Detecting Security Breaches in Personal Data Protection with Machine Learning

Chu-Hsing Lin*
Department of Computer Science
Tunghai University
Taichung, Taiwan
chlin@thu.edu.tw

Po-Kai Yang
Department of Computer Science
Tunghai University
Taichung, Taiwan
pky0023@thu.edu.tw

Yu-Chiao Lin
Department of Public Health
China Medical University
Taichung, Taiwan
u107070301@cmu.edu.tw

Abstract—**In the age of big data and the Internet of Things, large volume of information, such as medical data, commercial data, or government service data, is generated every second. The protection of personal data to reduce the risk of using information has become very crucial in the field of aforementioned application fields. In this paper, we designed a machine learning model, which can effectively filter out documents containing personal data, and prompt alert to the user. Words and phrases are punctured and marked with part-of-speech tagging and different weights given for different parts of sentence. The pre-trained neural network model and selected features are used to determine whether the sentence contains any personal data. We also compared accuracies among different models of neural network and convolution neural network. In addition, GPU was used to improve the training performance.**

Keywords—machine learning, neural network, conventional neural network, Personal Data Protection Act, intelligent information processing

I. Introduction

The Personal Data Protection Act (PDPA) of Taiwan came to effect in 2012, and was revised in 2015. [1] Later, the PDPA and the enforcement rules were both revised and enacted in March 2016. The later revision of the PDPA refers to foreign regulations such as the EU General Data Protection Regulation (GDPR), [2] the German Federal Data Protection Act, and the Austrian Federal Act concerning the Protection of Personal Data.

The public's understanding of personal data is limited to basic information such as personal identity card number, bank account number, personal password, birthday, and so on. Most people do not aware that they may send a document or article containing something implying or revealing personal information, not to mention the act may violate the law. Therefore, the top priority is to define what personal data is. According to Article 2 of the PDPA, personal information is defined as “the name, date of birth, ID Card number, passport number, characteristics, fingerprints, marital status, family, education, occupation, medical record, medical treatment, genetic information, sexual life, health examination, criminal record, contact information, financial conditions, social activities and other information which may be used to identify a natural person, both directly and indirectly.” In this research, we use this definition in the development of our proposed method and software system. However, the system also allows the users to make their own associations and weight adjustments of personal information to meet their own policy.

In recent years, with the rapid development of computer network and information technology, the community has paid more and more attention to the privacy issue of personal data. In this research, we proposed a mechanism that provides protection to users, reduce their risk of violating the law and reduce the chance of personal data leakage. By the use of the proposed mechanism, it can accurately and quickly scan the document for privacy verification. In the collection, use, processing, transmission of a document, it will prompt a warning if it contains some type of personal data.

Based on the above idea, in this paper we designed a machine learning model, [3-5] which can effectively filter out documents containing personal data, and remind of the user. All the words and phrases will be punctured and marked with part-of-speech (POS) tagging, and different weights will be given for different parts of sentence. Finally, the pre-trained neural network model and selected features are used to determine whether the sentence contains any personal data.

For the experiments, we randomly picked up from the Apple Daily News, [6] popular daily news in Taiwan. We used 220 sensitive-inclusive sentences and 220 sensitive-free sentences as training dataset, and some 60 news sentences as the test data, of which 30 contains sensitive phrase, the other 30 does not.

We conduct a series of experiments to demonstrate the effectiveness of the proposed methods. In this paper, we showed a series of experiments to compare the prediction accuracy and processing time among with or without custom dictionary, using NN or CNN model. We also use GPU to speed up the training work.

The organization of this paper is as follow. In Section 2, we describe the background knowledge of the paper. The design methodology appears in Section 3. In Section 4, we present the experimental details and the results. Finally, we give a conclusion.

II. Background

In 1981, the organization for economic cooperation and development (OECD) established 8 principles for personal data protection, including collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation and accountability, as a reference for its member to promote the protection of personal privacy data.

This research was based on the OECD’s principles and the PDPA regulations in order to study the localized development and realization of privacy impact assessment. [7] Furthermore, this paper provides protection for law-abiding users avoiding them from the risks. For the verification of personal data, we adopt the personal data of all the personnel’s

978-1-7281-5453-4/20/$31.00 ©2020 IEEE

computers and the database of the organization. As long as the system checks or scans a document containing some personal data as defined, a hint or notification will prompt out to alert that the action may involve personal data.

Based up the methodology, we developed a software system for practical use. In addition to basic scanning and inspection, it has the following features:

1) *Support a variety of file formats: such as e-mail, Office files, web pages, document files etc.*

2) *GPU-based acceleration: to improve computing performance when training dataset is large.*

3) *Resilience: allow users to define and set personal data according to the policy of different organizations.*

By using GPU-based computing, the software system can reduce the inspection time for the personal data processing work handled by the administrative and clerical staffs, and improve the efficiency of their work. For the basic system environment considerations, we use free open source software to reduce the basic system and environment configuration time. In addition, the software can be simply installed or a package type loading in users' computers, which makes it easy to operate for end users.

The software system can be divided into the server side and the client side, and the server is in charge of data validation from the client's. Conceptually, this system is similar to anti-virus software. When the user clicks on a folder, it scans its contents simultaneously. However, different from anti-virus software, we take the personal data defined by the server as the virus signatures. After scanning, the file path and the user's IP are sent to the server. It is notable that only the one who is authorized to use the database can upload the data. It is a way to lower the risks and implement the spirit of the PDPA.

A. Word Segmentation

In Chinese language, a word is the least meaningful and freely usable unit. Distinguishing words on hand is the primary goal for all of language processing system, such as machine translation, language analysis, language understanding, and information extraction. Therefore, automatic word segmentation (tokenization) system becomes an indispensable technology for language processing. Basically, a great majority of automatic word segmentation systems use the words and texts included in the dictionary to compare and find possible words. However, due to the ambiguous segmentation results, most of Chinese word segmentation systems put emphasis on solving this problem, and only few of them deal with words that are not in dictionary.

In all of well-known Chinese word segmentation systems, two of them are highly recommended: the CKIP [8] and the Jieba. [9] However, the CKIP, the Chinese Knowledge and Information Processing, is unstable that the data returned may be truncated, and the API is not good enough. Moreover, it is not open-source software, we cannot modify properly for our use. Consequently, we choose the Jieba as the tokenization tool, which is open-source and has custom-built lexicon for word segmentation. It is expected that more outstanding developers will devote themselves to developing word segmentation system in the near future, and the functions will become more comprehensive.

B. Neural Network

Neural network (NN for short) is a mathematical model that is proposed to simulate the neural network of the human brain. The theory of neural network was invented in 1950s, when scientists developed a neuron model called perceptron to investigate how the human brain works. In the 1980s, when the Hopfield neural network was proposed, the studies of neural networks became blooming. Today, with the immensely increase in computation speed, the application of neural network is almost everywhere. [10-11]

To understand how an NN works, we first depict a single neuron as shown in Fig. 1. In the figure, we let X=[X1, X2, …, Xn] be the inputs to the neuron and X=[W1, W2, …, Wn] be the weights, and b the bias. S is the summation operation to sum up the bias and the multiplication of the inputs and the associated weights. F is the activation function, which converts the result of S to the required output using a linear or nonlinear function and then is the output of the neuron.

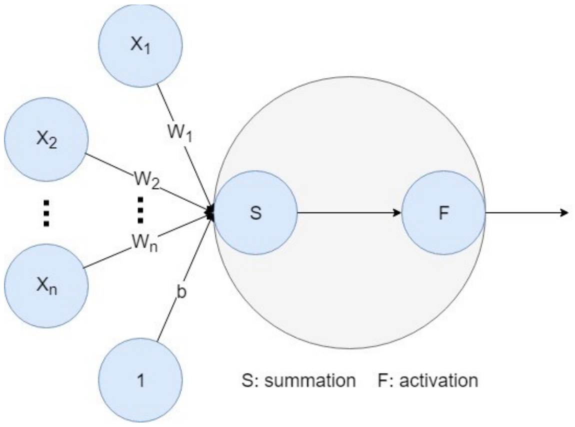


Fig. 1. A single neuron

An NN is composed of various types of neurons mentioned above, which can be divided into the input layer, the hidden layer, and the output layer. Different training and learning algorithms enables the neural network to output the results we expect. A set of data from the input layer is multiplied by the weight values and then transfer to the neurons of the hidden layer. Through the activation function, it is then transfer to the neuron of the output layer for processing, and finally a set of trained data is obtained from the output layer. An NN with one hidden layer is shown in Fig. 2.

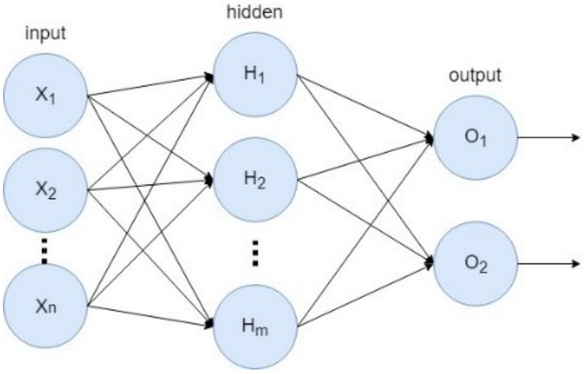


Fig. 2. An NN model with one hidden layer.

As shown in Figure 2, it has n neurons in the input layer, m neurons in the hidden layer, and two in the output layer, respectively. The links indicate the weights between the input layer and the hidden layer, and the weights between the hidden layer and the output layer, respectively. An activation function is used between layers to perform the operation, and the activation function can be linear or nonlinear. The activation functions used in this research are the Rectified Linear Units (ReLU) and the Softmax function. We used the ReLU function in between the input layer and the hidden layer, and the Softmax function between the hidden layer and the output layer. ReLUs are the most commonly used activation function in neural networks, especially in CNNs. The ReLU function we used is defined as Eq. (1):

$$f(Z) = \max(0, A \cdot W^T + b) \tag{1}$$

where Z=A·W^T+b, W is the weight, A is the input signals, and b is the bias. This function zeroizes the calculated value which is negative. In this way, the accumulated error of the calculated result can be avoided. In addition, Softmax function takes an un-normalized vector, and normalizes it into a probability distribution. After applying Softmax, each element falls in the interval [0, 1] and sum to 1. Since we have two neurons in the output layer, we use Softmax function to categorize the result into two classes, 0~0.50 and 0.51~1, respectively. The standard Softmax function on a vector Z is given by the standard exponential function on each coordinate, divided by the sum of all the coordinates, so the output coordinates sum to 1. The Softmax function g(zj) we used is defined as Eq. (2).

$$\mathrm{g}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \quad j = 1, \dots, K \tag{2}$$

An NN operates in the manner described above. There are many popular types of NNs such as back-propagation networks, Hopfield networks, and radial basis function networks. In this paper, we adopted the back-propagation network, it uses back-propagation learning algorithm to update the weights.

The algorithm first feeds forward the signals and calculates from the input layer, go through the hidden layer, and then to the output layer. Next, it calculates output error based on the predictions and the target value. Then it back propagates the error signals to adjust the weights between layers to minimize the total error. A proper function for error measure is required. The forward calculations performed are as follows: The signals a_i from the input layer are multiplied by a set of weights w_ij connecting between the input layer and the hidden layer. These weighted signals are then summed and combined with a bias b. This calculation forms the pre-activation signal $z_j = b + \sum_i a_i w_{ij}$ for the hidden layer. The pre-activation signal is then transformed by activation function f of the hidden layer to form the activation signal, that is, $a_j = f(z_j)$ In a similar fashion, the hidden layer activation signals a_j are multiplied by the weights w_{jk} between the hidden layer and the output layer, a bias b is added, and the resulting signal is transformed by the output activation function g to form the output, a_k, $a_k = \mathrm{g}(z_k)$, of the NN. Note that one common requirement for both of the error measure function and the activation function is differentiable.

C. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is a model based on neural network. CNNs are widely used in image recognition, recommendation systems, and natural language processing. In CNNs, there are two more layers than the conventional neural network, which are the convolutional layer and the pooling layer. [12]

- Convolutional Layer: Each convolutional layer consists of several convolutional units. The convolution unit locally extracts the data of the input arrays and puts the extracted arrays into the NNs for analysis. Weights can be added during the extraction process to change the original data. Contrast to conventional NNs, the convolutional layer enables the neural network to read more than one features at a time. After the input arrays is extracted by the convolution unit, which retains and highlights the features, the range of the input arrays is reduced and then the amount of computation is also reduced.
- Pooling Layer: In the convolutional layer, in order to increase the diversity of features, the features are randomly disordered and rearranged to increase the thickness of the input. On the other hand, the pooling layer is to organize and generalize the features, compress them, and preserve the main features in the process of compression. In this way, the feature range can be further narrowed, and the main features are highlighted.

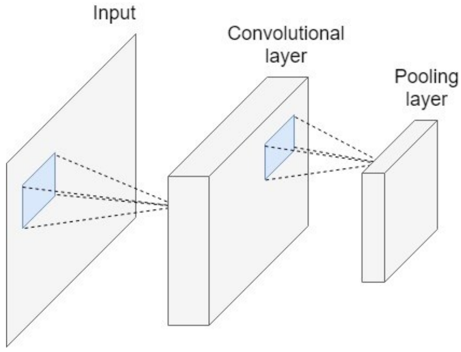


Fig. 3. convolutional layer and pooling layer

Fig. 3 shows the relationship between the convolutional layer and the pooling layer. The input array is processed through the convolutional layer to reduce the length and width of the arrays, but the thickness is increased. The pooling layer ensures that the main features are preserved while the length and width are further reduced. Therefore, in analyzing data with convolutional layer and pooling layer, the network is no longer one neuron for one feature, but one neuron for multiple features. In addition, the weights are changed to find the correlation between features, and greatly improve the accuracy of prediction.

D. GPU

Graphic Processing Unit (GPU) was introduced by NVIDIA in 1999. When the GPU was introduced, it contained more processing units than the CPU, enabling it to perform much more operations in multimedia processing. Today's top CPUs have only four cores or six cores, at most eight or even twelve for computing, but the average level of GPUs contains thousands of smaller and more efficient cores to optimize multitasking execution for simultaneous processing. [13-14]

Comparing to the concurrent computing of CPU, parallel computing is used by GPU. On the CPU, a problem is broken down into a series of discrete instructions, instructions must be executed one after the other, and only one instruction can be executed at a time. Parallel computing improves in many important details. To run with multiple processors, a problem can be broken down into discrete instructions that can be resolved simultaneously. Each part is further subdivided into a series of instructions, each of which can be carried out simultaneously on a different processor.

CUDA is a parallel computing architecture invented by NVIDIA. It uses the great processing power of GPU to increase tremendously the operation performance. GPU contains several streaming multiprocessors (SMs), and each SM contains many stream processors (SPs). Each SP can handle operations independently to achieve parallel processing of data. CUDA is of good performance at dealing with matrix operations, especially for image processing and video processing. It can significantly accelerate the computing speed for solving problems. A neural network model composed of basically large number of neurons in the layers. Each neuron is responsible for independent operations. Hence, we can treat neurons as images and use parallel matrix operations to improve the performance.

III. The Methods

In order to promote the software implementation of the PDPA, we propose a feasible framework for privacy protection. Currently, when we wrote the paper, there is no software system that can accurately determine whether there some personal data in a document. In convention, a feasible way to design such a system is first to build a database of person data and then search and compare with the database when encounter a document. However, this method is limited to specific companies, institutions or schools that have their own database for the predefined type of person data. Since the database only contains the predefined type of person data, it will leak the other types of personal data that do not appear in the database. Due to the diversely predefined personal data, for example, in some application gender is recognized as a part of personal data and some are not. Therefore, in this paper, we use neural network in machine learning to build the model for person data and to solve the above limitation and the problem of long processing time.

In this section, we describe how to construct the word segmentation system, design and training the neural network and the convolution neural network model.

A. The System Architecture

Information features can be used to training the machine learning model only after the obtained text has been processed by word segmentation. After one word in the text is segmented or tokenized, a weight value in the range of [0, 1] is assigned, depending on the parts of speech it belongs, such as nouns, verbs, adjectives, and so forth. The weight values are then put into the neural network model that we have trained on the server side to determine whether the incoming text contains a personal data. First, we briefly describe the client-server paradigm of the proposed system architecture.

- Server side: The server consisted of three components: the master server (MS), the tokenization server (TS) for word segmentation and the prediction server (PS) of machine learning. The MS server is responsible for receiving documents and communicating with the client. The TS breaks down the received document to sentences, segments each sentence to a series of words, and categorize the words with proper parts of speech. Afterward, each word is assigned to a weight values according to the parts of speech it belongs. The PS server is responsible for using the trained model to determine whether there exists a personal data in the sentence.
- Client side: At the client, it is mainly responsible for obtaining files to process. The types of documents are roughly divided into four categories, as shown in TABLE 1.

TABLE I. Document categories and Retrieval methods

Categories	Retrieval
Text documents	BufferedReader
Office documents	POI
Web page	Jsoup
E-mail	Java mail

The client automatically captures the transmitting files in user's computer, check the data sent to the webpage by the user, scanned the contents of the mailbox, and the obtained documents will be signed by digital signature. In this way, after the sever finding a personal data, it can confirm that from which client the person data is extracted, and log the information for future reference. The signed documents will be transmitted to the server side, and it will perform the procedure as aforementioned steps.

B. Words Segmentation Process

There are several kinds of Chinese words segmentation system, such as CKIP, Jieba, and so on. In this paper, we chose the Jieba system for words segmentation. The main reason is that it is open source code and we can change the code according to our needs. Most importantly, with a custom dictionary in it, we can set the required tokens and their associated parts of speech. The core of the Jieba system is the dictionary, which contains three components: word, weight, and parts of speech.

To analyze a sentence, the Jieba system takes it as input and ongoing a process of regular expression. It first uses the Trie tree to establish a directed acyclic graph (DAG) for the sentence. Following to the DAG from the beginning, it uses the HMM Viterbi algorithm to determine whether a new word appears, when the word is not in the dictionary. Taking the sentence “他即將來臨時” as an example. It will start from "他" and inquire the dictionary to find out if there is a word " 他即", and then continue to search from next to the word "即 ". On continuingly this way, and according to the dictionary, the weighted DAG, with weight values in each node, is created. From the weighted DAG, we can finally determine that the above sentence is segmented as “他/即將/來臨時”.

In the Jieba dictionary, a different weight value, between 0 and 1, is assigned to each parts-of-speech. We can assign the weight value 1/0 to a parts of speech that has been determined definitely to be “personal-data”/”non-personal-data.” The remaining parts of speech are assigned a weight between 0.1~0.9. If the possibility that a word is treated as a personal data, the assigned weight will be higher. For example, a noun, with high probability to consist in a personal data, is given a weight of 0.8, and a verb is least likely to and is given a weight of 0.0. Numerals or quantifiers may be associated with salary or phone number and is given a weight of 0.6, respectively.

The weights of the parts of speech in the dictionary are shown in TABLE 2.

TABLE II. WEIGHTS OF PARTS OF SPEECH

Parts of Speech	Weights
Noun	0.8
Verb	0.0
Adjective	0.2
Quantifier	0.6
Numeral	0.6
Preposition	0.2

For experiment, we retrieve randomly 500 non-repetitive sentences from the Internet daily news as dataset, and use 440 sentences of them as training data and the 60 sentences left as testing data. Among the training data, 220 sentences contain personal data such as, name, ID-card number, telephone number, etc., and the remaining 220 sentences not including any personal data. We use the Jieba system to extract the parts of speech of each of the 440 sentences, and set the weights. Further, the weights of a sentence are sent to the neural network for training. The extracted features, the weights of the parts of speech, for each sentence and their description are shown in TABLE 3.

TABLE III. FEATURES AND DESCRIPTION FOR A SENTENCE

Weights as Features	Features Description
The largest weight in the sentence	If it equals 1, use it as a feature, the sentence must include a personal data
The 2nd largest weight in the sentence	If the largest weight is not 1, use the top 5 weights as features to determine the possibility of containing a personal data for a sentence
The 3rd largest weight in the sentence	
The 4th largest weight in the sentence	
The 5th largest weight in the sentence	
The average value of non-zero weights within a sentence	This feature help determine the extent to which the suspicious data approaches a personal data
The ratio of non-zero weights to the whole sentence	This feature help determine the proportion of suspicious data in the entire sentence

C. *The Models Construction*

In this research, we design a neural network (NN) model and a convolution neural network (CNN) model to implement the proposed methods, respectively. The NN model has a 7-dimension input layer and a 3-layer hidden layer, with 512, 1024 and 256 neurons, respectively, and the number of neurons in the output layer is 2. The ReLU activation function is used after the input layer and first two of the hidden layers. The Softmax activation function is used between the last hidden layer and the output layer. A NN model is shown in Fig. 4.

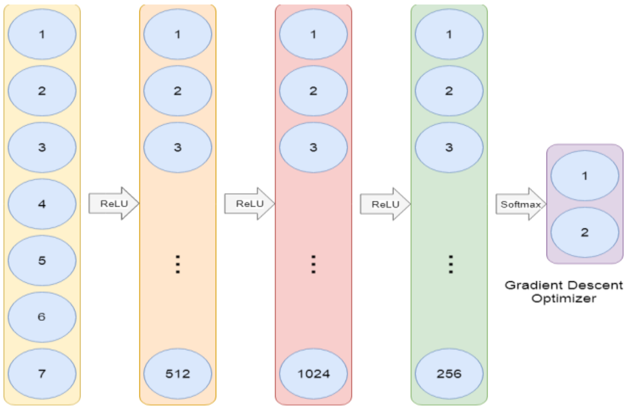


Fig. 4. A NN model for experiments.

The CNN model is shown in Fig. 5. It composes of an input layer, a convolution part and a NN part. The input layer is of 8 dimensions. The convolution part has two convolution layers (C1 and C2), two pooling layers (P1 and P2), and the NN part has an input array (IL), one hidden array (HL), and an output layer (OL). The ReLU function is used as activation function between the layers, except that the Softmax function is used between the last hidden layer and the output layer. Dropout was added to the last hidden layer. This allows the CNN model to ignore a portion of the neuron weights to avoid the overfitting problem.

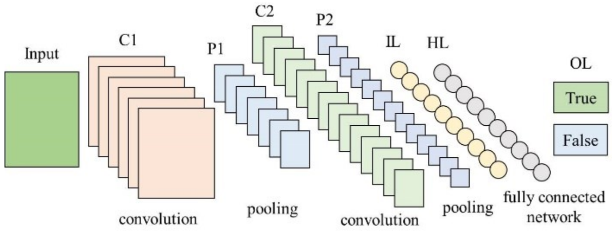


Fig. 5. CNN model for experiments.

IV. EXPERIMENTAL RESULTS

We conduct a series of experiments to demonstrate the effectiveness of the proposed methods. In this section, we show the five experiments to compare the prediction accuracy and processing time among with or without custom dictionary, using NN or CNN model. We also use GPU to speed up the training work and use batch normalization to improve the prediction accuracy.

The experimental environment is depicted as follows. We use the open source software operating system: Ubuntu 16.04 LTS. The CPU is Intel Core 2 Quad CPU Q8200 @2.33GHz*4 with 4G RAM. The GPU graphics card specification is: GeForce GTX650 Ti/PCle/SSE2.

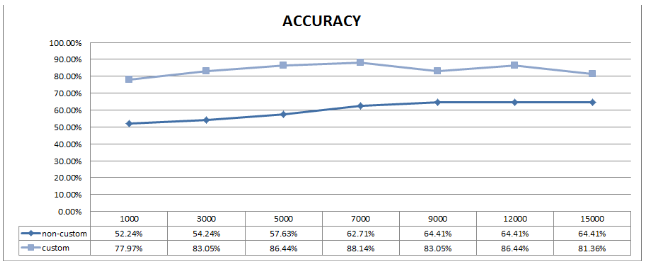
A. *Custom Dictionary.*

For the testing dataset, we used 60 sentences picked randomly from the Apple Daily News. 30 of them contain personal data, including name, ID number, address, telephone number, etc. The other 30 sentences do not include any personal data. We run two cases to calculate the accuracy, one is with the custom dictionary and the other is without. The result is shown in TABLE 4.

TABLE IV. ACCURACY COMPARISON W/ VS W/O CUSTOM DICTIONARY. (%)

No. of Training	1000	3000	5000	7000	9000	12000	15000
W/O	52.24	54.24	57.63	62.71	64.41	64.41	64.41
W/	77.97	83.05	86.44	88.14	83.05	86.44	81.36

The accuracy trend for different number of training is also shown in Fig. 6. It can be clearly seen from Fig. 6 that without any custom data, the accuracy can reach 64.41% when the number of training reaches 9000. In the simple use of the parts of speech as training materials, it can achieve an accuracy rate of 64.41%, which is a good result. On the other hand, if we use a custom dictionary, the accuracy rate was as high as 88.14% when the number of training reached 7000. However, we can see that the accuracy rate drops slightly after more than 7000 times, which may be the result of an overfitting problem. From the above experimental results, it also shows that, in the NN model, we can obtain the highest accuracy at the training number of about 7000 times.



	1000	3000	5000	7000	9000	12000	15000
non-custom	52.24%	54.24%	57.63%	62.71%	64.41%	64.41%	64.41%
custom	77.97%	83.05%	86.44%	88.14%	83.05%	86.44%	81.36%

Fig. 6. The accuracy trend for different number of training

B. *GPU Speedup.*

In the above experiment, for the NN model of 3-hidden layers, with neurons size of 2048*16384*1024, by using the CPU, it requires up to three hours of training time. Since the training time is too long, we use the GPU to accelerate the operation. The speedup rate is defined as the ratio of the CPU execution time to the GPU execution time.

The training time in seconds and the speedup rate for different number of neurons in the hidden layers are shown in TABLE 5.

TABLE V. THE TRAINING TIME AND THE SPEEDUP RATE FOR DIFFERENT NUMBER OF NEURONS.

NN Models	512*1024*256	2048*16384*1024
CPU	206.3 (s)	11977.25 (s)
GPU	56.76 (s)	1227.65 (s)
Speedup Rate	3.6	9.8

It can be seen from Table 5, for the NN model of 512*1024*256, the speedup rate is about 3.6, and it is increased to about 9.8 in the case of 2048*16384*1024 model. From the results, we can find that since the amount of neurons is not large enough in the model of 521*1024*256, the speedup rate is not obvious. However, in the model of 2048*16384*1024, the speedup rate can be significantly raised.

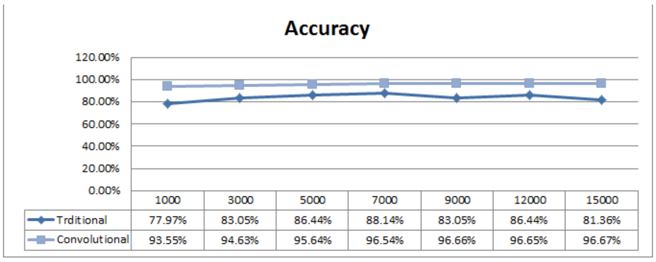
C. *NN vs. CNN*

In this experiment, we compare the prediction accuracy between the NN and the CNN models with different number of training. TABLE 6 shows the results.

TABLE VI. ACCURACY COMPARISON FOR NN VS. CNN. (%)

No. of Training	1000	3000	5000	7000	9000	12000	15000
NN	77.97	83.05	86.44	88.14	83.05	86.44	81.36
CNN	93.55	94.63	95.64	96.54	96.66	96.65	96.67

Fig. 7 sows the trends of accuracy between NN and CNN models. From Fig. 10, it can be clearly seen that when the CNN model is used, the accuracy is already as high as 93.55% when the number of trainings reaches 1000. Moreover, the accuracy rate reached 96.67% when the training number is increased to 15000 times. Compared with the NN model, the CNN model effectively improved the prediction accuracy. In addition, from the experiment, we also found that the false acceptance rate (FAR) approached to 0% and the false rejection rate (FRR) is about 3.45% for the CNN model. Note that after submission of this research we also built a CNN model with fully connected layer of 64*128*32. By conducting a series of experiments, we found that it obtained 96.66% of accuracy on 400 times of training.



	1000	3000	5000	7000	9000	12000	15000
Trditional	77.97%	83.05%	86.44%	88.14%	83.05%	86.44%	81.36%
Convolutional	93.55%	94.63%	95.64%	96.54%	96.66%	96.65%	96.67%

Fig. 7. The trends of accuracy between NN and CNN models.

D. *CNN and GPU Speedup*

We run on the two cases of different number of neurons, with 512*1024*256 and 2048*16384*1024, respectively, in the CNN model, and compare their speedup rates. The training time (in seconds) and the speedup rates for different CNN models are shown in TABLE 7.

TABLE VII. THE TRAINING TIME AND THE SPEEDUP RATE FOR DIFFERENT NUMBER OF NEURONS.

NN Models	512*1024*256	2048*16384*1024
CPU	312.2 (s)	12106.15 (s)
GPU	89.2 (s)	1256.71 (s)
Speedup Rate	3.5	9.6

It can be seen that in the CNN model of 512*1024*256, the speedup rate is about 3.5, and in the model of 2048*16384*1024, it is increased to about 9.6.

V. CONCLUSIONS

In recent years, with the rapid development of computer network and information technology, the privacy issue and personal data protection have become very crucial research topics. In this paper, we designed a machine learning model which can effectively filter out documents containing personal data and prompt the user. All the words and phrases will be punctured and marked with part-of-speech (POS) tagging, and different weights will be given for different parts of sentence. The pre-trained neural network model and selected features are used to determine whether the sentence contains any personal data.

In this paper, we developed a neural network model with a number of neurons of 2048*16384*1024. After training, the accuracy rate of whether personal data is included in a sentence reaches 88.14%, and we use GPU to improve the training speed, and we can train a set of neural network models in a very short time. In a neural network model with a neuron number of 2048*16384*1024, an acceleration effect of 9.8 times can be achieved.

In addition, we use the Convolutional Neural Network (CNN). It has two more layers than the neural network, namely the convolutional layer and the pooling layer, which enhances the degree of correlation between weights. Then dropout is added to prevent overfitting. We found that convolutional neural networks can be used to improve accuracy.

As to the accuracy comparison between NN and CNN models, the accuracy rate for CNN reached 96.67% when the training number is increased. Compared with the NN model, the CNN model effectively improved the prediction accuracy. Besides, with GPU, we found that in the CNN model of 512*1024*256, the speedup rate is about 3.5, and in the model of 2048*16384*1024, it is increased to about 9.6.

Acknowledgment

This research was partly supported by the Ministry of Science and Technology, Taiwan, under grant number MOST 107 - 2221 - E - 029 - 005 - MY3. The authors are also grateful to K. Y. Lee, T. C. Peng and C. S. Tsai for their assistance in the experiments and preparation of this paper.

References

[1] Taiwan Data Protection, https://iclg.com/practice-areas/data-protection-laws-and-regulations/taiwan

[2] EU GDPR, https://eugdpr.org/

[3] P. Louridas, C. Ebert, Machine Learning, IEEE Computer Society, pp.110-115, 2016.

[4] Henrik Brink, Joseph W. Richards, Mark Fetherolf, Real-World Machine Learning, Manning, 2017.

[5] Ethem Alpaydın, Introduction to Machine Learning, 2nd Ed., Cambridge, Mass.: MIT Press. 2010: 250. ISBN 978-0-262-01243-0.

[6] Apple Daily News, https://tw.appledaily.com/

[7] Roger Clarke, “Privacy Impact Assessment: Its Origins and Development,” Computer Law & Security Review, Volume 25, Issue 2, 2009, pp. 123–135.

[8] CKIP Lab, http://ckip.iis.sinica.edu.tw/

[9] Jieba, https://github.com/fxsjy/jieba

[10] J. A. Anderson, An Introduction to Neural Networks, Prentice Hall, 2003.

[11] Martin Hagan, Neural Network Design. PWS Publishing Company. 1996. ISBN 7-111-10841-8.

[12] Keiron O’Shea1 and Ryan Nash, An Introduction to Convolutional Neural Networks, arXiv:1511.08458v2, Dec. 2015.

[13] F. Silber-Chaussumier, A. Muller, R. Habel, “Generating Data Transfers for Distributed GPU Parallel Programs,” Journal of Parallel and Distributed Computing, Volume 73, Issue 12, December 2013, pp. 1649–1660.

[14] Jianliang Ma, Licheng Yu, John M. Ye, Tianzhou Chen, “MCMG Simulator: A Unified Simulation Framework for CPU and Graphic GPU,” Journal of Computer and System Sciences, Volume 81, Issue 1, February 2015, pp. 57–71.

[15] S. Aksoy and R. Haralick R, “Feature Normalization and Likelihood-based Similarity Measures for Image Retrieval, Pattern Recognition Letter, Special Issue on Image and Video Retrieval, 2000.

[16] Sergey Ioffe and Christian Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 32nd International Conference on Machine Learning, Lille, France, July 6-11, 2015.