ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Enabling Efficient Personally Identifiable Information Detection with Automatic Consent Discovery

Somchart Fugkeaw[1] and Pattavee Sanchol[2]

ABSTRACT

Personal data leakage prevention has now become a critical issue for implementing data management and sharing in many industries. Several data privacy regulations such as General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPPA), California Consumer Privacy Act (CCPA), and Thailand's Personal Data Protection Act (PDPA) have been issued to enforce organizations to collect, process, and transfer personally identifiable information (PII) securely. In this paper, we propose a design and development of PII RapidDiscover, an efficient Thai and English PII discovery system featured with automatic consent discovery. At the core of our proposed system, we introduce the PII scanning algorithm based on the Presidio library and a natural language processing (NLP) technique to improve the scan result of PII written in Thai and English. Finally, we conducted the experiments to demonstrate the efficiency of our proposed system.

Article information:

Keywords: GDPR, PDPA, DLP, PII Scanning, Consent Discovery, Natural Language Processing

Article history:
Received: April 4, 2023
Revised: May 10, 2023
Accepted: June 3, 2023
Published: June 10, 2023
(Online)

DOI: 10.37936/ecti-cit.2023172.252270

1. INTRODUCTION

Personally Identifiable Information (PII) is information that can be used to identify, contact, or locate a single person, or to identify an individual in a certain context. PII can be referred to direct identity information like name, citizen ID, biometric records, or indirect identity information such as contact address, car license plate, IP address, etc. In addition, PII can be defined as "personal information" such as health records, opinions, and financial health. Protecting the security and privacy of PII is one of the crucial organizational policies in satisfying compliance and avoiding breaches or loss. Current data privacy regulations or laws such as GDPR [1], California Consumer Privacy Act (CCPA) [2], Payment Card Industry Data Security Standard (PCI DSS) [3], Health Insurance Portability and Accountability Act (HIPAA) [4], Thailand's Personal Data Protection Act (PDPA) [5], and more have a focus on the protection of personal data from the unauthorized use, disclosure, collection, and processing. Basically, data controllers need to obtain consents from the data owners or data subjects before using or disclosing their personal information. To this end, the PII scanning and consent management system is essential for successful implementation to comply with any data privacy regulations. All regulations mentioned above focus on the privacy protection of PIIs found in any document sources. To effectively manage the privacy of PII, DLP software and other security mechanisms such as IDS, firewall, and encryption are employed by many organizations for high security and privacy protection of their internal and customer data.

Currently, there are PII scanning tools [6, 7] available in the market that are able to scan the PII instances from the target sources. They share a common feature in supporting manual or automatic scanning of the documents, files, web pages, and databases containing the PII instances, and generating the report. Most existing PII scanning tools are dedicated to scan and discover sensitive data such as personal information and payment data. The tools generally determine where personal and sensitive data are located and what PII instances are detected. This helps organizations gain insight into the visibility of personal data available in their control and ensure regulatory compliance.

Typically, techniques or libraries used for scanning are based on an exhaustive search through keywords, pattern-based scanning, regular expression matching, and natural language processing (NLP). Recently, a

[1,2] The authors are with Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani, Thailand, E-mail: somchart@siit.tu.ac.th and pattavee.san@gmail.com

new concept called human-in-the-loop NLP has been introduced by many works [8, 9, 10] to improve the accuracy of the results. However, existing PII scanning tools share the following common problems. First, existing tools have not been designed to fully support some different formats of PII instances used in different countries, such as citizen ID, telephone no., car license plate no., address details, etc. Second, the scanning results are usually turned out in the form of a set of PII instances and their properties. However, some sensitive information such as personal opinion, treatment records, personal preference, payment data, etc., are considered prohibitive issues in certain regulations [1, 2, 3, 4, 5], and they are overlooked by the existing systems. There are no tools providing both selective and comprehensive scanning results of the PII and their surrounding data. Third, most systems, especially the open-source PII scanning systems need to scan the whole content and convert them into text format before they are analyzed. This degrades the scanning performance when the system deals with a large volume of scanning tasks. Finally, existing systems have no capability to support the automatic checking of the consent status of the documents. Additional procedure in marking the status of documents and consent is done manually.

To this end, we are motivated to develop a more efficient PII scanning to complement these shortfalls in an integrated manner. In this paper, we proposed the *PII RapidDiscover* system to provide automatic and efficient Thai and English PII scanning and consent discovery. The system is based on the extension of the standard library called Presidio [27] and the NLP model through parallel programming. Major functions are to scan and recognize any Thai and English PII data contained in various sources such as PDF, Ms. Office, database, cookies, web pages, CSV files, and text files. Regarding the scanning performance, our scanning algorithm scans and interprets the data in a real-time fashion. At the core of the recognizer engine, the Presidio library is enhanced with our proposed PII recognition algorithm based on the Named Entity Recognition (NER) model and classification rules. The results of PII scanned are well classified to their respective type, which supports efficient security control over the documents. Then, the final results are systematically stored in the PII inventory. To discover the status of the consent of the files containing PII instances, we developed an automatic web service agent to check the data subject of each file with the consent management. This enables the integration of PII discovery and consent management to be done in an efficient and accurate manner.

As a result, our proposed system can resolve all the aforementioned problems. The system solution developed can be considered a profound solution to extend to cope with PII scanning and apply in other domains.

Contributions of our proposed system are summarized as follows.

1. We devised a PII recognition algorithm based on the integration of the Presidio library and NLP method that allows any public PII data set and customized or specific PII data to be trained in the system. This provides the flexibility to support any PII formats or languages to be used.
2. We introduced a classification rule set to classify the scan results of PII instances based on their sensitivity. This renders effective management and control of files containing PII.
3. Our system is efficiently implemented with on-the-fly scanning and recognition basis through parallel programming. This helps to improve the scanning performance.
4. We developed an automatic consent-checking method through a secure web service to check document files containing PII whether they have received consent from the data subject.
5. We conducted experiments to substantiate the accuracy and efficiency of our proposed model.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 presents our proposed system. Section 4 describes the experiments. Section 5 gives the concluding remark and identifies future works.

2. RELATED WORKS

Most research works [11, 12, 13, 14] related to PII discovery used string-matching techniques to discover PII and supported data breach detection. For example, AntMonitor [11] and Lumen [12], provided a blacklist of potential PII leaks represented in string. The system then worked on deep packet inspection (DPI) applied to each packet to match any PII strings in headers and/or payload. In the Recon system [13], classifiers were trained to detect PII within packets. The traffic was routed to analyzed at a centralized remote server. In [14], the authors proposed a PII detector for the virtual network (PIID VNF) to detect plaintext PII strings embedded in HTTP fields. The system investigated the packets containing PII, and then they were marked by the PIID. Depending on the desired system behavior, a PII-containing packet may be either taken into the analysis for further breach detection or instantly discarded by the PIID instance.

In [15], the authors proposed a method that can automatically discover various types of PII in the network system. They developed a list of constraints, or seed rules expressed in regular expressions. For PII data types that may not be as easily expressible as a regular expression, such as customer names, cities, and regions, seed rules were expressed as dictionaries containing lists of possible values. However, the seed rules generated did not cover all cases, formats, and languages.

In [6], the authors proposed a SecP2I approach entailing a privacy-preserving PII discovery platform in structured and semi-structured datasets. In their approach, PII is detected based on the knowledge-base, injection of regular expressions in SQL and NoSQL queries, and a reference base. However, the security protection based on SHA-1 is no longer secure.

In [16], J. Huang et al. presented a PII detection approach working on regular expressions to detect and remove PII from the natural language text in the dataset. Their proposed system is the novel one that focuses on detecting PII expressed in handwriting format. Technically, the system tester reads through the text of the entire dataset and manually marks up all the PII from it. The marked-up PII is used to test the implemented Python PII detection program. However, the search algorithm is exhaustive and may suffer from the performance problem if the volume of the data set is huge.

Recently, Alizadeh et al. [17] applied natural language processing and machine learning to detect PII data leakage. They used the public datasets, which are available contracts in PDF files that were converted to text files using the PyPDF2 Python library. Then, the PII can be extracted, and the necessary tokenization and learning model over PII can be done with the Natural Language Processing (NLP) and machine learning technique.

For the open-source PII scanner, CUSpider [18] is one of the common PII scanning tools developed by Columbia University. It is a forensic file-scanning program that can scan Windows desktops and laptops for PII. The scanning result is presented in a list of PII found in the target folders. It also provides a redaction option for several file types from within the application.

In [19], Silva et al. proposed and evaluated the use of Named Entity Recognition (NER) to identify, monitor and validate PII. In their method, the publicly available data sets are manually labeled, and they are trained with the machine learning models. Then, three NLP tools [20, 21, 22] are employed to evaluate the set of trained data. This work substantiated the performance of NLP tools in discovering the PII.

In [23], the authors develop a monitoring tool based on Python script to access websites where the PII data breach cases have been reported in the news periodically (i.e., once every 24 hours) via their RSS feeds. The PII web data are collected based on the keyword search throughout the target web pages.

In [7], Fugeaw et al. proposed AP2I as a PII scanning tool tailored for Thailand's PDPA compliance. The scanning process is based on the regular expression matching. The consent validation checking was also developed based on the consent flag done in the file name level. However, the system encounters the performance problem when it serves a large number of document files or other PII sources. Also, the detection of certain PII written in Thai is not recognized by the system. Hence, the flexibility and the extensibility to support other regulations are limited.

Mostly, data loss prevention (DLP) solutions [24, 25, 26] have been employed by enterprises to monitor and protect sensitive data by deploying software tools at the network levels or endpoint systems such as databases and file directories. However, DLP generally supports data monitoring, and it is not designed to discover the PII. The fundamental requirement is thus to establish foundation data and apply supporting tools to entail complete data privacy management tools.

3. OUR PROPOSED SYSTEM

Figure 1 illustrates our *PII RapidDiscover* system model. It comprises five main components: converter, analyzer, consent checker, PII inventory, and classification module. Our system is designed to be modular for ease of integration to any PII sources and existing consent management systems. The details of each system module are described as follows.

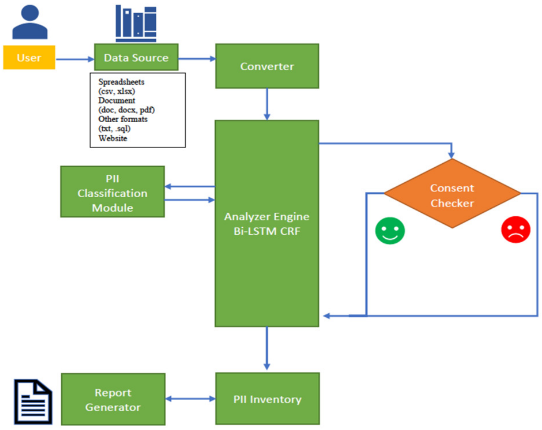


Fig.1: *PII RapidDiscover System Model.*

1) *Data sources* are the data files such as PDF, Ms. Office, Web pages, and database tables where the PII data reside.
2) *Converter* is responsible to convert any text data into CSV format.

To efficiently handle the different file types, we employed different handlers for each type of data. For the database, the converter uses sqlite3 to open the DB file and list all the tables. Then, the text in each table is scanned and dumped into the CSV file. The converter also collects metadata of data sources. For example, in the database, the table, and database name are inserted into each CSV record.

For web pages, we used bs4 to receive the URL as input and start scrapping the content of the webpage and the result is saved as a CSV file. Also, the converter inserts the webpage information into each CSV record.

For PDF and Ms. Office files, we used PyPDF2 and docx, respectively, to extract all the text in the files and store them in a CSV file. The converter inserts the line of text and page into the CSV file. All text data in each CSV file are sent to the Analyzer queue before text data are extracted and read by the Analyzer engine. The text data also provide information that can be referred to the source data. The converter limits the size of each CSV to no more than 1 MB.

For any text files, such as cookies, the converter can directly recognize and send them to the analyzer queue.

Figure 2 illustrates how the different file types are handled by the different conversion libraries.

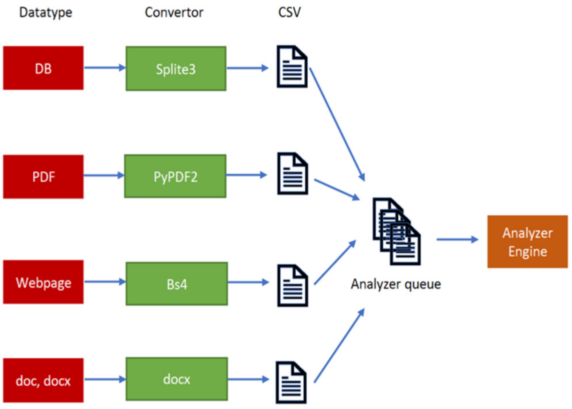
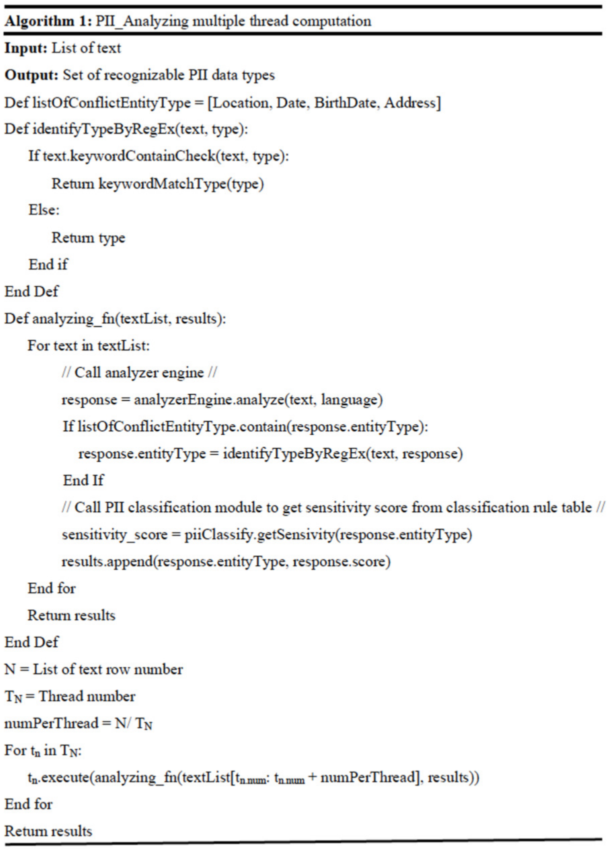


Fig.2: *File Conversion Methods.*

3) *Analyzer engine* takes CSV data, converted from all datatypes, from the analyzer queue. Then, it scans and recognizes the PII data. In the analyzer engine, we applied two methods, including (1) Named Entity Recognition (NER) which is a subtask of the information extraction in NLP, and (2) Regular Expression as a core for the detection of PII. For the NLP used in our system, we applied spaCy [22], which is an open-source Python software to support NLP model training. Also, we integrated the Presidio recognizer to improve the recognition rate of various PII formats. To detect certain PII types that are difficult to be recognized by the exact tag, such as an address, and descriptive sentences containing PIIs, we used regular expression (regex) to extract keywords and label them as "PIIs".

To support the scanning and recognition of many documents, we apply parallel programming by generating multiple threads to parallelize multiple transactions. For the large input size, the data will be split into the list of text data in the conversion step then each text will be handled by the thread spawned to recognize the PII. Here, we propose an algorithm to automate the scanning and recognition process in the Analyzer engine. The algorithmic process is presented below.



```
Algorithm 1: PII_Analyzing multiple thread computation
Input: List of text
Output: Set of recognizable PII data types
Def listOfConflictEntityType = [Location, Date, BirthDate, Address]
Def identifyTypeByRegEx(text, type):
    If text.keywordContainCheck(text, type):
        Return keywordMatchType(type)
    Else:
        Return type
    End if
End Def
Def analyzing_fn(textList, results):
    For text in textList:
        // Call analyzer engine //
        response = analyzerEngine.analyze(text, language)
        If listOfConflictEntityType.contain(response.entityType):
            response.entityType = identifyTypeByRegEx(text, response)
        End If
        // Call PII classification module to get sensitivity score from classification rule table //
        sensitivity_score = piiClassify.getSensivity(response.entityType)
        results.append(response.entityType, response.score)
    End for
    Return results
End Def
N = List of text row number
$T_N$ = Thread number
numPerThread = N/ $T_N$
For $t_n$ in $T_N$:
    $t_n$.execute(analyzing_fn(textList[$t_{n.num}$: $t_{n.num}$ + numPerThread], results))
End for
Return results
```

The algorithm starts by taking the text as input and transfers it into the analyzer engine. The engine generates threads to scan and recognize PII keywords contained in the text based on the RegEx and Bi-LSTM-CRF model. Then, the system classifies the PII data based on the classification rule. Then, it returns the results, which are PII instances with entity type and recognition score. As shown in the Algorithm 1, we introduced a list of conflict entity to retain possible tags that have similar patterns but different meaning. In our system, we applied RegEx to identify some entity types which can be ambiguous in giving the exact result. In essence, the algorithm checks all recognized PII tags whether there are some tags appearing in the list. If any tags are listed in the listofConflictEntity, the RegEx extraction process is executed to analyze the ambiguous tag with the keywords and exact patterns of the tag. This helps increase the accuracy rate of recognized set of PII. For example, "Bangkok" can be identified as a province (Location tag), however, it can be identified as the address. Here, the RegEx method was used to extract the location tag contained in the sentence and check with the keyword database to identify the type of PII. The RegEx function type in our system is described as follows.

(1) Extract sentence between two keywords:
$(? <=\text{Keyword1})(.*)(? = \text{Keyword2})$

(2) Extract sentence that starts with keywords: $\hat{}$Keyword.$*$\$

(3) Extract sentence that ends with keywords: . $*$. Keyword \$

To give more details about the core construct of the analyzer engine, the Bi-LSTM-CRF Model and classification rule are described in the next sections.

4) PII Inventory is a repository retaining all scanned results and it provides metadata of each scanned PII such as the location of the sources of PII discovered (URL or directory), the data owner who has the right to access that data sources, the data subject who owns the information, and the consent status that got from the data subject. The inventory provides insight into sensitive data and enables the PII traverse for the data sources and data owner.

5) Consent Checker is used to check whether document files containing PII instances have received consent. This function is required to ensure that all documents containing the PII have consent from the data subjects. Otherwise, it is failed to comply with the privacy regulations. Due to the compliance ground truth, this system model is crucial. We also proposed a dynamic checking algorithm that is applicable to the consent management system (CMS) where the list of document files having consent from the data subject is collected. In this paper, we assume that the consent management system is a separate system where all document files and their consent status are stored. Therefore, our system only checks the status by getting a pair of PII tags and consent status flags from the CMS. As shown in Figure 3, the system checks whether the data subjects have consented to their PII. The consent checker uses the data sources, and file name acquired from the scanning process to check the existence of that file in the consent management system. The consent checker uses a flag to mark the consent status of the files from the data subject by sending API to web service requests for consent checking automatically. We assume that if any files have been consented, they are available in the consent management system. On the other hand, if the consent checker cannot find the file containing PII inside the consent management system, there are no consents obtained from the data subject. Then, the system sends an alert to the data protection officer. If there is a file that matches the input data, the consent checker will mark that file with ‘yes’ flag; otherwise, the data will be marked ‘no’. Then, the results will be stored in the PII inventory. Figure 3 presents how the consent status checking is done in our scheme.

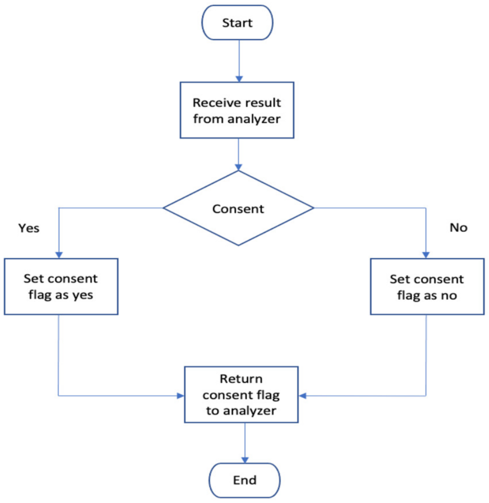
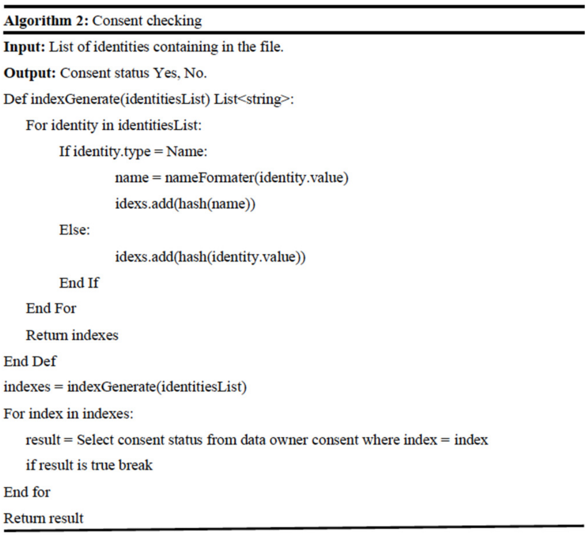


Fig.3: *Consent checking process and PII Inventory data flow.*

The pseudocode below shows the procedure of how the consent status of the documents are detected and the indices are created.



Algorithm 2: Consent checking

```
Input: List of identities containing in the file.
Output: Consent status Yes, No.
Def indexGenerate(identitiesList) List<string>:
    For identity in identitiesList:
        If identity.type = Name:
            name = nameFormater(identity.value)
            idexs.add(hash(name))
        Else:
            idexs.add(hash(identity.value))
        End If
    End For
    Return indexes
End Def
indexes = indexGenerate(identitiesList)
For index in indexes:
    result = Select consent status from data owner consent where index = index
    if result is true break
End for
Return result
```

For the dynamic consent checking algorithm, we used the identity information such as Name, Passport ID, etc. contained in the scanned file to search the consent status from the data owner’s consent database. In our model, we use the hash value of the identity data for indexing to improve the search performance and preserve the privacy of the identity information.

6) Report Generator generates reports such as consent tracking report, and PII detection summary report.

7) *PII Classification Module* is responsible to classify the PII scanned results. In our system, the threads accept recognized PII results and classify them into three categories: public data, private data, and sensitive data. Table 1 shows the classification of PII types.

Table 1: *Classification Rule of PII Instances.*

PII Types	Sensitivity
Restricted data – data that are highly sensitive, such as biometric data, financial, political opinion, and medical records	Highly Sensitive
Private data – data that are relevant to personal attributes, such as their date-of-birth, home address and phone number.	Sensitive
Public data – data that can be available in “public media”, such as telephone directories, business directories, social media applications, and websites.	Non-Sensitive

In fact, our classification scheme can classify any kind of PII data through the training phase of our NLP module. Since most common PII data are deterministic, the abundant class is effectively handled. For the rare class or the class of underrepresented data, we retrain the model with several cases of the rare class together with the abundant class.

4. Bi-LSTM-CRF MODEL

Technically, we construct the NLP model based on the Bi-LSTM-CRF Model [28, 29] that integrates Long-Short term memory networks (LSTM) [30] and Conditional Random Field (CRF) [31]. Then, we trained the model with the sample data set. Importantly, the Bi-directional LSTM that leverages the LSTM network with an additional layer passes the data from the backward direction. In our training model, we applied Bi-LSTM as a core of our training model by using the preceding words and the successive words to effectively discover PII instances. In addition, we used the conditional random field (CRF) based on the name entity recognition (NER), which is applied to predict the sequence of labels from the dependencies of the word analysis. It also characterizes interpretable sequences of tags that imposes several hard constraints. For example, the I-PER tag (a subset of the personal name word) is not followed by the I-LOC tag (a subset of the location name word). Therefore, instead of decoding the label independently, a conditional random field is used to model the label sequence.

Let $x = \{x_1, x_2, \ldots, x_n\}$ and it represents an input sequence where x_i is the input vector of the i$^{\text{th}}$ word and $y = \{y_1, y_2, \ldots, y_n\}$ that represents tag prediction sequences from sequence x. P is the matrix of the output score from the Bi-LSTM network. P has the size of $n \times k$ where k is the number of unique tags. $P_{i,j}$ is the score of the j^{th} tag of the i^{th} word in a sentence. We can define the score as:

$$S(x|y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=0}^{n} P_{i, y_i}$$

where A is a matrix of transition scores and $A_{i,j}$ describes the score of transition from tag i to tag j. Here, y_0 and y_n indicate the start and end tags from the sentence. A softmax of all possible tags represents the probability for sequence y as:

$$P(y|x) = \frac{\exp(S(x,y))}{\sum_{y' \in Y_x} \exp(S(x,y'))}$$

In the training phase, we used the maximum conditional possibility estimation to maximize the log probability of the correct tag sequence by:

$$\log(P(y|x)) = S(x,y) - \log\left(\sum_{y' \in Y_x} \exp(S(x,y'))\right)$$

Decoding is to predict the label $y*$ with the highest score given by:

$$y^* = \underset{y' \in Y_x}{\operatorname{argmax}}\, S(x,y')$$

In our system, the input of the main Bi-LSTM-CRF model training data is a list of words tokenized from example PII information text, then the outputs of the tokenization process are fed into Bi-LSTM network. Finally, the output vectors of Bi-LSTM are fed into the CRF layer to decode the best tag sequence. Figure 4 illustrates the Bi-LSTM model used in our Analyzer engine.

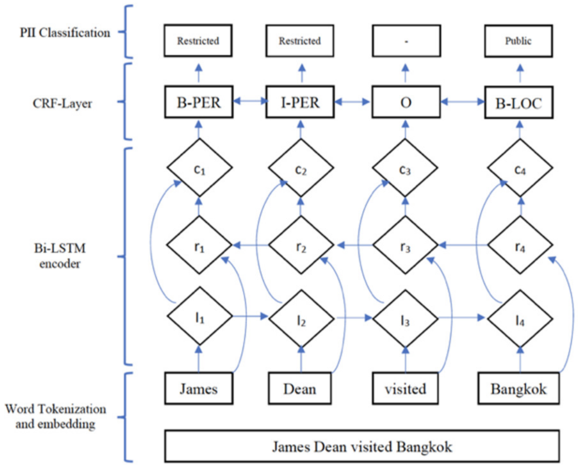


Fig.4: *Bi-LSTM CRF Architecture.*

In our model, the engine allows public and customized data set to be trained in our system. Each PII entity type is specifically trained and recognized by the system. Accordingly, our analyzer engine can support any PII format, language, and type. The engine generates the score of the scan results. For those instances that have scores below 80%, they will be sent to be trained with the user-defined training data. In this process, the input of the latest scan result is then used to generate the additional training data by assigning the corresponding PII tag. Finally, the Bi-LSTM CRF model is updated after the training is done.

After we trained the Bi-LSTM CRF model for the NLP module and integrated it with the PII engine, we can use the PII engine to scan the PII data. The result is given as an array list of the category of the PII, the location, and the accuracy score.

Specifically, we used Adam with batch size = 32 as our optimization algorithm for the NLP model. The initial learning rate was 0.001, the state unit of Bi-LSTM was 512 units, and the dropout was 0.6. In the experiment, we trained 100 epochs to select the best result to update the model. For the hyperparameter, we optimized the LSTM units to obtain the result when the unit reached 512.

5. EXPERIMENT

This section explains the experiment setting and the comparative result of documents containing PII data written in Thai between our proposed system and two works. The evaluation was done in two parts: the accuracy of the NLP model and scanning performance.

5.1 NLP Model training

Firstly, we trained the NLP model to recognize and label PII data. For this purpose, we used the NLP tool called spaCy [22] to train our model.

For the PII dataset used in our system, we extended the standard Thai NER data set, which is an open-source dataset from PyThaiNLP project [33]. We included more entity classes from 3 classes, including person, organization, and location. The total entities classes were 13. However, the dataset from the PyThaiNLP project has been designed to support standard NER tag only. Therefore, we collected only PII dependent tags originally obtained from [33], which provides 6,000 sentences. However, some tags such as birthdate, passport no., and ThaiCitizen ID are not covered in PyThaiNLP's dataset. Consequently, additional 100 sentences were generated to be used in our system. Table 2 shows examples of PII tags related to identity information. It presents the type of PII and a number of words and tokens. Basically, a word represents the exact PII keyword, while the token refers to possible word fragments of a sentence that may or may not correlate to the PII keyword.

Table 2: *PII label description.*

Type	Word	Token
Name	3300	15313
Phone	108	391
Email	30	112
ThaiCitizen ID	20	20
Passport NO.	20	20
Address	4488	9239
Birth Date	30	30

We conducted the tests by splitting the data set with 80:20 ratio (training:testing), to compare the Bi-LSTM-CRF models and spaCy tool. Here, we compared our work with [29] using Bi-LSTM-CRF and CRF model [33] using NER and POS [33]. Table 3 shows the comparison of the F1 score of each model. F1 score is a weighted mean of the precision, percentage of relevant results calculated by $\frac{Truepositive}{Truepositive+Falsepositive}$, while the recall, characterized as the percentage of relevant results, is calculated by $\frac{Truepositive}{Truepositive+Falsenegative}$. The following equation denotes how the F1 score is computed.

$$F = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Based on the result shown in Table 3, the F1 scores achieved by the original Bi- LSTM-CRF [29] are almost similar to our scheme using Bi-LSTM CRF over spaCy tool, while the scheme [33] yielded fewer scores than [29] and ours. This confirms that Bi-LSTM-CRF is applicable for extending to recognize Thai words as it supports English words.

Table 3: *PII label description.*

Type	[29]	[33]	Our Model
Name	96.85	89.13	96.68
Phone	98.02	86.90	99.01
Email	100	100	100
Thai Citizen ID	98.28	85.60	99.65
Passport NO.	100	100	100
Address	79.36	71.52	90.73
Birth Date	85.56	79.65	95.66

Table 4, shows the results of the average value of precision, recall, and F1score computed from all sample PII instances of our model, [28], and [33].

Table 4: *Precision, Recall, and F1 score Comparison.*

Model	Precision	Recall	F1
[29]	94.10	93.92	94.01
[33]	87.71	87.35	87.53
Our Model	97.63	97.18	97.39

Based on the results obtained, our model delivers best recognition rate as it gave the highest score for all precision, recall, and F1 than [29] and [33].

5.2 PII Scanning Performance

We conducted the experiments on Intel Core I5-7200U using 8 GB of RAM running on Ubuntu server 20.04, Ubuntu Server 20.04 using Python language to create a scanning service, consent checker, and the GUI interface which serves on the web server. The PII inventory and consent management system are developed in MySQL. The scanning service uses a pattern-based recognizer with the analyzer engine to discover PII entities in the text. The Presidio library [27] was used as a core in the Analyzer engine, where the NLP is additionally embedded.

To evaluate the scanning performance, we compared the scanning performance of our proposed scheme with [29] and [33].The processing time was measured based on the varied number of text lines in PDF files.

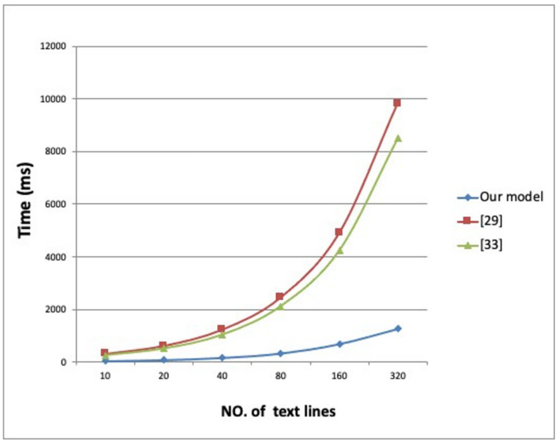


Fig.5: *Performance evaluation.*

As shown in Figure 5, our proposed scheme outperforms [29] and [33]. This is because our scheme leverages multi-thread processing to boost the speed of scanning a large volume of documents while the employment of the integration of Presidio and our proposed recognizer using NLP and RegEx method helps improve the rapid PII recognition. This advantage is even obvious when the number of documents was increased. Our scheme yielded a better performance rate over the related works.

6. CONCLUSION AND FUTURE WORK

We have proposed the PII scanning and discovery system that automatically and adaptively scans and discovers PII in any endpoints systems. Our system can automatically discover the consent status of data sources to provide insight into PII management. Furthermore, we enhance the accuracy of the scanner by customizing the recognizer to work with Thai PII data sources. Since obtaining consent from the data subjects before their PII are used is mandated by most privacy laws, including Thailand's PDPA, the effective collection of PII and the capability to check the consent for all documents are important. These capabilities are supported by our system. Finally, we conducted the experiment to show the efficiency of our system. For future work, the tool will be implemented to enable the data owner to manage the consent of the existing files efficiently. To render more functionality in serving PII scanning on image documents, we will consider NLP and optical character recognition (OCR) techniques for the PII recognition. In addition, the cryptographic-based access control mechanism will be implemented on the PII inventory to guarantee the privacy of the PII.

References

[1] `https://gdpr-info.eu`

[2] `https://oag.ca.gov/privacy/ccpa`

[3] `https://www.pcisecuritystandards.org`

[4] `https://www.cdc.gov/phlp/publications/topic/hipaa.html`

[5] `http://www.ratchakitcha.soc.go.th/DATA/PDF/2562/A/069/T_0052.PDF`

[6] A. Mrabet, M. Bentousi and P. Darmon, "SecP2I A Secure Multi-party Discovery of Personally Identifiable Information (PII) in Structured and Semi-structured Datasets," *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, pp. 5028-5033, 2019.

[7] S. Fugkeaw, A. Chaturasrivilai, P. Tasungnoen and W. Techaudomthaworn, "AP2I: Adaptive PII Scanning and Consent Discovery System," *2021 13th International Conference on Knowledge and Smart Technology (KST)*, Bangsaen, Chonburi, Thailand, pp. 231-236, 2021.

[8] I. Arous, L. Dolamic, J. Yang, A. Bhardwaj, G. Cuccu, P. Cudré-Mauroux, "Marta: Leveraging human rationales for explainable text classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, pp. 5868–5876, 2021.

[9] Z. Liu, Y. Guo and J. Mahmud, "When and why does a model fail? A human-in-the-loop error detection framework for sentiment analysis," *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pp. 170-177, 2021.

[10] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma and L. He, "A survey of human-in-the-loop for machine learning," *Future Generation Computer Systems* , vol. 135, pp. 364-381, 2022.

[11] A. Shuba, A. Le, E. Alimpertis, M. Gjoka, and A. Markopoulou, "Antmonitor: System and applications," *arXiv:1611.04268*, 2016.

[12] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman and V. Paxson, "Haystack: A multi-purpose mobile vantage point in user space," *arXiv:1510.01419v3*, Oct. 2016.

[13] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes. Recon, "Revealing and controlling pii leaks in mobile network traffic," in *Proceeding of the 13th Annual Int. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, vol. 16, New York, NY, USA, 2016.

[14] S. J. Y. Go, R. Guinto, C. A. M. Festin, I. Austria, R. Ocampo and W. M. Tan, "An SDN/NFV-Enabled Architecture for Detecting Personally Identifiable Information Leaks on Network Traffic," *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, Zagreb, Croatia, pp. 306-311, 2019.

[15] Y. Liu, H. H. Song, I. Bermudez, A. Mislove, M. Baldi and A. Tongaonkar, "Identifying personal information in internet traffic," in *Proceeding of the 2015 ACM on Conference on Online Social Networks, COSN '15*, New York, USA, pp. 59–70, ACM, 2015.

[16] J. Huang, B. Klee, D. Schuckers, D. Hou and S. Schuckers, "Removing Personally Identifiable Information from Shared Dataset for Keystroke Authentication Research," *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, Hyderabad, India, pp. 1-7, 2019.

[17] F. Alizadeh, T. Jakobi, A. Boden, G. Stevens and J. Boldt, "GDPR Reality Check - Claiming and Investigating Personally Identifiable Data from Companies," *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Genoa, Italy, pp. 120-129, 2020.

[18] "CUSpider," Accessed on: Sep. 23, 2020. [Online]. Available: `https://cuit.columbia.edu/content/cuspider-pii-scanning-application`

[19] P. Silva, C. Gonçalves, C. Godinho, N. Antunes and M. Curado, "Using NLP and Machine Learning to Detect Data Privacy Violations," *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, pp. 972-977, 2020.

[20] S. Bird, E. Klein and E. Loper, *Natural Language Processing with Python*, Boston, USA: O'Reilly Media, 2009.

[21] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, Jun. 2014.

[22] ExplosionAI. (2019) spacy - industrial-strength natural language processing. [Online]. Available: `https://spacy.io`

[23] Y. Liu et al., "Identifying, Collecting, and Monitoring Personally Identifiable Information: From the Dark Web to the Surface Web," *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Arlington, VA, USA, pp. 1-6, 2020.

[24] E. Costante, D. Fauri, S. Etalle, J. den Hartog and N. Zannone, "A Hybrid Framework for Data Loss Prevention and Detection," *2016 IEEE Security and Privacy Workshops (SPW)*, San Jose, CA, USA, pp. 324-333, 2016.

[25] A. Guha, D. Samanta, A. Banerjee and D. Agarwal, "A Deep Learning Model for Information Loss Prevention From Multi-Page Digital Documents," in *IEEE Access*, vol. 9, pp. 80451-80465, 2021.

[26] S. Fugkeaw, K. Worapaluk, A. Tuekla and S. Namkeatsakul, "Design and Development of A Dynamic and Efficient PII Data Loss Prevention System," in *Proc. of the 17th International Conference on Computing and Information Technology*, Springer, vol. 251, pp. 23-33, Bangkok, Thailand, 2021.

[27] `https://github.com/microsoft/presidio`

[28] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "Neural Architectures for Named Entity Recognition," *Association for Computational Linguistics: Human Language Technologies*, 2016.

[29] S. Thattinaphanich and S. Prom-on, "Thai Named Entity Recognition Using Bi-LSTM-CRF with Word and Character Representation," *2019 4th International Conference on Information Technology (InCIT)*, Bangkok, Thailand, pp. 149-154, 2019.

[30] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] J. D. Lafferty, A. McCallum and F. C. N. Pereira, "Conditional random fields:Probabilistic models for segmenting and labeling sequence data," *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282-289, 2001.

[32] N. Tirasaroj and W. Aroonmanakun, "Thai named entity recognition based on conditional random fields," *2009 Eighth International Symposium on Natural Language Processing*, Bangkok, Thailand, pp. 216-220, 2009.

[33] W. Phatthiyaphaibun, "Thai Named Entity Recognitions for PyThaiNLP," [Online]. Available: `https://github.com/wannaphongcom/thai-ner.`

Somchart Fugkeaw (Member, IEEE) received the bachelor's degree in management information systems from Thammasat University, Bangkok, Thailand, the Master's degree in computer science from Mahidol University, Thailand, and the Ph.D. degree in electrical engineering and information systems from The University of Tokyo, Japan, in 2017. He is currently an Assistant Professor with the Sirindhorn International Institute of Technology, Thammasat University. His research interests include information security, access control, cloud computing security, blockchain, big data analysis, and high performance computing. He has served as a reviewer for several international journals, such as IEEE ACCESS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON BIG DATA, the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, COMPUTER & SECURITY, the IEEE SYSTEM JOURNAL, IEEE Internet of Things Journal, and ACM Transactions on Multimedia Computing Communications and Applications.

Pattavee Sanchol received a Bachelor degree in Computer Engineering from Kasetsart University, Thailand, and Master degree in Engineering and Technology from Sirindhorn International Institute of Technology, Thammasat University, Thailand in 2022. His research interests are cyber security, access control, cloud computing security, programming languages, and networking.