The Journal of Supercomputing (2022) 78:9392–9413
https://doi.org/10.1007/s11227-021-04239-9

Using machine learning to detect PII from attributes and supporting activities of information assets

Yu-Chih Wei[1] · **Tzu-Yin Liao**[1] · **Wei-Chen Wu**[2]

Accepted: 30 November 2021 / Published online: 17 January 2022

Abstract

Since the implementation of the EU General Data Protection Regulation ("GDPR") and similar legislation on personal data protection in Taiwan, enterprises must now provide adequate protection for their customers' personal data. Many enterprises use automated personally identifiable information ("PII") scanning systems to process PII to ensure full compliance with the law. However, personal data saved in non-electronic form cannot be detected by these automated scanning systems, resulting in PII not being able to be accurately identified. We propose a random forest ("RF") approach to detect unidentified PII to close the loopholes. Relevant peripheral information attributes of PII are identified and used in our study for machine learning and modeling to establish a model for detecting PII that otherwise cannot be detected by automated scanners. Our study shows that the F1-measure of our proposed model achieves at least 90%, a higher accuracy rate than that of automated scanners in detecting PII in an enterprise's inventory of information assets. Finally, the results of the experiment in our study show that our proposed model can shorten the time required for detecting PII by 100 times and increase the F1-measure by 2% when compared with the PII detection conducted manually.

Keywords Personally Identifiable Information · Machine Learning · Time Evaluation

✉ Tzu-Yin Liao
t108ab8019@ntut.org.tw

Yu-Chih Wei
vickrey@mail.ntut.edu.tw

Wei-Chen Wu
weichen@ntub.edu.tw

[1] National Taipei University of Technology, Taipei, Taiwan

[2] National Taipei University of Business, Taipei, Taiwan

1 Introduction

For an enterprise to meet its customer's expectations of services, its customers are often required to provide certain information for services to be provided or feedback on the services provided. The information provided by the customers may include their personal and confidential information. In light of the EU GDPR and the legislation on personal data protection in Taiwan, enterprises will be at risk of law violation if they do not have adequate information management infrastructure in place.

For the purposes of information management, an inventory of information assets, including PII, must first be created. Information security risk assessments are then carried out to detect anomalies against items in an inventory. Due to the sheer volume of information assets in an inventory, risk assessments carried out manually often incur huge professional fees, and human errors, such as omissions and mistakes in manual detection of PII, are ultimately inevitable.

To comply with the regulations, enterprises must conduct their own PII inventory inspections, and auditors must also audit PII inventories provided by enterprises. An enterprise may conduct its initial PII inventory inspections based on its standard procedures: firstly, by inspecting file names, followed by inspecting whether the content of a file contains any PII. This procedure takes up a lot of time due to the sheer volume of files, and as a result, it is prone to human errors and omissions during the inspection procedures. Even though there are scanning tools available to help enterprises quickly scan an entire computer, this type of tools can only identify assets directly related to the personal information, such as names, ID numbers and telephone numbers, and they require scanning the content of a file one at a time. It takes up longer time. We propose a new method that can be used to detect PII by using peripheral information in an information asset inventory and information contained in relevant business processes of an enterprise.

The PII detection using machine learning is discussed in this paper, whereby information assets (including PII) contained in an information asset inventory are examined using machine learning. In our study, we focus on the detection of non-electronic personal data that automated personal asset scanning systems cannot detect or accurately identify. We seek to set up an effective detection mechanism with machine learning to improve the accuracy in detecting unidentified PII, so that the cost of consultants and auditors can be reduced, and the quality of risk assessment data is improved. Attributes of information assets that relate to certain business processes of an enterprise or contain key peripheral information are identified and analyzed for our RF approach.

Lastly, the lengths of time required for the manual PII detection are recorded in this paper to compare with those required for the PII detection carried out by the proposed model in this study to evaluate the differences in time efficiency between the two. The PII detection carried out manually may incur human negligence or omissions, such as carelessness or those because of subjective choices, and it takes time to inspect the data one item at the time. If desired results of PII detections can be achieved by a simple machine learning model, it will not only save consultants' time but also allow them to focus on more important issues. It is with this in mind

that we carry out the comparison of time efficiency between the manual and the machine learning detection methods.

The main contribution of this study is conducting PII detections using peripheral information of personal data. Potential PII is identified using machine learning, and reminders are sent to inspectors when there may be identification errors on assets. More than 90% of F1-measure score is reached without data files being scanned. Auditors may conduct further interviews and audits based on results of our PII detections. The efficiency of audits can be greatly increased though quick and accurate anomaly reminders.

In summary, our contributions are as follows:

- A method is proposed where peripheral information of an asset inventory and information on related business processes are used to detect PII, rather than relying on direct scans of asset files. This method scores 90% in F1-measure.
- A comprehensive method is proposed to evaluate model indicators, rather than using only accuracy indicators, such as F1-measure, accuracy, recall and precision. We use behavior detection to calculate the actual time a user spends on answering a question and calculate the F1-measure of the user's answer and evaluate it with the proposed model.

The following sections of this paper are arranged as follows: Sect. 2 describes relevant research on risk assessments, machine learning and time evaluation. Section 3 presents our research methods. Section 4 describes the results of this study. Conclusions and recommendations are given in Sect. 5.

2 Related works

Very few academic works have been carried out on this topic. Some studies on risk assessments, risk predictions and anomaly detection by machine learning are mentioned here for informational purposes. Eminagaoglu et al. [1] studied the information security risk of a human resources department of a logistics company. Eleven machine learning classifiers were used, and 342 samples of human resources data were collected for their study. However, the number of samples used in their study was too low to be sufficient for effective training and testing for machine learning. Zhao et al. [2] used wavelet neural networks ("WNNs") for information risk assessments. Wavelet proposed the WNNs to avoid shortcomings of back-propagation neural network ("BPNN"). The WNNs are known for their strong learning ability and high accuracy. The WNNs' network architecture is also relatively simple, and they can achieve convergence of speed more quickly. The WNN models are more suitable for quantitative than qualitative analyses.

Paltrinieri et al. [3] used deep neural networks (DNNs) to predict risk associated with oil wells. Kaplan et al. [4] defined risk (R) as a combination of answers to these three questions: (1) what could go wrong (scenario s), (2) the likelihood of that happening (probability p), and (3) the severity of the consequences (consequence c).

$$R = f(s, p, c) \tag{1}$$

In Paltrinieri's study, they examined each layer of the risk factors, e.g., how equipment at the oil wells was operated, what accidents had occurred and how experts could have prevented the accidents. These factors were then used as indicators for their risk assessment models. The output of their models represented the probability of accidents occurring.

In the past, risk assessments were commonly carried out by qualitative methods. The shortcomings of these qualitative methods are that they tend to be subjective. In recent years, quantitative research has been carried out on information security risk assessments [5, 6]. Ali Mostafaeipour et al. [5] proposed to use fuzzy algorithm to quantify risks, targeting the vulnerabilities and uncertain elements in the supply chain, including political crisis, demand fluctuations, strategic changes, the financial instability, natural disasters, etc.. He proposed a fuzzy supply chain evaluation model to reduce risk brought by aforementioned elements. Song Shijun [6] proposed to collect expert opinions to generate qualitative indicators with the Delphi method and then to quantify risk with the AHP and the fuzzy methods. The AHP method can be used to assign different weight values to different risks, instead of just using qualitative methods.

The above papers focus on quantifying risk into numerical values and measuring risk across industries or departments. However, research field of the above papers belongs to the stage of risk identification, and almost no research is conducted in the area of automated detection of asset identification. We expect to recognize the correctness of asset recognition through a model with automatic inspection.

There is not as much research about conducting personal information risk assessments using AI. We published two papers about personal information risk assessments. The first paper was about an information security risk assessment model with the privacy consideration. The information security risk assessment ("ISRA") is often dealt alongside with the privacy impact assessment ("PIA"), even though they are carried out separately. In that paper, it discussed the possibility of combining the ISRA and the PIA and conducting them simultaneously while preserving the original method used for the ISRA, rather than carrying out the PIA separately [7]. This paper [7] focuses on the integration of different standard frameworks into a privacy-considered information security risk assessment model. However, there is no discussion of the detection of personal assets identification. In the whole risk assessment process, if not recognized at the asset identification stage, it will lead to the failure of the output in the subsequent risk assessment stage. By combining the inspection model proposed in this study with the evaluation steps in aforementioned paper [7], the whole evaluation mechanism will be improved.

In the second paper [8], it was proposed that the PII data field checking could be carried out by using machine learning, the reasons being that people might fail to fill in their complete personal information during information security assessment processes or personal data risk assessment processes due to intentional or negligent omissions, but an automated checking mechanism could be used to set up reminders for missing personal information for reconfirmation or future audits. The authors marked out potentially anomalous assets, using an unsupervised

learning model by selecting data with higher similarities and data that contain less personal information.

The two papers mentioned above mainly focus on conducting information security risk assessments through machine learning or deep learning. There are other literature studies discussing methods that are used in evaluating risks in information security risk assessments. However, almost none focuses on technology that deals with the PIA relating to personal information. Even though an anomaly detection mechanism on PII data fields was proposed in the second paper [8], it focused on PII assets that had already been identified. This study focuses on how to use machine learning to detect whether an asset is PII, so that an enterprise can reduce the risk of personal information being leaked and can create an accurate inventory of personal data.

In the following sections, we will discuss supervised and unsupervised algorithms relating to machine learning.

K-NNs are one of the most popular machine learning methods for word segmentation [9]. In the IEEE International Conference on Data Mining (ICDM), the following algorithms were voted the most influential methods: C4.5, k-means, SVM, k-NN, naive Bayes and CART [10].

Malini et al. [11] analyzed credit card fraud identification techniques using the k-NN outlier detection. Machine learning, genetic programming, fuzzy logic, sequence alignment, etc., were used to detect credit card fraudulent transactions. Their results proved that the k-NN method was accurate and efficient.

Many unsupervised machine learning algorithms for detecting outliers, e.g., the k-NN, LOF, the anti-k-NN algorithms, etc., have been developed based on the distance-based outlier detection technique developed by Knorr et al. [12]. Knorr defined outliers as objects that deviated substantially from a fixed distance threshold. The distance threshold was further extended to k-NN distance, i.e., the distance of a point to its kth nearest neighbors. Later studies proposed dealing with outliers in a different way by assigning each object an anomaly score representing its outlierness, instead of basing on its k-NN distance [13].

Local anomaly detection algorithms, such as the LOF, are developed by measuring the local deviation of a given point to its k-NN. The local methods are more flexible than the global methods and better reflect the outlierness of each group of data. The LOF compares the local reachability density (lrd) of an object to the local densities of its neighbors. Points that have a substantially lower density than their neighbors (with higher LOF scores) are considered to be outliers [13].

We have discussed supervised and unsupervised machine learning models above. Sathya and Abraham [14] mentioned that if data presented a nonlinear distribution, supervised learning method would improve the effect of a model more effectively. Supervised learning models are therefore used for the experiments in this study. Next, we look for a standard to evaluate the quality of a model. Most evaluating methods are based on statistics. The most used methods to evaluate the quality of a model are confusion matrix indicators (F1-meausre, accuracy, recall, precision, etc.). However, one cannot tell how a model may work in practice by looking at indicators alone. This study proposes to calculate the actual time spent

by a user on a Web page through behavior detection. We will discuss the literature relating to behavior detection below.

A huge amount of research has been conducted in mouse cursor movements to study users' behaviors. Jeremy Goecks et al. [15] used a neural network algorithm to predict the amount of mouse cursor activity as an indicator of users' interests.

Mark Claypool et al. [16] mentioned that recommendation systems could intelligently make recommendations to users that allowed them to be attracted to the recommended things. However, how these recommendations are made behind the scenes must be analyzed through relevant data. The authors of this paper find in their experiment that the length of time spent on a Web page, the number of times spent scrolling a Web page, and a combination of the time and the number of times spent scrolling a Web page are strongly correlated with a user's definite interests in the Web page, whereas a single scroll or single mouse click alone is not effective in predicting a user's definite interest in a Web page.

Laila Paganelli et al. [17] analyzed how tasks were executed by users as recorded by Java Script and the usability of Web pages based on quantitative data, such as the time spent by a user on finding a reference on a Web page and on executing a task, and the total time spent on each Web page when executing a task.

Noboru Nakamichi et al. [18] looked to identify Web pages with low usability, e.g., Web pages that were difficult to use or understand, or pages that responded differently from what were expected. The authors devised a quantitative approach to evaluate user browsing time (in seconds), differences between targeted task completion time and actual time taken for completing the tasks, cursor movements including cursor moving distance (in pixels), mouse click location and cursor moving speed (in pixels per second), the mouse wheel scroll distance (in delta) and eye movements including gaze point sequences, the distance between gaze points (in pixels) and the speed of eye movements between gaze points (in pixels per second). It was found that the data on the eye movements between gaze points could significantly filter out the less informative pages.

Daniel Martín-Albo et al. [19] mentioned that when browsing a Web site, users usually thought of a specific goal (e.g., buying a product), which was a process that consisted of multiple decision-making processes, involving the user moving from an unconscious to a conscious decision-making stage that required the user to focus on a specific goal and execute it. The authors divided participants in their experiment into different groups according to, among others, their mouse click behaviors. They found that users' behaviors before they clicked on a mouse were usually unconscious. The authors compressed the data of cursor movements to save space and use that to train their model. They found their model performed faster with the compressed data than without. The authors suggested that the kinematic theory could be used to distinguish between areas of intentional and unintentional cursor movements and would be useful for researchers and business analysis services.

Moreover, to ensure fairness when comparing models, we remove any time delay due to distraction from the total time spent by a consultant when answering questions. TimeMe.js suggested by Zissman [20] is used in this study. TimeMe.js [20] is a JavaScript program available as an open-source software on Github. TimeMe.js can be used to track user motion trails with precision and calculate the time spent

Table 1 An extract of CkipTagger and Jieba-zh_TW comparison [22]

Tool	(WS) prec	(WS) rec	(WS) f1	(POS) acc
CkipTagger	97.49%	97.17%	97.33%	94.59%
Jieba-zh_TW	90.51%	89.10%	89.80%	–

by a user on a Web page or on a particular element of a Web page. When a user becomes idle (i.e., when his or her mouse cursor does not move or there is no entry from his or her keyboard), TimeMe.js will stop tracking and stop timing. TimeMe.js meets the requirements of this study and is therefore chosen for the purposes of calculating the time spent by users on Web pages or Web elements for the experiment of this study. Its data are then used for our analyses.

3 Methodology

This section analyzes anomaly processing and screening with automated systems in risk assessments and the machine learning models in this study. The time performance of the machine learning models developed for this study and that of the human behavior detection are compared and evaluated. In our study, Chinese text is first preprocessed, segmented into words or phrases and converted into vectors in a vector space model ("VSM"). The segmentation process involves Chinese text processing, the selection of features, the calculation of weighted features and the selection of classification algorithms. An output of the VSM model represents a category that a segmented Chinese character or phrase belongs to. The methodologies used in this study are discussed in the following subsections, namely an introduction to word segmentation, comparisons of word frequency and word similarity methods, descriptions of machine learning methods of k-nearest neighbors (k-NNs), support vector machines (SVMs), random forests (RFs) and synthetic minority oversampling techniques (SMOTE), and lastly a discussion on time calculation method.

3.1 Word segmentation

Jieba is a widely used PHP Chinese word segmentation module[21]. It allows users to add new words and build their own customized dictionaries. This function gives Jieba better accuracy in word segmentation than other traditional word segmentation modules. However, there are other word segmentation tools available these days that offer even better accuracy. The Academia Sinica in Taiwan has released an open-source library, CkipTagger, which implements NLP tools, such as word segmentation (WS), part-of-speech tagging (POS) and named entity recognition (NER). It also offers features such as (1) performance improvements, (2) no word limits on sentences and (3) supporting user-defined recommended and must-have word lists. CkipTagger is tested with the ASBC 4.0 Test Split (50,000 sentences). The results for accuracy, recall and F1-measure of CkipTagger are all about 97%, higher than

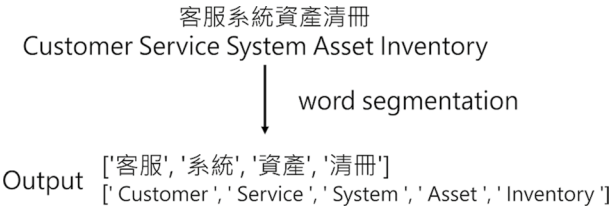

Fig. 1 Word segmentation results

those of Jieba (Table 1). Therefore, CkipTagger is chosen as a tool for word segmentation in this study [22]. Sentences are first segmented into words or phrases in accordance with grammatical usage and meanings by word segmentation tools in this study. Unknown words or phrases are then checked to see if they are Chinese proper nouns or translated European or English proper nouns. If they are not proper nouns, they are then determined to the parts of speech, i.e., the grammatical groups, they belonged to. If the unknown words or phrases are not proper nouns or compound words, the unknown words or phrases are further selected and segmented by a bottom-up merging algorithm. Figure 1 shows the results of the word segmentation in our study.

3.2 Comparisons of word frequency and word similarity methods

The TF-IDF is evolved from the IDF proposed by Sparck Jones (1972, 2004) [23]. The TF-IDF uses statistical methods to evaluate the importance of a word in a document, a file or a corpus based on the frequency of the word appearing in the document. The more important the word is, the higher its TF-IDF score is. The TF is used to calculate the frequency of a word in a document, whereas the IDF the frequency of a word in a corpus. The TF-IDF is obtained by multiplying the two results.

The latent semantic analysis (LSA) is used to minimize global reconstruction errors by finding the best subspace approximation to the original document space. It is based on the singular value decomposition (SVD) and projects the document vectors into an approximated subspace to calculate the cosine value to represent similarity [24].

Multi-words are mainly segmented by two methods: (1) a statistical method and (2) a linguistic method through grammatical rules and sentence structures.

Zhang, Yoshida and Tang [25] used a statistical method in their study to compare the performance of the TF-IDF, the LSI and multi-words in text classification. They analyzed their semantic quality and statistical quality and discussed their application to the Chinese and English text retrieval and classification. Their results showed that the LSI and the multi-words method produced better semantic quality than the TF-IDF, and the TF-IDF produced better statistical quality than the other two methods [26]. A comparison table of the three methods is shown in Table 2. The smaller the value of a method in the table is, the better the method is. As scores are assigned to words or phrases in accordance with their number of occurrences (i.e.,

Table 2 TF-IDF, LSI, and multi-word comparison [26]

Task	TF-IDF	LSI	Multi-words
Chinese information retrieval	1	2	3
Chinese text classification	2	1	2

the frequency) in a document or a corpus in this study, the TF-IDF, which offers better statistical quality, is chosen to calculate word frequency for our study.

3.3 K-Nearest neighbors

The k-NN algorithm is one of the most popular unsupervised machine learning algorithms for text classification [9]. It is also known as a lazy algorithm. It is a local algorithm. It calculates the distance of an object to its neighboring points though Euclidean distance and sets parameter k to decide the number of nearest neighbors to be in the voting process of the algorithms. The classification of a target object is decided by the majority of votes from the voting neighbors. If $k=1$, the object is assigned to its nearest neighboring object to determine its classification.

In this study, a grid search method is used to find the optimal K value for the k-NN. We use F1-measure as a metric for measuring the performance of our k-NN model. Generally, the larger the value of k is, the less sensitive to noise variations a k-NN model will be. However, a large k value is not necessarily suitable for specific tasks that a k-NN model is designated to tackle [27]. A combined method of the grid search and cross-validation is therefore adopted in this study to find the optimal value of K for our model.

In our study, independent data are also used to verify and determine k value in cross-validation. The data are divided into N number of subsets, one set of which is treated as the testing data and the remaining (N-1) sets of which training data. This process is repeated N times, and each subset is verified once.

3.4 Support vector machines (SVM)

Kernel functions are used in SVMs to project features onto high-dimensional space, find decision boundaries among entities and classify entities. SVMs are mainly used to determine kernel functions. It requires prior knowledge or cross-validation of various kernel functions to find the best kernel function suitable for a particular purpose. The most common kernel functions are linear, polynomial, Gaussian kernel and sigmoid kernels. C (cost) and gamma are important parameters in SVMs. C is a fault-tolerant item in SVM and a soft-margin loss function. The larger a C value is, the smaller the fault tolerance will be; the less the support vectors there are, the smaller the hard margin will be. On the contrary, the smaller a C value is, the larger the fault tolerance will be; the more the support vectors there are, the wider the margin will be. Gamma is a parameter that affects kernel functions. The gamma parameter defines how far the influence of a single training

example reaches in feature space, with low values meaning "far" and high values meaning "close" [28], when original data are projected onto it by kernel functions in SVMs. It means that the larger a gamma value is, the closer a neighboring point is; the more effect it will have on its neighbor, the more likely it will overfit.

3.5 Radom forests (RF)

RF is a classification algorithm developed based on decision trees. It uses the bagging method to learn and ensemble trees from training datasets, in which the data are randomly selected. All features are segmented in such a way so that they can be in the Gini coefficient minimum to ensemble a CART tree. Lastly, a voting or an average method is used to further classify features. The former is mainly used for classification works, whereas the latter regression works. The width and depth are the main parameters for RFs. The former means the number of trees in the forest, whereas the latter the maximum depth of the tree [28].

3.6 Synthetic minority oversampling techniques (SMOTE)

After the preprocessing of data, we find we have an imbalanced dataset and a synthetic minority oversampling technique (SMOTE) is used in our study to improve the minority samples in our dataset. The SMOTE generates statistical data in accordance with distances of existing minority samples. It finds five neighboring samples of a randomly selected existing minority sample. The preset values of these five samples are adjusted. A new sample will be formed from these five neighboring samples. This process is repeated until the number of majority samples is the same as the minority samples [29].

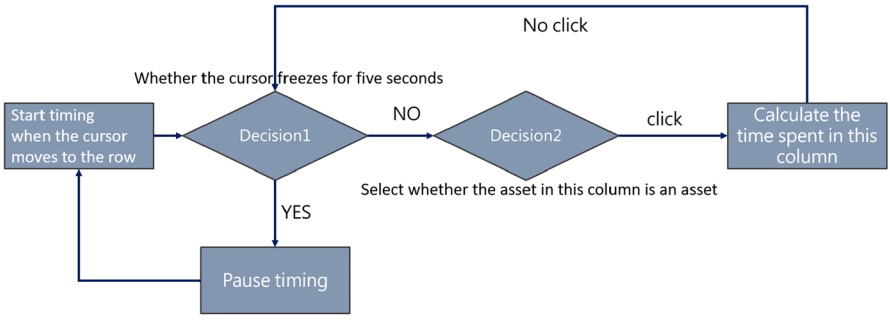

Fig. 2 Time evaluation flowchart

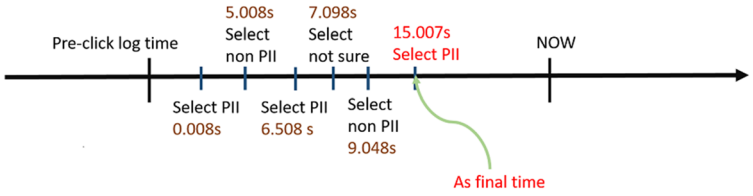

Fig. 3 Collection time diagram

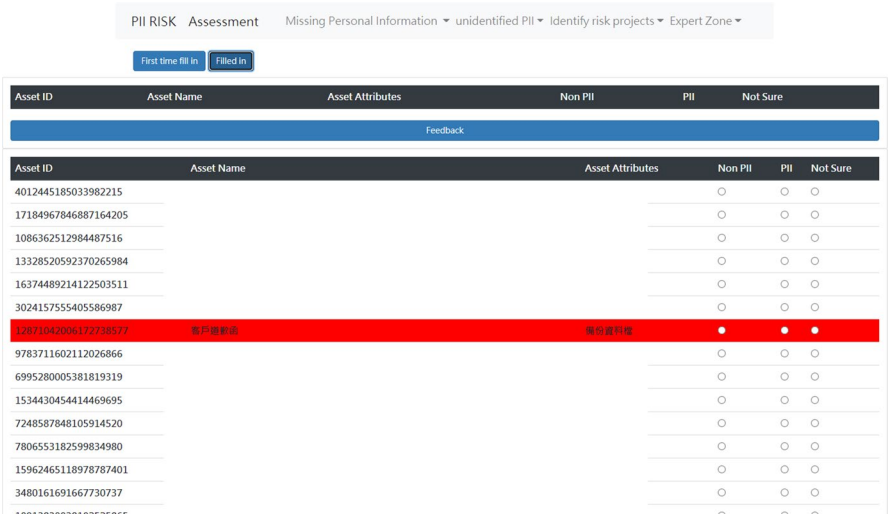

Fig. 4 Screenshot of Web operation

3.7 Time evaluation

Java Script and CSS are used in this study to calculate actual time spent by users on a particular task. A timer starts, when a user moves his or her mouse cursor to an asset in a row, and stops, if his or her cursor or keyboard is idle for more than 5 s (see Fig. 2). When a user clicks on the radio button, the data are sent back to the database using AJAX for storage. (i.e., a log of every action taken by a user). When a user clicks the feedback button, the time last saved on the log as well as the total time saved before last in the system will be added up and presented, as shown in Fig. 3. As to the user interface design, when a user has filled in certain data in a form, the data are presented in a dropped down menu called "Already Done." If a user is unsure about certain answers, he/she can also go the "Already Done" option from the dropped down menu to change the answers and increase the overall time required to answer the questions in relation to the assets. The feedback system is shown in Fig. 4.

4 Evaluation

4.1 Dataset

Our study data are collected from a listed company in Taiwan. The dataset contained 4096 information assets, each of which has ten variables, including asset name, asset attribute, supporting activity, PII-related activity, PII operation, context of usage, supporting product/service, user, related department and service category (see Table 3). The target variable in this study is PII. The ten variables are converted into vectors for the VSM. The CkipTagger is used in our study for word segmentation; the TF-IDF and one-hot encoding are used for the text vectorization, and the k-NN, SVM and RF for word classification. There are two experiments in this study. Consultants and auditors usually conduct their initial examinations through asset names and asset attributes. We first look to classify our dataset through these two features and see what the results may be. Secondly, we add features of supporting activity and supporting product/service in our classification process to see the relationship between the assets. If high F1-measure scores can be obtained in our initial classification in our first experiment, it will save time for enterprises exercises and help detect anomalous assets in their PII detection.

4.2 Experimental environment

Table 4 shows the implementation environment of this study.

4.3 Personnel background

For the research phase of the time evaluation, we invited masters students who had studied ISO27001-related courses and had researched in the field of risk management to participate in our manual inspection process. The expert, who marked answers for the inspection process, is an employee of the company who provided the dataset for our study. The company is a well-known Taiwanese enterprise.

4.4 First experiment—implementation process

There are 4,096 raw information assets in total, all of which are preprocessed. Asset names of the raw information assets are segmented into words or phrases. As our main task is to identify PII of the company's customers, uncertain data or data with missing values are removed from our study data. Data relating to employees or internal correspondence of the company are also removed, as it has negative impact on our machine training and is not considered PII for the purposes of this study. Duplicate words, punctuation marks, stopped words, building floor numbers, numbers, and words with only one Chinese character are also removed. There are 728 words or phrases segmented from the asset names remaining, each of which is then treated as a feature of a column in the inventory, totaling 728 features/columns. There are 47 categories of asset attributes in the raw information data. All the 47 categories of

Table 3 Features descriptions

	Features	Descriptions
1	Asset Name	A brief description of an asset
2	Asset Attribute	A category of an asset, e.g., an application form
3	Supporting Activity	An asset-related business activity
4	PII-Related Activity	An asset-related business activity with a PII-related activity
5	PII Operation	The PII operation phase of an asset-related business activity, e.g., the collection and use of an asset
6	Context of Usage	The context in which an asset-related business activity is used in an enterprise, e.g., for an administrative purpose or for repair or logistic purposes
7	Supporting Product/Service	The product or service in relation to which a business activity is provided
8	User	A customer for whom an asset-related business activity is provided
9	Related Department	A department with which an asset-related business activity interacts, e.g., data processing, data storage, etc
10	Service category	The category of a service in relation to which an asset-related business activity is provided

Table 4 Experimental environment

	Specification and Software Version
CPU	Intel Core i7-8700 CPU 3.2 GHz-4.6 GHz
RAM	DRR4 16 GB 2666 *4 total 64 GB
SSD	Micron Crucial MX500 500 GB SATAIII
System OS	Windows pro 10 2004
Django	2.27
Python	3.7.3
scikit-learn	0.21.3

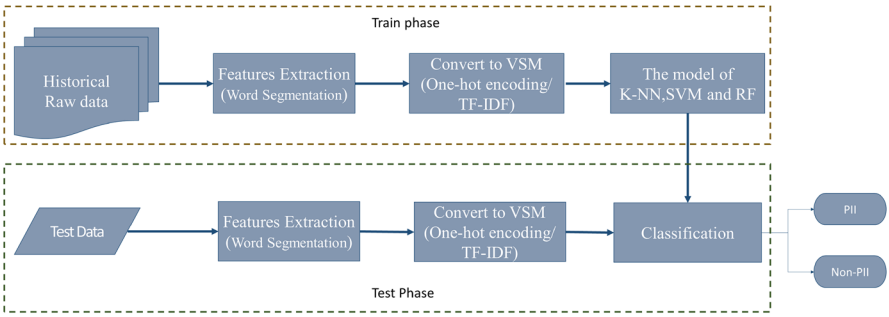

Fig. 5 The process of unidentified PII detection

the asset attributes are treated as one feature. Each category is given a categorical number, representing the numerical value of the entry in the dataset.

After the word segmentation and classification of the asset names and the asset attributes of the raw information data, there are a total of 1,105 features for our model, containing both PII and non-PII information assets. Our model is then used to process the features for PII detection (see Fig. 5).

4.4.1 The K-NN model

After the data are preprocessed, they are then arranged into 1,225 rows of information assets in our inventory. One-hot encoding and the TF-IDF are carried out to train our model for feature extraction. One-hot encoding is carried out on the features of si1,…,si728 which are the result in the word segmentation of asset names. The category of asset attributes is si729. After one-hot encoding, "1" is entered in the inventory indicating a feature existent in an asset name, whereas "0" nonexistent. After the TF-IDF, if the feature exists in an asset name, it is given a TF-IDF score, and if not, "0."

When validating our model, we use the train_test_split in scikit-learn to split the testing and training data. There are 980 rows of information assets in the inventory as training data and 245 as testing data for validation. KNeighbors classifier is used to classify all training data. Cross-validation is used to find the best k value. We use

Table 5 The result of F1-measure in TF-IDF and one-hot encoding

F1-measure	TF-IDF	One-hot encoding
Test data (245 rows data)	83.42%	82.93%
Evaluation (22 rows data)	63.33%	81.82%
Classification time	3.75 s	8.1 s
Test time	38.9 ms	57.5 ms

GridSearchCV to perform cross-validation. The F1-measure, the average and standard deviation are obtained through 15-fold cross-validation, the results of which are compared to find the best k.

In total, 245 rows of data in the inventory are tested with TF-IDF and one-hot encoding in our model. The F1-measures of test data are 82.93% and 83.42% with one-hot encoding and TF-IDF, respectively. Twenty-two rows of data in the inventory are tested with TF-IDF and one-hot encoding in our model. The F1-measures of evaluation are 81.82% and 63.33% with one-hot encoding and TF-IDF, respectively. TF-IDF scores are higher than one-hot encoding in test data, but there is not much difference between TF-IDF and one-hot encoding. However, in terms of the overall result of the model, one-hot encoding scores higher than TF-IDF. One-hot encoding is therefore chosen for our study due to its stability. The results of two methods are shown in Table 5. We also conduct an experiment for hierarchical k-NN model. The results of one-hot encoding k-NN are shown in the next section.

4.4.2 The hierarchical k-NN model

A hierarchical k-NN model, like a single k-NN model, processes data though word segmentation and assigns categorical values to asset attributes. However, a hierarchical k-NN model consists of three k-NN models, namely (1) an asset name single k-NN model, (2) an asset attribute single k-NN model, and (3) a combined k-NN model of the two single k-NN models.

In our study, asset names are segmented into 728 features after the word segmentation for the asset name single k-NN model. The target variable of the asset name single k-NN model is the same as that of the one-hot encoding k-NN model. Categories of the asset attributes are treated as features in the asset attribute single k-NN model. The target variable of the asset attribute single k-NN model is the same as that of the one-hot encoding k-NN model. Features of the combined k-NN model are the features of the other two models combined. The target variable of combined k-NN model is also the same as that of the one-hot encoding k-NN model.

Table 6 F1-measure of hierarchical k-NN model

F1-measure	Asset name	Asset attribute	Combined
Test data (245 rows data)	98.58%	78.53%	82.18%
Evaluation (22 rows data)	77.25%	73.21%	77.25%
Classification time	3.12 s	1.18 s	610 ms
Test time	27.9 ms	19.9 ms	11.9 ms

Table 7 F1-measure of SVM model

F1-measure	SVM
Cross-validation (CV = 10)	76.72%
Test data (245 rows data)	83.27%
Evaluation (22 rows data)	76.19%
Parameters	C:5 Gamma: 0.25 Kernel: rbf
Train/classification time	15 h 11 min 24 s
Test time	45.9 ms

Our results show that the one-hot encoding k-NN model is better than the hierarchical k-NN model, as the F1-measures of the single k-NN for both the cross-validation and evaluation are higher (see Table 6). Consequently, the single k-NN model is used in this study.

4.4.3 The SVM model

As the results for the single k-NN model are better than the other models in our previous experiment, we proceed to use the single k-NN model for our subsequent experiments. As mentioned above, selecting the best kernel function is one of the main tasks in SVMs. A grid search is used in this study to select kernel functions and c and gamma parameters. Table 7 shows part of our SVM grid search results for C ∈{0.1, 0.5, 1, 5, 10, 15, 20, 100, 150, 175, 200, 250, 1000}, Gamma ∈ {0.01, 0.05, 0.1, 0.25, 0.5, 1, 2, 3, 4, 5, 10} and Kernel ∈{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}. When cross-validation is set at 10, we obtain the highest average F1-measure score of 76.72% with a minimum standard deviation, 83.27% is obtained for the partitioned test data and 76.19% for evaluation in our SVM model.

4.4.4 The RF model

The RF model used in our study is also a single k-NN model with a grid search for setting parameters and width values.

The width values are increased from 10 to 200 in increments of 10, and the depth values are set at 5, 10, 15 and 20 to find the best parameters. The results are shown in Table 8. Tenfold cross-validation is used in this study, and the highest average F1-measure score with minimum standard deviation for the tenfold cross-validation is 83.99%, for the partitioned test data 81.25%, and for evaluation 75.49% in our RF model.

4.5 Second experiment—implementation process

We add the data on business processes (supporting activity) and related business services (supporting product/service) in our second experiment, so that assets can

Table 8 F1-measure of RF model

F1-measure	RF
Cross-validation (CV = 10)	83.99%
Test data (245 rows data)	81.25%
Evaluation (22 rows data)	75.49%
Parameters	Depth:20, Width:180
Train/Classification time	2 min 29 s
Test time	339 ms

be classified with more precision by analyzing the relationship between them. Business processes define how a business activity should be implemented and how a department of an enterprise is designated to carry out the implementation of that activity. Data on related business services provide insights on types of customers that an enterprise provides and services and categories of services that a department of an enterprise provides. There are 4,096 sets of original data, which are preprocessed, as those in the first experiment. The two features of supporting activity and supporting product/service are combined during the word segmentation processes. Redundant words, punctuations, numbers, floor numbers and "a" as a single word are removed. One-hot encoding is used in our VSM model, as one-hot encoding performs better in our first experiment. Totally, 930 features are used in our VSM model. In total, 982 sets of data are left after the data are consolidated. Two hundred and forty-three sets of data are different from the first experiment due to the lack of information on related business processes (supporting activity) and related business services (supporting product/service) from some data. It is confirmed that our dataset is imbalanced. Data on PII operation only occupy 30% of the whole dataset. The subsections below compare three classification tools and results of our experiments on sampling (see Fig. 6).

4.5.1 Comparisons of K-NN, SVM and RF models

A grid search and cross-validation are used at this phase. The parameters of the k-NN, SVM and RF models are set within the same ranges as those in the

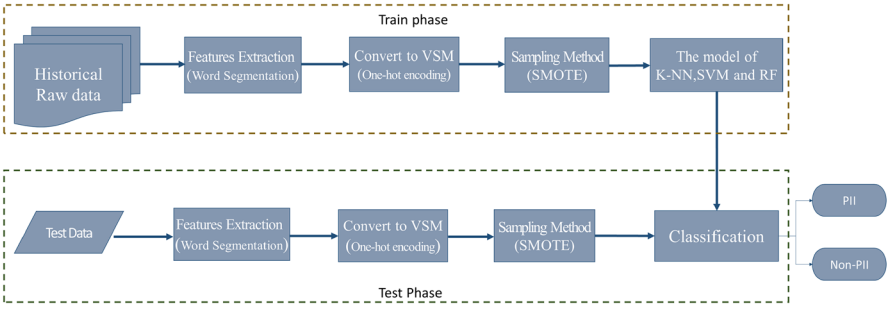

Fig. 6 The process of unidentified PII detection

Table 9 F1-measure and parameters of three models

F1-measure	RF	SVM	K-NN
Cross-validation (CV = 10)	91.96%	81.81%	74.24%
Test data (20%)	82.83%	83.50%	81.08%
Parameters	Depth, 20 Width, 40	C: 15 Gamma: 0.01 Kernel: rbf	K:1
Train/classification time	1 min 46 s	1 h 11 min 24 s	2.71 s
Test time	13 ms	31.9 ms	21.9 ms

Table 10 F1-measure and parameters of three models (with SMOTE)

F1-measure	RF	SVM	K-NN
Cross-validation (CV = 10)	93.57%	95.55%	91.03%
Test data (20%)	94.70%	93.51%	92.95%
Parameters	Depth, 20 Width, 110	C: 20 Gamma: 0.05 kernel: rbf	K: 2
Train/classification time	2 min 10 s	2 h 29 min 12 s	3.35 s
Test time	19.9 ms	61.8 ms	31.9 ms

first experiment. When CV equals 10 in the cross-validation, the highest average F1-measure with minimum standard deviation for a partitioned RF model is 91.96%, for a SVM model 81.81% and for a k-NN model 74.24%. One hundred and ninety-six sets of test data are tested, and the F1-measure for the test data in the RF model is 82.83%, for the SVM model 83.50% and for the k-NN model 81.08%. The results of our k-NN, SVM and RF models showing their highest average F1-measure scores with minimum standard deviation are shown in Table 9.

4.5.2 Comparisons of K-NN, SVM and RF models (with SMOTE)

A grid search and cross-validation are also used in this phase of our experiment. It is mentioned in Sect. 4.5 that our dataset is imbalanced. The SMOTE sampling technique is added in this phase to be used together with our models to compare with the models in our experiments as described in Sect. 4.5.1. The CV value is set at 10 like before, and the highest average F1-measure scores with minimum standard deviation for the RF, SVM and K-NN models are 93.57, 95.55 and 91.03%, respectively. The F1-measures for the 196 sets of the test data in the RF, SVM and K-NN models are 94.70, 93.51 and 92.95%, respectively. The best parameters for the models in this phase are shown in Table 10.

Our experiments demonstrate that there is no significant difference between the results of our two experiments if the SMOTE sampling technique is not used to balance our dataset when features of supporting activity and supporting product/service are added in the word segmentation processes. When the SMOTE sampling

technique is used, the overall F1-measure scores over 90%, an increase in score over 10%. The RF and SVM models (with SMOTE) perform similarly well, both better than the k-NN model. However, the RF model performs faster than the SVM model not only during the training but also during the testing phase. Therefore, the RF model with the SMOTE sampling technique is chosen to be the final model to be used in this study.

4.6 Comparison of manual inspection and model performance in our research

In this section, we compare the timing and F1-measure of model and manual inspection. The personnel described in Sect. 4.3 is the subjects of our experiment. Each of the six participants is asked to examine 100 assets and determine whether they are PII. Finally, the expert is asked to mark the answers of the participants. For the sake of fairness, these 100 items are removed from the training set and the model is retrained for time and F1-measure comparison. Table 11 shows the results after testing. Overall, the average F1-measure is 80.85% and the average time taken is 267,960.0 (ms) for the manual inspection. On the contrary, the F1-measure of the model used in the study is 82.76% and the time taken by the model is only 38.4(ms) (see Table 11).

5 Conclusion

In our study, we use k-NN, SVM and RF models to detect unidentified PII. Two experiments are carried out to classify features. In the first experiment, three models, namely the TF-IDF single k-NN, one-hot encoding single k-NN and the hierarchical k-NN models, are tested with a K-NN model. Our test results show that the F1-measure of the TF-IDF single k-NN model is lower than the one-hot encoding single k-NN model, possibly due to the fact that the data used for our study are not populations, that the frequency of a word appearing in different samples is different and that it has no positive impact on the target variable.

The reason why the F1-measure of the hierarchical k-NN model is lower than the one-hot encoding k-NN model is that the number of asset attribute features

Table 11 Comparison results

	F1-measure (%)	Time (ms)
Person 1	79.71	305,946
Person 2	80.00	234,179
Person 3	65.57	295,060
Person 4	85.00	257,560
Person 5	88.48	423,464
Person 6	83.12	206,822
Person 7	84.08	152,689
Our model	82.76	38.4

used for training is insufficient and that assets that share the same asset attributes may include both PII and non-PII assets. The use of asset names for training, however, results in better detection of PII. As both asset names and asset attributes are used in the combined k-NN model of the hierarchical k-NN model for training, the F1-measure of hierarchical k-NN model is lower than the one-hot encoding single k-NN model.

The F1-measure of our single one-hot encoding k-NN model achieves more than 80% after it is tested by cross-validation and additional data. The results of our k-NN classification model are used on the CVM and RF classification models for experiment. It is found that the three models performed similarly well in our first experiment with the k-NN model performing slightly better.

In our second experiment, we add supporting activity and supporting product/service as features to classify our dataset in the RF, SVM and k-NN models. Due to the lack of information on supporting activity and supporting product/service from certain data, some part of the dataset is removed after preprocessing, which causes imbalance in our dataset. We compare our models with and without the SMOTE sampling technique and find that the performance of the models with SMOTE significantly increases. Not only is the performance better than without SMOTE, but also the performance is improved in the first experiment. We conclude therefore that the additions of supporting activity and supporting product/service as features can help identify data more precisely as to whether they are PII assets. Models in our first experiment already achieve 80% accuracy and can therefore be used to efficiently reduce time required in detecting PII assets. Models in both our experiments can be used to help enterprises to save manpower and time for the purposes. Traditionally, auditors would manually review each asset name and asset attribute in an inventory. The model proposed in our study utilizes simple machine learning to expeditiously determine whether any row of data may contain PII, and it can therefore help reduce the time required for audits and professional cost.

The experimental results show that the F1-measure of the model in this study is improved by nearly 2% compared with that of the manual work and the use of the model in this study shortens the time required for carrying out tasks by 100 times. If the model is tasked to collect a large amount of data and the characteristic selection method is improved, its correctness will certainly be improved in the future. In the future, more data on asset information, business processes and related service information can be collected to retrain models to help enterprises make more precise and relevant decisions, avoid incurring unnecessary fines or penalties and provide better personal data protection for consumers.

Acknowledgements This work was partially supported by Onward Security (No.209A136), National Taipei University of Technology-Beijing University of Technology Joint Research Program (No. NTUT-BJUT-110-01) and Ministry of Science and Technology (NO. 110-2637-H-027-004-).

References

1. Eminagaoglu M, Eren S (2010) Implementation and comparison of machine learning classifiers for information security risk analysis of a human resources department In: 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM), 2010 IEEE, pp 187–192
2. Zhao D-M, Liu J-X, Zhang Z-H (2009) Method of risk evaluation of information security based on neural networks In: 2009 International Conference on Machine Learning and Cybernetics, 2009 IEEE, pp 1127–1132
3. Paltrinieri N, Comfort L, Reniers G (2019) Learning about risk: machine learning for risk assessment. Saf Sci 118:475–486
4. Kaplan S, Garrick BJJRa (1981) On the quantitative definition of risk. Risk Anal 1(1):11–27
5. Mostafaeipour A, Qolipour M, Eslami HJTJOS (2017) Implementing fuzzy rank function model for a new supply chain risk management. J Supercomput 73(8):3586–3602
6. Shijun S (2020) Risk management and countering measurements by computer modeling and simulation technology in the approval and early preparation stages of a large international project. J Supercomput 76(5):3689–3701
7. Wei Y-C, Wu W-C, Lai G-H, Chu Y-CJTJoS, (2020) pISRA: privacy considered information security risk assessment model. J Supercomput 76(3):1468–1481
8. Wei Y-C, Wu W-C, Chu Y-C (2019) (2019) Personally identifiable data field checking using machine learning. International Conference on Frontier Computing. Springer, pp 1789–1796
9. Manning CD, Manning CD, Schütze H (1999) Foundations of statistical natural language processing The MIT Press, America
10. Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Philip SY (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14(1):1–37
11. Malini N, Pushpa M (2017) Analysis on credit card fraud identification techniques based on KNN and outlier detection. In: 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2017 IEEE, pp 255–258
12. Knorr EM, Ng RT (1997) A unified approach for mining outliers Paper presented at the Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research
13. Campos GO, Zimek A, Sander J, Campello RJ, Micenková B, Schubert E, Assent I, Houle ME (2016) On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Min Knowl Disc 30(4):891–927
14. Sathya R, Abraham A (2013) Comparison of supervised and unsupervised learning algorithms for pattern classification. Int J Adv Res Artif Intell 2(2):34–38
15. Goecks J, Shavlik J (2000) Learning users' interests by unobtrusively observing their normal behavior In: Proceedings of the 5th international conference on Intelligent user interfaces, 2000 pp 129–132
16. Claypool M, Le P, Wased M, Brown D (2001) Implicit interest indicators. In: Proceedings of the 6th international conference on Intelligent user interfaces, 2001 pp 33–40
17. Paganelli L, Paternò F (2002) Intelligent analysis of user interactions with web applications In: Proceedings of the 7th international conference on Intelligent user interfaces, 2002 pp 111–118
18. Nakamichi N, Shima K, Sakai M, Matsumoto K-i (2006) Detecting low usability web pages using quantitative data of users' behavior In: Proceedings of the 28th international conference on Software engineering, 2006 pp 569–576
19. Martín-Albo D, Leiva LA, Huang J, Plamondon R (2016) Strokes of insight: user intent detection and kinematic compression of mouse cursor trails. Inf Process Manag 52(6):989–1003
20. Zissman J (2020) TimeMe.js. https://github.com/jasonzissman/TimeMe.js
21. Huiqin W, Weiguo L (2018) Analysis of the Art of War of Sun Tzu by Text Mining Technology. In: 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), 2018. IEEE, pp 626–628
22. Li P-H, Ma W-Y (2019) CkipTagger. https://github.com/ckiplab/ckiptagger
23. Jones KS (1972) A statistical interpretation of term specificity and its application in retrieval. J Document 28(1):11–21
24. Berry MW, Dumais ST, O'Brien GW (1995) Using linear algebra for intelligent information retrieval. SIAM Rev 37(4):573–595

25. Justeson JS, Katz SM (1995) Technical terminology: some linguistic properties and an algorithm for identification in text. Nat Lang Eng 1(1):9–27
26. Zhang W, Yoshida T, Tang X (2011) A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Syst Appl 38(3):2758–2765
27. Ma C-M, Yang W-S, Cheng B-W (2014) How the parameters of k-nearest neighbor algorithm impact on the best classification accuracy: In case of parkinson dataset. J Appl Sci 14(2):171–176
28. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
29. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intel Res 16:321–357