Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

TC 11 Briefing Papers

Exsense: Extract sensitive information from unstructured data

Yongyan Guo, Jiayong Liu, Wenwu Tang, Cheng Huang*

College of Cybersecurity, Sichuan University, Chengdu 610065, China

ARTICLE INFO

Article history:
Received 9 June 2020
Revised 13 November 2020
Accepted 17 December 2020
Available online 31 December 2020

Keywords:
Sensitive information
Unstructured data
Sequence labeling
BERT
Attention mechanism

ABSTRACT

Large-scale sensitive information leakage incidents are frequently reported in recent years. Once sensitive information is leaked, it may lead to serious effects. In this context, sensitive information leakage has long been a question of great interest in the field of cybersecurity. However, most sensitive information resides in unstructured data. Therefore, how to extract sensitive information from voluminous unstructured data has become one of the greatest challenges. To address the above challenges, we propose a method named ExSense for extracting sensitive information from unstructured data, which utilizes the content-based and context-based extract mechanism. On the one hand, the method uses regular matching to extract sensitive information with predictable patterns. On the other hand, we build a model named BERT-BiLSTM-Attention for extracting sensitive information with natural language processing. This model uses the latest BERT algorithm to accomplish word embedding and extracts sensitive information by using BiLSTM and attention mechanism, with an F1 score of 99.15%. Experimental results on real-world datasets show that ExSense has a higher detection rate than using individual methods (i.e., content analysis and context analysis). In addition, we analyze about a million texts on Pastebin, and the results prove that ExSense can extract sensitive information from unstructured data effectively.

1. Introduction

With the rapid development of the Internet, a large amount of sensitive information is stored and transmitted on the Internet. Large-scale sensitive information leakage incidents are frequently reported in recent years. In March 2018, the *New York Times* reported that Facebook's 50 million user information was leaked by a company named Cambridge Analytica Cadwalladr and Graham-Harrison (2018). According to IBM's 2019 Cost of a Data Breach Report IBM Security (2019), the average cost of a data breach in 2019 is $3.92 million, a 12% increase from 2014, and the average size of a data breach is 25,575 records. Also, once sensitive information is leaked it can lead to significant contractual or legal liabilities; serious damage to personal image and reputation; or legal, financial, or business losses Ohm (2014). In this context, the issue of sensitive information leakage has received considerable critical attention.

Sensitive information leakage can be caused by both internal and external factors. The 2019 Data Breach Investigations Report Verizon (2019) released by Verizon shows that

* Corresponding author.
E-mail address: opcodesec@gmail.com (C. Huang).

https://doi.org/10.1016/j.cose.2020.102156
0167-4048/© 2020 Elsevier Ltd. All rights reserved.

data breaches caused by external factors include hacking and social attacks. Also, internal information breaches cannot be ignored, such as unintentional leakage caused by weak security awareness, internal data misuse by authorized users, and corporate espionage. Many security measures including firewalls, access control, IPS/IDS have been considered for data leakage caused by external factors. However, there is no effective way to deal with data leakage caused by internal factors. This is because sensitive information often resides in common unstructured data (e.g., email messages, blog posts, news, configuration files, etc), making it difficult for people to realize the occurrence of data leakage.

Currently, more than 80% of the data on the Internet is unstructured data Allahyari et al. (2017). Unstructured data usually refers to information that does not reside in a relational database. In other words, the data structure of unstructured data is irregular or incomplete and there is no predefined data model. In particular, it should be noted that although some documents like CSV, JSON, XML have some organizational properties, they usually do not have a clear predefined data model. Compared to structured data, these data still difficult to retrieve, analyze and store. Unstructured data is easily processed by humans but is very hard for machines to understand Gupta and Gupta (2019). It is thus beneficial to devise a means to process unstructured data, which helps us automatically detect sensitive information from it and prevent data leakage.

In order to protect sensitive information, many researchers focus on data leakage prevention (DLP) Hart et al. (2011); Meli et al. (2019); Shapira et al. (2013); Shu et al. (2015b). The existing methods Lin et al. (2020); Noor et al. (2019); Shvartzshnaider et al. (2019); Trabelsi (2019) can be classified into two categories: content-based analysis and context-based analysis. Content-based methods inspect data content based on features of sensitive information itself, such as regular expressions and data fingerprints. Content-based methods have high detection accuracy for sensitive information with predictable patterns (e.g., IP, email, API KEY). Context-based methods detect sensitive information based on contextual features around the monitored data. For sensitive information without predictable patterns (e.g., Login Password Combo), the context-based approach is more effective. Therefore, in order to extract sensitive information more comprehensively and accurately, appropriate methods should be adopted for different sensitive information.

Deep learning methods have achieved tremendous success in the field of computer vision and pattern recognition. Neural networks based on dense vector representation have achieved great results in many NLP tasks. Compared with traditional machine learning, deep learning makes multi-level automatic feature representation learning possible. With this in mind, deep learning methods used in the field of data leakage protection has become the trend.

In this work, we propose a method to extract sensitive information from unstructured data, utilizing the content-based and context-based extract mechanism. On the one hand, the method uses regular matching to extract sensitive information with predictable patterns. On the other hand, we build a model named BERT-BiLSTM-Attention to label sensitive information entities in the text based on contextual features. To ensure its accuracy, we compare it to several popular baseline methods and achieve significant improvements. Experimental results on real-world datasets show that the hybridization of content analysis and context analysis tends to produce better performance when compared to the individual methods. In general, the specific contributions of our research are the following:

- The paper proposes a framework named ExSense to extract sensitive information. ExSense contains two main modules: regular expressions are used to extract content-based sensitive information with predictable patterns; automatically machine learning extractor is used to extract context-based sensitive information with natural language processing technologies.
- In the context-based analysis, the paper presents the sensitive information extraction problem as a sequence labeling problem and builds a BERT-BiLSTM-Attention model. This model utilizes the BiLSTM neural network and the attention mechanism. Experimental results show that the performance of this model is better than other baselines, with an F1 score of 99.15%.
- The paper analyzes about 1 million texts on Pastebin and displays different types of sensitive information samples by using ExSense, which proves that the effectually and accurately of our framework.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 presents a detailed description of the sensitive information extraction method in this paper. Section 4 presents the experiments and analysis related to this work. Section 5 summarizes conclusion and proposes future works.

2. Related work

A prerequisite for sensitive information protection is extracting sensitive information, so in this section, we first review the existing sensitive information detection methods. Secondly, we introduce technologies related to the method proposed in this paper, which includes the sequence labeling model and attention mechanism. At the end of this section, we summarize the existing methods including contentbased analysis and contextbased analysis.

2.1. *Sensitive information detection methods*

In recent years, two categories of approaches have been considered for sensitive information detection: content-based analysis and context-based analysis.

The content-based analysis scans the data in storage and transmission, which is generally unstructured data. Common techniques are introduced below. Regular matching is used to identify sensitive information that follows predictable rules. Meli et al. (2019) detected API KEY leaks in Github. They used regular matching and search APIs to obtain candidate secrets, and then used entropy to check the randomness of information. Data fingerprinting is to compare the digests or hash values of the monitored data to be detected with fingerprints of known sensitive information. Shapira et al. (2013) proposed a fingerprinting method that extracts fingerprints from the core confidential content while ignoring non-relevant parts of a document. This method improves the robustness of the rephrasing of confidential content. Shu et al. (2015b) presented a sequence alignment technique to detect complex data leakage patterns. This algorithm is designed to detect long and inaccurate sensitive data patterns. Artificial intelligence technologies have been widely applied, many studies have introduced machine learning and natural language processing technologies to detect sensitive information leakage Alneyadi et al. (2015); Hart et al. (2011); Lin et al. (2020). Lin et al. (2020) designed a machine learning model, which can filter out confidential documents, and prompt alert to the user.

The context-based analysis is not to identify the presence of sensitive information, but to perform contextual analysis of the monitored data. Shvartzshnaider et al. (2019) proposed a data leakage detection method based on contextual integrity. Mathew et al. (2010) suggested modeling the behavior of normal users and alerting users when they deviated from normal behavior to avoid leakage of data. Shu et al. (2015a) proposed to detect anomalous access patterns in relational databases with a finer granularity based on mining database trace stored in log files. Their method is able to detect role intruders in database systems. Noor et al. (2019) proposed a machine learning framework for investigating data breaches based on semantic analysis of adversarys attack patterns in threat intelligence repositories. Katz et al. (2014) proposed a context-based approach that leverages the context of the keyword and statistical methods.

2.2. *Sequence labeling model for information extraction*

Sequence labeling models are often used for information extraction, such as the Named Entity Recognition. At present, most mainstream DLP systems detect data leakage at the document level. The difference is that for a given text sequence, the sequence labeling model labels each element in the sequence, which can extract sensitive information entities at the token level.

Models based on traditional machine learning algorithms mainly include Hidden Markov (HMM) and Random Condition Field (CRF). Passos et al. (2014) applied CRF to named entity recognition tasks and achieved effective results on the CoNLL2003 dataset. Nguyen and Guo (2007) compared the performance of algorithms including HMM, CRF, and structured SVM in part-of-speech and handwritten digit recognition tasks, and found that the structured SVM performed best. However, models based on traditional machine learning algorithms rely on artificially constructed features and domain-specific knowledge, making them difficult to apply to new domains.

With the development of deep learning, many sequence labeling models based on deep learning have appeared. Yubo et al. (2015) applied a convolutional neural network to the event extraction task. Ling et al. (2015) used BiLSTM for part-of-speech, which has obvious advantages over traditional algorithms. Huang et al. (2015) combined BiLSTM with the traditional machine learning algorithm CRF and achieved outstanding results in multiple natural language processing tasks. Deep learning-based models do not require artificially constructed features and are highly portable. Usually, deep learning models could perform better than traditional machine learning models in many natural language processing tasks.

2.3. *Attention mechanism for improving detection accurate*

The attention mechanism is initially applied in the field of image processing. In 2014, Mnih et al. (2014) combined RNN and attention mechanism for image classification. Bahdanau et al. (2014) first applied the attention mechanism in the field of natural language processing. In 2017, Vaswani et al. (2017) published a paper ”Attention is all you need”, and the Transformer structure they proposed was composed of the attention mechanism. Then in 2018, Google Devlin et al. (2018) proposed a pre-trained model BERT with a core structure of Transformers. This model achieved the best results in 11 natural language processing tasks. By utilizing an attention mechanism, a model can focus on information that is more critical to the current task. Therefore, the attention mechanism can optimize neural network models and improve the accuracy.

In this work, we summarize contentbased and contextbased extraction techniques. Content-based methods are usually easy to implement, but these methods also have some drawbacks. When sensitive information is altered or modified, the performance of methods such as regular matching and data fingerprinting will be greatly reduced. Context-based methods are usually implemented using machine learning and have considerable accuracy and scalability when detecting sensitive information leaks. In the process of actual detection, to collect contextual information such as user behavior is not easy and will take some time. It is worth noting that the textual context around sensitive information does not have this disadvantage. Therefore, the textual context is a suitable feature for detecting sensitive information. Different from these above works, we combine the content-based and contextbased analysis methods, that is, to adopt appropriate methods for different sensitive information.

3. Methodology

For a given unstructured data including various types of unstructured text, our goal is to extract sensitive information from it. First of all, we need to figure out what sensitive information is and the types of sensitive information. Ohm summarized the definition of sensitive information by surveying dozens of different laws and regulations. Sensitive information describes the information that can be used to enable privacy or security harm when placed in the wrong hands Ohm (2014). In this paper, we summarize the most common sensitive information into four types, including personal information, network identity information, secret and credential information, and financial information. Sensitive information examples of each type are as follows:

- **Personal information**: Name, SSN, Date of Birth, Nationality, Address, Phone number, Occupation, Health, Education.
- **Network identity information**: IP address, MAC address, Email, Social Media Account, System Account, Internet browsing history, Chat history
- **Secret and Credential information**: Login Password Combo, Processed Password (Salt, Encryption), Encryption Negotiation, Security Question, API key/token, Private key, Digital Certificate.
- **Financial information**: Consuming records, Bank Account information (Account, Bank Name, Bank Number), Credit Card information (Card number, CVV, Expiration), Digital Currency (Bitcoin, etc).

Following that, we describe our approach for extracting sensitive information from unstructured data. We briefly outline the overall strategy here before discussing details in the following subsections. As mentioned in Section 1, our proposed method for extracting sensitive information utilizes content-based and context-based extraction mechanism.

The architecture of our proposed method ExSense is shown in Fig. 1. Unstructured data comes from text documents in various formats on the internet. In data preprocessing, we utilize parsing tools to extract text content from rich text and then use common processing methods (i.e., text cleaning, text segmentation and text replacement) to get text sequences.

The sensitive information extraction method consists of two parts. Regular expressions extract content-based sensitive information with predictable patterns, such as IP address, email, API key, private key, and certificate. We develop regular expressions that can extract this sensitive information accurately and efficiently. Besides, in order to extract context-based sensitive information, we build a BERT-BiLSTM-Attention model to identify sensitive information entities in preprocessed sequences. This model uses BERT to accomplish word embedding and extracts sensitive information by using BiLSTM with an attention mechanism.

3.1. *Data preprocess*

The first step in preprocessing is to extract text from unstructured documents. Most documents on the Internet are rich text documents. Compared with plain text documents, rich text documents contain format information (e.g., font color, size, etc), and use tags to represent the above information. Common rich text document formats include HTML, XML, pdf, doc, pst, rtf. Except for plain text documents, which can be read directly, the text content in rich text documents cannot be directly obtained, so we use several parsing tools to extract text content in rich text documents. Parsing tools include HTMLParser, PDFLib, python-docx, libpst, etc.

In the training process, there is some information that is useless to train the model, and we need to preprocess it. First of all, remove all non-ASCII characters and remove whitespace at the beginning and end of each line. Secondly, convert all uppercase letters to corresponding lowercase letters. Thirdly, for English text, use NLTK's lemmatization to reduce the inflectional forms from each word to a common base or root. Next, in order to process text into sequences, we need to implement text segmentation. For sentence segmentation, each

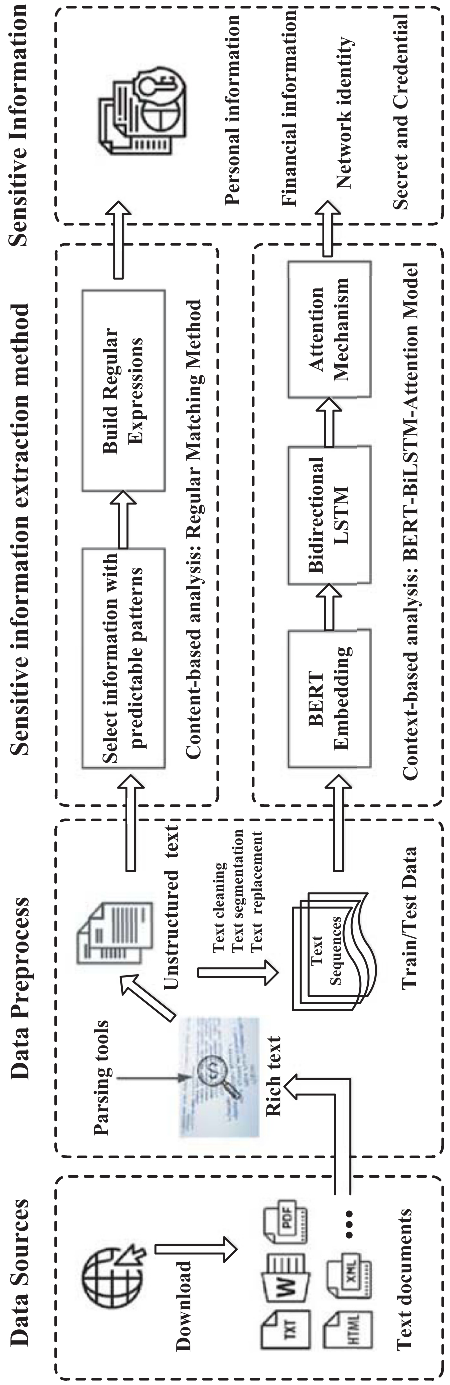


Fig. 1 – ExSense Architecture.

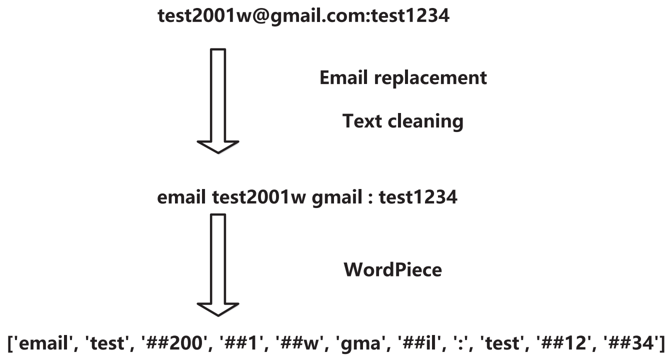

test2001w@gmail.com:test1234
Email replacement
Text cleaning
email test2001w gmail : test1234
WordPiece
['email', 'test', '##200', '##1', '##w', 'gma', '##il', ':', 'test', '##12', '##34']

Fig. 2 – Data preprocessing example.

Table 1 – Examples of email and URL related to sensitive information in text.

Example	Content
Example1	hacker123@gmail.com: 123456
Example2	facebook: https://www.facebook.com/abcdefg.hijklm.12

Table 2 – Email and URL replacement examples.

Raw text	Replacement
hacker123@gmail.com	email hacker123 gmail
https://www.facebook.com/abcdefg.hijklm.12	http facebook abcdefghijklm12

line of text is treated as a sentence. For word segmentation, we utilize WordPiece for tokenization.

In particular, since the formats of email and URL are special, these two types of information need to be further processed. We found that email is used as usernames for various accounts. In addition, email and password are often leaked at the same time, such as Example 1 in Table 1. URLs are related to certain network identity information, such as Example 2 in Table 1, which is the Facebook homepage of a user.

Both types of information need to be replaced with a natural language-like form. We replaced email with ”email username domain” and URL with ”http domain rest” (”rest” means all letters and numbers after the domain). Examples of replacements are shown in Table 2.

We demonstrate data preprocessing through an example. The raw text ”test2001w@gmail.com:test1234” is a sentence containing an email and password. The preprocessing of this sentence is shown in Fig. 2. After the above preprocessing, the sentence will be processed into a text sequence.

3.2. *Regular matching method*

In this section, we develop regular expressions to extract information with predictable patterns. It is noteworthy that some information seemingly has patterns, such as an SSN consisting of 9 digits. Such information cannot be extracted through regular expressions because the patterns are not unique. The 9-digit information is not necessarily the SSN, and it may also be a phone number, so simply extracting it through regular expressions will yield a high false positive ratio. According to the sensitive information types mentioned in Section 3, there are 23 types of information that can be extracted using regular expressions. The detailed categories and regular expressions are summarized in Table A.1 in the Appendix. While these types are not exhaustive, they represent many of the most popular sensitive information.

Table 3 – Tags of Sensitive Information Entities.

Type	Tag
Personal information	B-I and I-I
Network identity information	B-N and I-N
Secret and Credential information	B-S and I-S
Financial information	B-F and I-F
Non-sensitive information	O

3.3. *BERT-BiLSTM-Attention Model*

In many cases, sensitive information exists in natural language and can be extracted by natural language processing. As described in Section 2.2, in order to precisely locate sensitive information in unstructured data, we can use a sequence labeling model to extract sensitive information entities at the token level. The sequence labeling model can label each element of a sequence. Usually, a sequence is a sentence, and an element is a word in the sentence. In the field of natural language processing, many tasks apply sequence labeling models. Taking Named Entity Recognition as an example, the objective is to tag entities like names, locations, and organizations in the given input sequence. This work can also be regarded as a special case of Named Entity Recognition, identifying the sensitive information entities from unstructured data.

The sequence labeling strategy chosen in this paper is ”BIO”. In ”BIO” labeling strategy, ”B” represents the beginning of an entity, ”I” represents the inside of an entity, ”O” represents the outside of an entity. According to the types of sensitive information, there are 9 types of tags for sequence elements, as shown in Table 3.

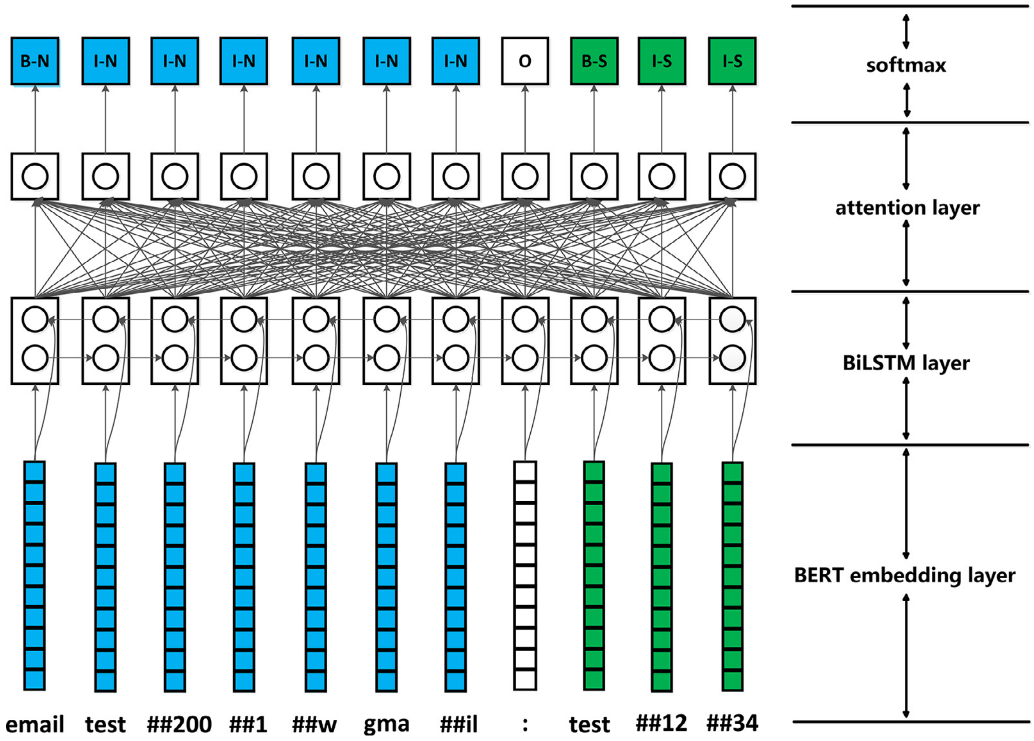


Fig. 3 – BERT-BiLSTM-Attention Model.

Our proposed sequence labeling model is named BERT-BiLSTM-Attention Model. The overall architecture of this model is shown in Fig. 3. The first layer is the word embedding layer that generates an embedding vector e_t for each token x_t in the input sequence X. The second layer is the BiLSTM layer, the embedding vector is used as an input to the BiLSTM layer that generates its hidden state representation h_t as a concatenation of the forward and backward LSTM states. The third layer is the attention layer that learns which states to focus in particular and generates the attention-focused hidden state representation l_t. The last layer is a fully connected layer that uses the softmax activation function to classify each element in the sequence.

3.3.1. *BERT Word embedding*

The preprocessed text needs to be encoded for computer recognition. A word vector is a row of real valued numbers where each point captures a dimension of the words meaning and where semantically similar words have similar vectors. Commonly used word vectors are static word vectors (e.g., word2vec Mikolov et al. (2013) and Glove Pennington et al. (2014)) and dynamic word vectors (e.g., ELMo Peters et al. (2018) and BERT Devlin et al. (2018)). Compared to static word vectors, dynamic word vectors can make the same word have different word vectors in different contexts. The dynamic word vector which can solve the problem of polysemy is very suitable for extracting sensitive information. Because the same word may represent different types of sensitive information. For example, under normal circumstances, a name belongs to Personal information. However, some people use their name as a password, in which case the name represents Secret and Credential information.

The preprocessed sequence is $X = \{x_1, x_2, x_3, \cdots, x_n\}$, where x_t is the t-th word in the sentence. After calculation by BERT, a word vector $E = \{e_1, e_2, e_3, \cdots, e_n\}$ of the word sequence X is generated, where e_t is the word vector of x_t.

3.3.2. *Bidirectional LSTM*

LSTM is developed to address the vanishing gradient problems of RNN. A basic LSTM cell consists of various gates to control the flow of information through the LSTM connections. Given the inputs e_t and h_{t-1}(previous hidden vector state), an LSTM cell performs various non-linear transformations to generate a hidden vector state h_t. The transformations are as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, e_t] + b_f\right) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, e_t] + b_i) \tag{2}$$

$$C_t = \tanh(W_C \cdot [h_{t-1}, e_t] + b_C) \tag{3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, e_t] + b_o) \tag{5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{6}$$

Where W represents a weight matrix connecting two layers, b represents a bias vector, C represents the cell state, and

σ and tanh are activation functions. i_t is the input gate, f_t is the forget gate, and o_t is the output gate.

BiLSTM capture both the past features and the future features via forward and backward states respectively. Therefore, we can create a new hidden vector by using the hidden vector representations from forward $\overrightarrow{h_t}$ and backward $\overleftarrow{h_t}$ LSTM:

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{7}$$

3.3.3. *Attention mechanism*

The objective of the attention layer is to capture the most important semantic information in a sequence, rather than focusing on all the information. The attention mechanism used in our model references the approach of Zheng et al. (2018). In the sequence labeling task, we highlight the important tokens in a given input sequence by using the attention mechanism. The attention mechanism is implemented as follows:

$$m_{t,t'} = \tanh\left(W_m h_t + W_{m'} h_{m'} + b_m\right) \tag{8}$$

$$a_{t,t'} = \sigma\left(W_a m_{t,t'} + b_a\right) \tag{9}$$

Where, $a_{t,t'}$ is an element of the attention matrix and is used to capture the similarity between the hidden state representations h_t and $h_{t'}$. W_m and $W_{m'}$ are the weight matrices corresponding to the hidden states h_t and $h_{t'}$. W_a is the weight matrix corresponding to their non-linear combination. b_m and b_a are the bias vectors.

The attention-focused hidden state representation l_t generated by the attention mechanism is the weighted summation of $a_{t,t'}$ and $h_{t'}$.

$$l_t = \sum_{t'=1}^{n} a_{t,t'} \cdot h_{t'} \tag{10}$$

4. Experiments

4.1. *Datasets*

The datasets used in the paper are collected from Pastebin (https://pastebin.com). This website is a text sharing platform where users can store any text. Pastebin contains voluminous sensitive information. For example, some users accidentally uploaded personal information, password credentials, and financial information. Developers and engineers leaked internal configurations and API keys. In addition, several hackers uploaded illegally obtained sensitive information to Pastebin.

We collected public documents in Pastebin from November 11, 2019, to February 1, 2020, for a total of 1,035,634 documents. For training the BERT-BiLSTM-Attention model, we manually collected some text containing sensitive information from 12,673 documents. According to the data preprocessing method in Section 3.1, we obtained 144,967 text sequences as training data. The number of sequences for each type is shown in Table 4. The preprocessed text sequence is manually labeled according to the ”BIO” labeling strategy.

Table 4 – The statistics of sensitive information sequences.

Type	Amount
Personal information	15,885
Network identity information	7327
Secret and Credential information	39,184
Financial information	10,221
Non-sensitive information	72,350
Total	144,967

4.2. *Experimental environment and evaluation metrics*

The experimental environment is shown as follows:

Hardware configuration: Centos 7 operating system; 6-core CPU; 16GB memory.

Software configuration: Python 3.6; keras 2.3.1; tensorflow 1.14.0.

We use precision (P), recall (R) and F1-score to measure the performance of the BERT-BiLSTM-Attention model. The specific calculation formula is shown as follows:

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$R = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{13}$$

True Positive (TP) indicates that the model predicts the sensitive information correctly. False Positive (FP) indicates that the model predicts the non-sensitive information incorrectly. False Negative (FN) indicates that the model predicts the sensitive information incorrectly. True Negative (TN) indicates that the model predicts the non-sensitive information correctly. As shown in equations, the precision rate represents the accuracy of predicting positive samples. The recall rate represents the probability of being predicted positive samples in the actual positive samples. The F1-score comprehensively reflects the precision rate and recall rate.

4.3. *Experimental settings*

We follow the splits of training (101479 sequences), validation (14496 sequences) and test (28992 sequences) data. The ratio of training, test and validation set is 7: 2: 1.

In the model training, we tune the parameters according to the F1-score. The setting of model parameters has a significant impact on the detection rate of sensitive information. The parameters are set as shown in Table 5. Among them, the sequence length is the length of text sequences input to the sequence labeling model. In the embedding layer, the static word vector model (i.e., word2vec) is ”GoogleNews-vectors-negative300” and the dynamic word vector (i.e., BERT) is ”uncased_L-12_H-768_A-12”. The LSTM hidden size is the number of hidden units in the BiLSTM layer. In the attention layer, the activation function is sigmoid. Dropout is a way to

Table 5 – Parameter setting.

Parameter	Value
Sequence length	32
Word embedding	768(BERT) or 300(word2vec)
LSTM hidden size	256
Dropout	0.4
Epoch	100
Optimizer	RMSprop
Learning rate	0.0001
Batch size	512

prevent overfitting by randomly dropping units from the neural network during training. Other hyper-parameters include epoch, optimizer, learning rate, and batch size.

4.4. *Effect of BERT-BiLSTM-Attention*

In this section, in order to verify the feasibility of the BERT-BiLSTM-Attention model, we analyze the effects of different word vectors and popular sequence labeling models to extract sensitive information through comparative experiments. Some of these sequence labeling models (BiLSTM, BiLSTM-CRF) have been applied to DLP systems Gomez-Hidalgo et al. (2010); Ong et al. (2017); Park et al. (2020), which utilize common NER tools to detect named entities such as people, location, and organization in data breaches. Comparative experiments are divided into two groups. The first group compares the performance of each model when using static word vector word2vec, and the results are shown in Table 6. The second group compares the performance of each model when using the dynamic word vector BERT, and the results are shown in Table 7. It can be seen from the results of Table 6 and Table 7 that the performance of the BERT-BiLSTM-Attention model is superior to other models.

4.4.1. *Effect of different word vector models*

Static word vectors will merge multiple semantics of polysemy words, and will not change according to the context after training. Therefore, static word vectors cannot solve the problem of polysemy. Dynamic word vectors can dynamically adjust the word embedding according to the current context. As mentioned in Section 3.3.1, polysemy often appears in sensitive information.

Here, we use the static word vector (word2vec) and dynamic word vector (BERT) to implement word embedding. As shown in Table 6 and Table 7, the performance of each model based on BERT has a significant improvement over the performance based on word2vec. The results prove that better performance can be obtained by using a dynamic word vector to generate word embedding vectors.

4.4.2. *Effect of different sequence labeling models*

Although LSTM/GRU has advantages for sequence encoding, it can only learn contextual information through the previous word or the latter word. The difference is that the attention mechanism can assign different weights to each word of the input sequence, and highlight more critical and important information so that the model can make more accurate judgments. According to our empirical practice, the attention mechanism does not bring greater computation and storage costs.

We compare the performance of the BiLSTM-Attention model with other baselines. As shown in Table 7, the performance of BiLSTM-Attention is the best, with an F1 score of 99.15%. Comparing the performance of BiLSTM-Attention and BiLSTM, the results prove that adding the attention mechanism can improve the accuracy of extracting sensitive information.

4.4.3. *Visualization of attention matrix*

We visualize the attention matrix to explain the models decisions that highlight important tokens. For example, ”test2001w@gmail.com: test1234” is a sentence containing sensitive information, where ”test2001w@gmail.com” is an email and ”test1234” is a leaked password. The preprocessing of this sentence is shown in Fig. 2. The input sequence is ”['email', 'test', '##200', '##1', '##w', 'gma', '##il', ':', 'test', '##12', '##34']”. Fig. 4 is a heat map of the attention matrix. Each element of the heat map highlights the importance of a token with respect to its neighboring context. Darker color represents important tokens.

In Fig. 4, we observe some black boxes located in the center. This proves that ”email test ##200 ##1 ##w gma ##il” plays a key role in predicting the label of ”test” (i.e., ”B-S”, the beginning of the password). Because when sensitive information is leaked, the email and password are usually leaked together.

4.5. *Exsense vs individual methods*

In this section, we evaluate the performance of content-based analysis (i.e., Regular matching), context-based analysis (i.e., BERT-BiLSTM-Attention), and ExSense respectively. In total, 4162 documents containing sensitive information are selected for testing, which included 713,738 sensitive information entities. As mentioned in Section 1, appropriate methods should be adopted for different sensitive information. Table 8 shows the detection rates for different methods. It can be seen that ExSense has a high detection rate compared to individual methods. This indicates that the combined method is more effective in extracting sensitive information.

4.6. *Sensitive information extraction results*

We extracted sensitive information from 1,035,634 documents on Pastebin and found sensitive information in 93,174 documents. Fig. 5 displays the extracted sensitive information by a visualization format to highlight important textual data points. We create word clouds that convey crucial information for different types of sensitive information. The following is the analysis of sensitive information keywords and the display of related sensitive information samples.

Fig. 5 (a) shows the keywords of personal information. Personal information includes information relating to identifiable individuals, such as name, address, date of birth, SSN, and phone number. The data samples are as follows:

First Name : S*****

Last Name : H**** C******

Date of Birth : **/*/19**

Table 6 – Results for each model when using word2vec.

Model	Precision	Recall	F1-score
W2V+BiGRU Chung et al. (2014)	81.31%	84.52%	82.28%
W2V+BiGRU+CRF Lerner et al. (2020)	87.63%	86.16%	86.76%
W2V+BiLSTM Yang et al. (2016)	82.40%	86.39%	84.35%
W2V+BiLSTM+CRF Huang et al. (2015)	88.18%	87.47%	87.82%
W2V+CNN+LSTM Wang et al. (2016)	76.21%	74.92%	75.56%
W2V+BiLSTM+Attention	89.81%	86.42%	88.08%

Table 7 – Results for each model when using BERT.

Model	Precision	Recall	F1-score
BERT+BiGRU Chung et al. (2014)	98.22%	98.73%	98.47%
BERT+BiGRU+CRF Lerner et al. (2020)	98.80%	99.28%	99.04%
BERT+BiLSTM Yang et al. (2016)	98.34%	99.40%	98.87%
BERT+BiLSTM+CRF Huang et al. (2015)	98.71%	99.31%	99.01%
BERT+CNN+LSTM Wang et al. (2016)	98.11%	99.29%	98.69%
BERT+BiLSTM+Attention	**98.72%**	**99.58%**	**99.15%**

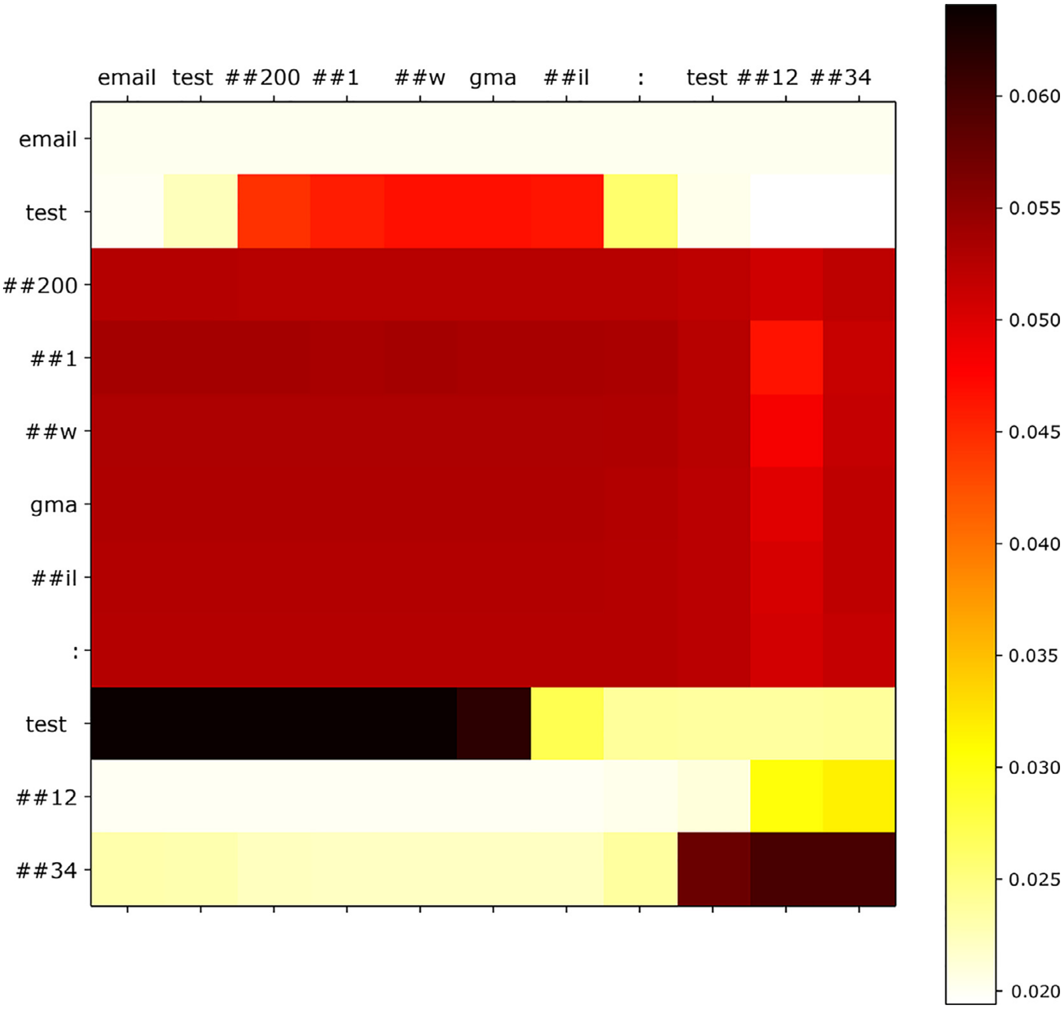
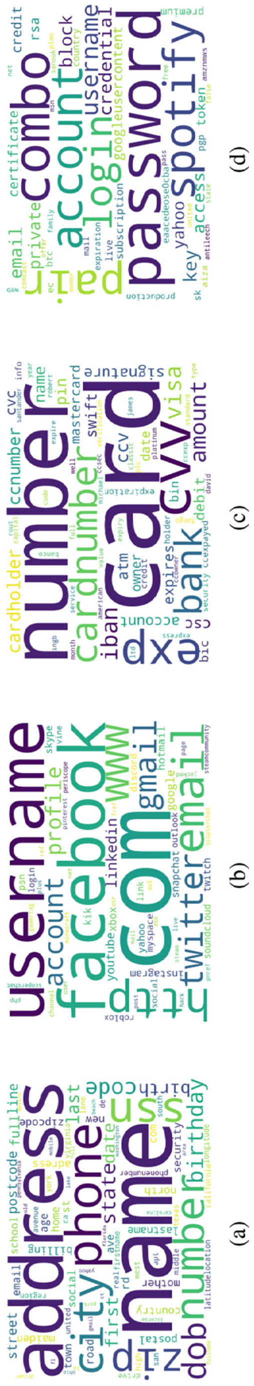


Fig. 4 – Attention matrix heat map.



(a)

(b)

(c)

(d)

Fig. 5 – Sensitive information word clouds.(a) Personal information, (b) Network identity information, (c) Financial information and (d) Secret and Credential information.

Table 8 – Comparsion of Exsense and individual methods.

Methods	Detection rate
Content-based analysis	29.60%
Context-based analysis	88.24%
ExSense	95.60%

*Street Adress : S**** ***
*Country : ***
*Town/City : M***** ******* *****
*State/Province : *******
*Zip Code : 7*****
*Phone Number : 0***********

Fig. 5 (b) shows the keywords of network identity information. The extraction of network identity information includes not only email and social media accounts but also IP and MAC addresses of network devices. The data samples are as follows:

*Email: e******@hotmail.com*
*Second email: e******@gmail.com*
*Instagram: W************** (as of 2015)*
*Kik:L**********, x********, (main)G********
*Vine accounts: https://vine.co/u/1*******************
*Youtube: G*********
*Instagram: G********
*GooglePlus: https://plus.google.com/+G********/posts*
*Psn: U***********, A**********, j***************
*Facebook: https://www.facebook.com/j*******.*.********
*Skype:g********
*MAC:02:**:**:**:**:***
*IP:1**.***.***.****

Fig. 5 (c) shows the keywords of financial information. Financial information is usually leaked bank account and credit card information. The data samples are as follows:

*Card Type : *******
*Credit Card Number : 46***************
*Exp. Date : ** / 20***
*Name On Card : S****** ** *****
*Cvv2 : 5***
*ATM Pin: 5****
*IBAN: BE** **** **** *****
*SWIFT BIC: C********

Fig. 5 (d) shows the keywords of secret and credential information. Secret and credential information involves login password combinations, keys, certificates, etc. The data samples are as follows:

*Email : b***************@aol.com*
*Password : n*********
*Combo : b***************@aol.com:n*********
*Login to HornyHostel with Username :n*********** Password: H************
*A**************@yahoo.com:i**********
*a*********:r************
*AIza************************************
——BEGIN PRIVATE KEY——
(Content omitted)
——END PRIVATE KEY——
——BEGIN CERTIFICATE——
(Content omitted)
——END CERTIFICATE——

5. Conclusion and future work

In this work, we present ExSense, a sensitive information extraction method from unstructured data. ExSense utilizes a hybrid approach that combines content and context analysis. Appropriate methods (i.e., content analysis and context analysis) are used for different sensitive information. In content analysis, regular expressions are used to extract sensitive information with predictable patterns. In context analysis, we build a sequence labeling model named BERT-BiLSTM-Attention, which extracts sensitive information from the text based on contextual features. To validate the BERT-BiLSTM-Attention model, we compare the experimental result with all baselines, and the result shows that our proposed model indeed can get more outstanding performance. Besides, it is shown on real datasets that ExSense is more effective than using content analysis or contextual analysis alone. Finally, we analyzed the public text on Pastebin, and the results proved that ExSense can effectively extract sensitive information from unstructured data.

In a word, our proposed framework can provide technical support for privacy protection. Specific application scenarios may include: deployed on the organization's network boundary to prevent data leakage by detecting data flow; performing real-time detection of unstructured data exposed on the Internet to provide threat intelligence analysts with early warning of data leakage events.

A limitation of this study is that ExSense can identify limited types of sensitive information, but new types of sensitive information may emerge over time. In light of this limitation, we will explore the following directions in the future: explore open sensitive information extraction methods for extracting more and unknown sensitive information. Still try to use active learning algorithms to label and train voluminous unlabeled data.

Ethics Statement

In this section, we discuss issues related to the ethical conduct of this research.

The Institutional Review Board informed us that the data we collected was outside the scope of the review because we only collected public documents in Pastebin, which are publicly available on the Internet. In addition, we did not obtain research data through any illegal means.

The sensitive information involved in this study had already been leaked on the Internet before our collection. Nevertheless, we have taken several measures to prevent further leakage of sensitive information. All research data are stored on an internal server, isolated from the Internet. In order to protect privacy, the examples in Section 3.1 are for illustrative purposes only, not real sensitive information. The samples of sensitive information presented in Section 4.6 are also desensitized.

The Pastebin data used in this work was used for research purposes only, and we did not use the sensitive information we collected to do anything illegal. We also did not cause any adverse consequences.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Yongyan Guo: Conceptualization, Methodology, Software, Data curation, Writing - original draft. **Jiayong Liu:** Conceptualization, Methodology, Investigation. **Wenwu Tang:** Investigation, Software, Data curation. **Cheng Huang:** Conceptualization, Methodology, Validation, Writing - review & editing.

Acknowledgments

This document is the results of the research project funded by the National Natural Science Foundation of China (No. 61902265), and Sichuan Science and Technology Program (No. 2020YFG0047, No. 2020YFG0076).

Appendix A

Table A1 – Regular expressions for sensitive information.

Type	Name	Regular expressions	
Network identity information	IP address	\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}	
	MAC address	([A-Fa-f0-9]{2}-){5}[A-Fa-f0-9]{2}	
	Email	[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+	
Secret and Credential information	API key/Token	Google	AIza[0-9A-Za-z\-_]{35} [0-9]+-[0-9A-Za-z_]{32}\.apps\.google usercontent\.com
		Twitter	[1-9][0-9]+-[0-9a-zA-Z]{40}
		Facebook	EAACEdEose0cBA[0-9A-Za-z]+
		Picatic	sk_live_[0-9a-z]{32}

		Stripe	sk_live_[0-9a-zA-Z]{24}
			rk_live_[0-9a-zA-Z]{24}

		Square	sq0atp-[0-9A-Za-z\-_]{22}
			sq0csp-[0-9A-Za-z\-_]{43}

		Amazon	access_token$production$[0-9a-z]{16}$ [0-9a-f]{32}
			amzn\.mws\.[0-9a-f]{8}-[0-9a-f]{4}-[0-9 a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}

		Twilio	AKIA[0-9A-Z]{16}
		MailGun	SK[0-9a-fA-F]{32}
		MailChimp	key-[0-9a-zA-Z]{32}
	Key and Certificate	RSA Private Key	——BEGIN RSA PRIVATE KEY—— [\r\n]+(?:\w+:.+)*[\s]*(?:[0-9a-zA-Z+\/=]{64,76}[\r\n]+)+[0-9a-zA-Z+\/=]+[\r\n]+ ——END RSA PRIVATE KEY——
		EC Private Key	——BEGIN EC PRIVATE KEY—— [\r\n]+(?:\w+:.+)*[\s]*(?:[0-9a-zA-Z+\/=]{64,76}[\r\n]+)+[0-9a-zA-Z+\/=]+[\r\n]+ ——END EC PRIVATE KEY——
		PGP Private Key	——BEGIN PGP PRIVATE KEY BLOCK—— [\r\n]+(?:\w+:.+)*[\s]*(?:[0-9a-zA-Z+\/=]{64,76}[\r\n]+)+[0-9a-zA-Z+\/=]+[\r\n]+=[0-9a-zA-Z+\/=]{4}[\r\n]+ ——END PGP PRIVATE KEY BLOCK——
		General Private Key	——BEGIN PRIVATE KEY—— [\r\n]+(?:\w+:.+)*[\s]*(?:[0-9a-zA-Z+\/=]{64,76}[\r\n]+)+[0-9a-zA-Z+\/=]+[\r\n]+ ——END PRIVATE KEY——
		Certificate	——BEGIN CERTIFICATE—— (?:[\NA-ZA-Z0-9+/]{4}*(?:[A-Z A-Z0-9+/]{2}==|[A-ZA-Z0-9+/]{3}=) ——END CERTIFICATE——
		Certificate Request	——BEGIN CERTIFICATE REQUEST—— (?:[\NA-ZA-Z0-9+/]{4}*(?:[A-ZA-Z0-9+/]{2}==|[A-ZA-Z0-9+/]{3}=) ——END CERTIFICATE REQUEST——

REFERENCES

Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K. A brief survey of text mining: classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919 2017.

Alneyadi S, Sithirasenan E, Muthukkumarasamy V. Detecting data semantic: a data leakage prevention approach, 1. IEEE; 2015. p. 910–17.

Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 2014.

Cadwalladr C, Graham-Harrison E. Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. The guardian 2018;17:22.

Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 2014.

Devlin J, Chang M-W, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 2018.

Gomez-Hidalgo JM, Martin-Abreu JM, Nieves J, Santos I, Brezo F, Bringas PG. Data leak prevention through named entity recognition. In: 2010 IEEE Second International Conference on Social Computing. IEEE; 2010. p. 1129–34.

Gupta S, Gupta S. Natural language processing in mining unstructured data from software repositories: a review. Sādhanā 2019;44(12):244.

Hart M, Manadhata P, Johnson R. Text classification for data loss prevention. In: International Symposium on Privacy Enhancing Technologies Symposium. Springer; 2011. p. 18–37.

Huang Z, Xu W, Yu K. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 2015.

IBM Security, P.I., 2019. Cost of a data breach report. https://www.all-about-security.de/fileadmin/micropages/Fachartikel_28/2019_Cost_of_a_Data_Breach_Report_final.pdf.

Katz G, Elovici Y, Shapira B. Coban: a context based model for data leakage prevention. Inf. Sci. (Ny) 2014;262:137–58.

Lerner I, Paris N, Tannier X. Terminologies augmented recurrent neural network model for clinical named entity recognition. J. Biomed. Inform. 2020;102:103356.

Lin C-H, Yang P-K, Lin Y-C. Detecting security breaches in personal data protection with machine learning. In: 2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM). IEEE; 2020. p. 1–7.

Ling W, Luís T, Marujo L, Astudillo RF, Amir S, Dyer C, Black AW, Trancoso I. Finding function in form: compositional character models for open vocabulary word representation. arXiv preprint arXiv:1508.02096 2015.

Mathew S, Petropoulos M, Ngo HQ, Upadhyaya S. A data-centric approach to insider attack detection in database systems. In: International Workshop on Recent Advances in Intrusion Detection. Springer; 2010. p. 382–401.

Meli M, McNiece MR, Reaves B. In: NDSS. How bad can it git? characterizing secret leakage in public github repositories.; 2019.

Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 2013.

Mnih V, Heess N, Graves A, et al. Recurrent models of visual attention. In: Advances in neural information processing systems; 2014. p. 2204–12.

Nguyen N, Guo Y. Comparisons of sequence labeling algorithms and extensions. In: Proceedings of the 24th international conference on Machine learning; 2007. p. 681–8.

Noor U, Anwar Z, Malik AW, Khan S, Saleem S. A machine learning framework for investigating data breaches based on semantic analysis of adversarys attack patterns in threat intelligence repositories. Future Generation Computer Systems 2019;95:467–87.

Ohm P. Sensitive information. S. Cal. L. Rev. 2014;88:1125.

Ong YJ, Qiao M, Routray R, Raphael R. Context-aware data loss prevention for cloud storage services. In: 2017 IEEE 10th International Conference on Cloud Computing (CLOUD). IEEE; 2017. p. 399–406.

Park J-s, Kim G-w, Lee D-h. Sensitive data identification in structured data through genner model based on text generation and ner. In: Proceedings of the 2020 International Conference on Computing, Networks and Internet of Things; 2020. p. 36–40.

Passos A, Kumar V, McCallum A. Lexicon infused phrase embeddings for named entity resolution. arXiv preprint arXiv:1404.5367 2014.

Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–43.

Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. Deep contextualized word representations. arXiv preprint arXiv:1802.05365 2018.

Shapira Y, Shapira B, Shabtai A. Content-based data leakage detection using extended fingerprinting. arXiv preprint arXiv:1302.2028 2013.

Shu X, Yao D, Bertino E. Privacy-preserving detection of sensitive data exposure. IEEE Trans. Inf. Forensics Secur. 2015;10(5):1092–103.

Shu X, Zhang J, Yao DD, Feng W-c. Fast detection of transformed data leaks. IEEE Trans. Inf. Forensics Secur. 2015;11(3):528–42.

Shvartzshnaider Y, Pavlinovic Z, Balashankar A, Wies T, Subramanian L, Nissenbaum H, Mittal P. Vaccine: Using contextual integrity for data leakage detection. In: The World Wide Web Conference; 2019. p. 1702–12.

Trabelsi S. Monitoring leaked confidential data. In: 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS). IEEE; 2019. p. 1–5.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.

Verizon, 2019. 2019 data breach investigations report. https://enterprise.verizon.com/resources/executivebriefs/2019-dbir-executive-brief.pdf.

Wang J, Yu L-C, Lai KR, Zhang X. Dimensional sentiment analysis using a regional cnn-lstm model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers); 2016. p. 225–30.

Yang Z, Salakhutdinov R, Cohen W. Multi-task cross-lingual sequence tagging from scratch. arXiv preprint arXiv:1603.06270 2016.

Yubo, C., Liheng, X., Kang, L., Daojian, Z., Jun, Z., et al., 2015. Event extraction via dynamic multi-pooling convolutional neural networks.

Zheng G, Mukherjee S, Dong XL, Li F. Opentag: Open attribute value extraction from product profiles. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2018. p. 1049–58.

Yongyan Guo is currently pursuing his masters degree in the College of Cybersecurity, Sichuan University, China. His current research interests include data breach protection, attack detection, and artificial intelligence.

Jiayong Liu received his B.Eng. degree in 1982, M. Eng. degree in 1989, and Ph.D. degree in 2008 from Sichuan University, China. He is currently a professor in School of Cybersecurity, Sichuan University, China. His research interests include network information processing and information security, communications and network information system.

Wenwu Tang received the masters degree from SichuanUniversity, Chengdu, China, in 2010. and received the certificate of Network Security Engineer in 2014. At present, his main research directions include web security, big data application, computer forensics, attack detection and other fields.

Cheng Huang received the Ph.D degree from SichuanUniversity, Chengdu, China, in 2017. From 2014 to 2015, he was a visiting student at the School of Computer Science, University of California, CA, USA. He is currently an Assistant Research Professor at the college of Cybersecurity, Sichuan University, Chengdu, China. His current research interests include Web security, attack detection, artificial intelligence.