm by a given set of input variables (features).

Each sub-figure focuses on different perspectives of data orientations. Each data point can be simply retrieved from

[3] Available at: http://scikit-learn.org

[4] A video of 3D model visualization is available at: https://youtu.be/FnpCw-2RSnM

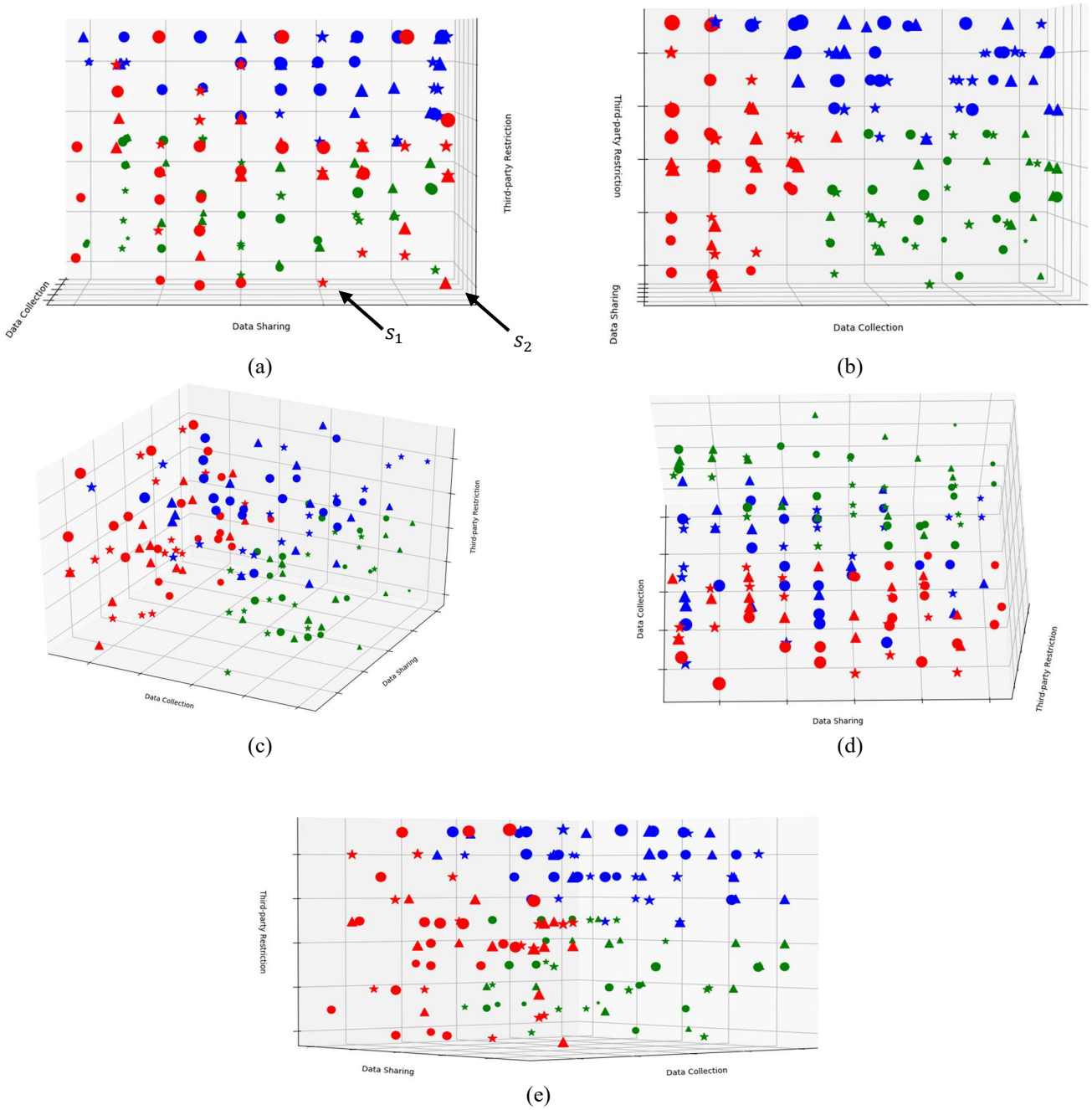


**Fig. 3.** Experimental results for 151 different service providers when each one represents data sharing, data collection, and third-party restricion according to the terms of services [4].

original input dataset, which allows a user to understand the detail of each terms of services for each service provider.

Fig. 3.(a) focuses on data sharing and third-party restriction. Although we have some understanding of data points based on their shape, the clustering algorithm indicates which service provider has the similar policy of data sharing and third-party policy. For instance, although $s_1$ and $s_2$ have different shapes, and both have about the same attitude (frequency of terms), they

**Table 3.** K-Means Centroid values

| | Data Collection | Data Sharing | Third-party Restriction |
|---|---|---|---|
| **Cluster 1** | 7.77 | 4.22 | 4.03 |
| **Cluster 2** | 3.18 | 5.42 | 7.58 |
| **Cluster 3** | 2.23 | 4.12 | 2.48 |

are clustered as one group which means they have some similarity between their policies.

Table 3 represents the centroid of clusters for each feature (variables). As shown in this table, the method is able to classify this 3-dimensions features with different centroid values.

## VI. Conclusion

In this paper, we proposed a method that generates a vector representation of terms of service for different cloud service providers. The proposed method allows users to freely subscribe to different services when users are aware of terms without reading the content. The proposed method is capable to cluster different terms by considering different features and combining different features to understand other perspectives view of different policies. Each feature describes as a perspective view of one dimension. It is capable to be applied to high-dimensions of input variables. The proposed method enables users to monitor their data privacy and find overlap of term of services from different perspectives. For instance, a user may understand what term of services apply to two service providers without reading both terms, or in another example, the user may understand the common term of services that applies to two service providers, e.g., both collecting email address of the users. We developed a proof of concept that shows how the algorithm clusters results. We also shown a visualization of different term of services from different cloud service providers. We plan to extend the proposed method by generating a vector representation of a large number of terms of services from a variety of cloud service providers. It will help any user to understand the term of services, data privacy from majority of cloud service providers without reading their terms of services.

As a future work, we will apply the output of the proposed method which is a machine learning model, to DPM that allows us to securely and efficiently preserve users' data privacy according to the terms of services. Each term of services which is described as a written document can be applied to DPM when DPM scrambles the content based on different class of terms of services.

## References


[1] Edvardsson, Bo, Bård Tronvoll, and Thorsten Gruber. "Expanding understanding of service exchange and value co-creation: a social construction approach" Journal of the Academy of Marketing Science 39.2 (2011): 327-339.

[2] Shirvani, Hussein, and Hamed Vahdat-Nejad. "Storing shared documents that are customized by users in cloud computing" Computing 98.11 (2016): 1137-1151.

[3] Liu, Meng, et al. "Privacy-Preserving Detection of Statically Mutually Exclusive Roles Constraints Violation in Interoperable Role-Based Access Control" Trustcom/BigDataSE/ICESS, 2017 IEEE. IEEE, 2017.

[4] Bahrami, Mehdi, and Mukesh Singhal. "A Light-Weight Permutation based Method for Data Privacy in Mobile Cloud Computing" In Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on, pp. 189-198. IEEE, 2015.

[5] Bahrami, Mehdi, Dong Li, Mukesh Singhal, and Ashish Kundu. "An efficient parallel implementation of a light-weight data privacy method for mobile cloud users" In Proceedings of the 7th International Workshop on Data-Intensive Computing in the Cloud, pp. 51-58. IEEE Press, 2016

[6] Bahrami, Mehdi, Mukesh Singhal, and Zixuan Zhuang. "A cloud-based web crawler architecture" Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on. IEEE, 2015.

[7] Reiss, Frederick, Sriram Raghavan, Rajasekar Krishnamurthy, Huaiyu Zhu, and Shivakumar Vaithyanathan "An algebraic approach to rule-based information extraction" In Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, pp. 933-942. IEEE, 2008

[8] Mykowiecka, Agnieszka, Małgorzata Marciniak, and Anna Kupść. "Rule-based information extraction from patients' clinical data" Journal of biomedical informatics 42, no. 5 (2009): 923-936.

[9] Chiticariu, Laura, Yunyao Li, and Frederick R. Reiss. "Rule-based information extraction is dead! long live rule-based information extraction systems!" In EMNLP, no. October, pp. 827-832. 2013.

[10] Bigi, Brigitte, Yan Huang, and Renato De Mori. "Vocabulary and language model adaptation using information retrieval" In INTERSPEECH. 2004.

[11] Mackiewicz, Andrzej, and Waldemar Ratajczak. "Principal components analysis (PCA)" Computers and Geosciences 19 (1993): 303-342.

[12] Tang, Duyu, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. "Coooolll: A Deep Learning System for Twitter Sentiment Classification" In SemEval@ COLING, pp. 208-212. 2014.

[13] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[14] Bahrami, Mehdi. "Cloud Computing for Emerging Mobile Cloud Apps" Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on. IEEE, 2015.

[15] Mollah, Muhammad Baqer, Md Abul Kalam Azad, and Athanasios Vasilakos. "Security and privacy challenges in mobile cloud computing: Survey and way ahead" Journal of Network and Computer Applications, 2017.

[16] Namiot, Dmitry, and Elena Zubareva. "On one approach to delivering information to mobile users" International Journal of Open Information Technologies 5.8 (2017): 12-17.

[17] Gai, Keke, Meikang Qiu, and Hui Zhao. "Privacy-Preserving Data Encryption Strategy for Big Data in Mobile Cloud Computing." IEEE Transactions on Big Data (2017).

[18] Wagstaff, Kiri, et al. "Constrained k-means clustering with background knowledge." ICML. Vol. 1. 2001.

[19] Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding" Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007.

[20] Ward Jr, Joe H. "Hierarchical grouping to optimize an objective function." Journal of the American statistical association 58.301 (1963): 236-244.