Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-15623-3

A privacy-preserving word embedding text classification model based on privacy boundary constructed by deep belief network

Bo Ma[1] · Edmund Lai[1] · Wei Qi Yan [1] · Jinsong Wu[2]

Received: 15 October 2021 / Revised: 22 August 2022 / Accepted: 22 April 2023

Abstract
To effectively extract and classify the information from reports or documents and protect the privacy of the extracted results, we propose a privacy classification named Word Embedding Combination Privacy-preserving Support Vector Machine (WECPPSVM) model to classify the text. In addition, this paper also proposes the Privacy-preserving Distribution and Independent Frequent Subsequence Extraction Algorithm (PPDIFSEA), which calculates the degree of independence of the training data input to the classification model by training the Deep Belief Network(DBN) in PPDIFSEA, then obtains the Privacy Boundary(PB). PB is an indispensable condition for both data sampling and privacy noise generation. And this model can protect privacy by injecting the privacy noise into the classification result, this method can interfere with the background knowledge-based privacy attack. Our quantitative analysis shows that the WECPPSVM proposed in this paper can approach mainstream text classification algorithms in terms of text classification accuracy while preserving privacy without increasing computational complexity. In addition, the fusion study and privacy threat evaluation also verify that the proposed PPDIFSEA method combined with WECPPSVM achieves an acceptable level of classification accuracy and privacy protection.

Keywords Privacy-preserving · Support vector machine(SVM) · Independence degree(ID) · Word embedding · Deep belief network(DBN) · Privacy boundary(PB)

1 Introduction

Word embedding is a language modeling in natural language processing (NLP); in this type, the word and phrases will reflect in the vectors. The mapping process from observation to

✉ Bo Ma
rcn4743@aut.ac.nz

[1] School of Engineering, Computer & Mathematical Sciences, Auckland University of Technology, 55 Wellesley Street East, Auckland 1010, Auckland, New Zealand

[2] Department of Computer Sciences, Universidad de Chile, Av. Libertador Bernardo O'Higgins, Santiago 1058, Región Metropolitana, Chile

Published online: 15 September 2023

vector can process by probability model, neural networks, dimension reduction [37]. The word vector generated by the word vector model is input into the classification model for classification tasks, and its accuracy is higher [23, 52].

However, privacy leakage will happen when using word embedding models. In 2019, Miles proposed a membership attack methods [44] to attack the personal data-sets via the weakness of word embedding in text classification task [24].

1.1 Motivations

Text classification using the Word embedding model is an important step in extracting the information from documents. Using the word embedding model, naive Bayes, k-nearest neighbors, and support vector machine are three typical methods for text classification [36]. In the natural language process(NLP), Word embedding is a method of encoding the meaning of words. It uses the storage form of real-valued vectors to make the distribution of some words in vector spaces with similar purposes similar.

However, Naive Bayes (NB) and k-nearest neighbors require the input texts to have relatively low dimensions (less than 10). Although document vectors' dimensionality can be reduced by restricting the number of feature words, if there are synonyms in the contexts, the model cannot reduce the dimensionality by filtering feature words [8]. Considering this limitation, only thousands of words can be selected in natural language processing. However, the normal documents are complicated and heterogeneous, which are tough to be expressed through those thousands of words. In addition, if the dimensionality of vector documents is reduced, extracting or training work may expose sensitive information, even if the classification result for documents or sentences in the documents can be protected by utilizing differential privacy [43].

In detail, the sensitive subject S_i as privacy information is the subject's identity from a text piece. In addition, the author's name as privacy information in the text also belongs to one of a feature in the document. The target of a document may contain complex information. In the pre-processing stage, however, the main task is to reduce the complexity of the document to reduce the training cost for the machine learning model. Although this method can be finished via dimensional reduction since dimensionality reduction may reveal the document distribution, it will suffer privacy attacks by using statistical membership attack methods [44]. Therefore, we improve the existing word embedding model support with a privacy-protected Support Vector Machine(SVM) method. Because support vector machines can increase the dimension of training data instead of reducing it, this makes it difficult for privacy attackers to face high-dimensional training text. Increasing the dimension, and adding the interfering substance can improve the privacy level of the model with limited loss of the accuracy of the process. Among them, if the added interference data is very similar to the privacay data, the privacy attacker can mistakenly recognize that the interference data is the privacy data so that the private data cannot be accurately guessed. In this paper, we use the method of adding privacy noise to act as the work of adding interference and the method of using support vector machines to increase the dimensionality and thus improve the process privacy security.

1.2 Threat model

In most existing Natural Language Process applications, the sensitivity of the embedding vector is usually defined or calculated by the user [11]. Moreover, these custom-sensitive vectors and their calculation framework are usually calculated via the public cloud, thereby keeping the privacy of the original data in the user's local area, but the embedded model as a general model will be shared by the service provider or platform in each user terminal.

In certain conditions, when sharing the training models may be "more privacy-protecting" than sharing raw data. But the model itself contains previously trained data information, and this private information can be extracted from the embedded model. In the threat model of this paper, what kind of sensitive information is included in the word embedding model? Can the attacker extract the victim's sensitive information through some methods?

- Assume that D_{train} is a training data set formed by the victim Eli with personal private information, which may contain sensitive information x_s.
- E is an embedded model, its model is shared, and allows anyone to add calculation tasks as $\Pi(x)$.
- R_{target} is the embedding vector composed of the sensitive target data set.
- Assume Bob is the attacker. He uses the online public data set as B_d. This data set includes the personal data set D_{train} used by Bob. The two data sets have the same distribution and the extracted data contains part of which has been marked data and unlabeled raw text data.

In most existing Natural Language Process(NLP) applications, the sensitivity of the embedding vector is usually defined or computed by the user [11]. Moreover, these custom-sensitive classes and their text classification frameworks are usually inference through the public cloud, and the results predicted by the machine learning model may need to be released publicly. Classification models allow privacy attackers to easily develop ways to guess privacy information.

Sharing trained models may be "more privacy-preserving" than sharing raw data, where cloud computing is required. But the model itself contains the previously trained data information, and this privacy information can be extracted from the embedded text classification model. Based on the previous information, it can be concluded that in the threat model mentioned in this paper, what privacy(sensitive) information may be contained in the word embedding model? Can the attacker extract or guess the victim's privacy(sensitive) information in some way?

- Suppose D_{train} is a training data-set formed by victim Eli with personal privacy information, which may contain sensitive information x_s.
- E is an embedded classification model whose model is shared and allows anyone to add computational tasks as $\Pi(x)$.
- R_{target} is the embedding vector consisting of the sensitive target data-set.
- Assume Bob is the attacker. He uses an online public data-set as B_d. This data-set includes the personal data-set D_{train}, and it contains Eli's information. Both datasets have the same distribution, and the extracted data contains partially labeled data and unlabeled raw text data.
- If Bob can obtain the same result from the text classification model as the published text classification prediction by selecting the training dataset in D_{train}, then Bob can obtain the privacy(sensitive) information x_s by statistical means.

1.3 Contributions

In order to solve the problem of privacy leakage mentioned in the aforementioned threat model, a secure machine learning model needs to leak as little privacy as possible while ensuring the availability of data and prediction results. Under this principle, in order to protect the privacy of text processing tasks to the greatest extent and limit the interference of the task itself, we propose a privacy-preserving text classification framework to enhance privacy protection and the framework has three main contributions:

- We insert the deep belief network into the independent calculation method to predict the distribution of input data-sets for seeking the privacy(noise) boundary, it provides the accuracy range of privacy noise sampling. The privacy(noise) boundary is the range for sampling input data sets and the sampling work is preparing to generate the privacy noise for the privacy-preserving method. At the same time, PPDIFSEA can also check whether the sub-string in the word vector belongs to the sensitive class in the later stage, so as to ensure that all classification results containing privacy can be protected.
- We improve our privacy-preserving method by sampling privacy noise from the privacy (noise) boundary with less loss of training accuracy for text classification tasks.
- We combine the Support Vector Machine to decrease the privacy leakage risk of the existing word embedding text classification model.

Among them, for the first contribution point, we propose to use a deep belief network to calculate the privacy boundary, which can help the model of the classification task to sample the training data, thereby generating the noise required to protect privacy. The scope of this sampling is defined by the privacy boundary. Users can input a privacy budget to guide the scheme to generate different levels of protection measures through the privacy budget. The second contribution point is to verify whether the privacy noise is appropriate through our proposed algorithm, so as to appropriately reduce the interference caused by the privacy noise to the classification accuracy. In order to support the privacy protection method mentioned in contribution point 2, the classification accuracy can still be guaranteed to an acceptable level, we propose a third contribution point to improve the performance of support vector machines in the text by combining word embedding classification accuracy.

1.4 Overview of proposed solution

Figure 1 shows an overview of the proposed architecture. The proposed word embedding-based privacy protection scheme is divided into four steps, including independence calculation, word embedding encoding step, classification model training step, and verification step. The type of input includes input data to be trained, classification labels corresponding to the input data, and part of the input unlabeled data is also included in the input data as data to be classified. The preset privacy budget, that is, which labels are

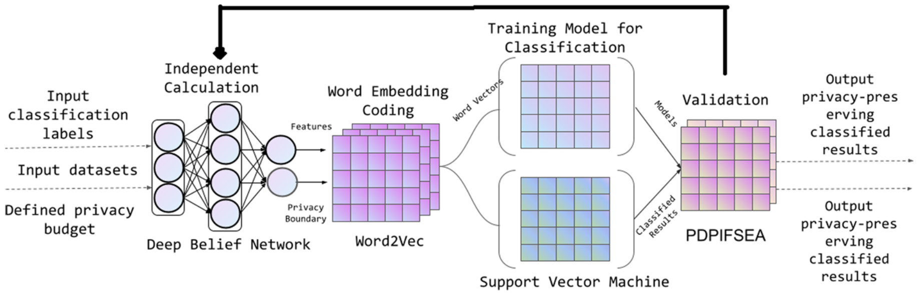


Fig. 1 The architecture of the proposed Word Embedding Combination Privacy-Preserving Support Vector Machine scheme

the privacy classes that need to be protected. The final output includes the privacy-protected classification result (this result is the unlabeled data in the input data that needs to be classified) as a privacy-preserving classification model for future inference. In the first step, we propose an independent calculation method using deep belief networks to help the model find the privacy boundaries of different words in the input data, and use the obtained privacy boundaries and preset privacy classes to Generate the corresponding noise. In the second step, word vectors [35] are generated from the pre-trained data through Word2Vec. The third step is to input the classified word vector data to train the support vector machine classification model, and the trained and verified model classifies the unlabeled data that needs to be classified. The final step is to add privacy noise to the classification results according to the privacy budget and verify the privacy level of the classification results. This step will use the Independent Frequent Subsequence Extraction Algorithm (PPDIFSEA) method to verify the privacy level. If the privacy boundary of the word vector needs to be changed or updated, the first step and the classification process will be repeated to achieve the protection level specified by the privacy budget.

2 Related work

2.1 Word embedding word2vec model

In order to extract keywords from heterogeneous word documents, an efficient way is to encode the input documents with a word embedding model [1, 15, 18]. This encoding work has to be completed before extraction. In Fig. 1, the second step is the word embedding(encode) process before the classification and sequence extraction.

Pertaining to word embedding approaches, word2vec is an efficient algorithm developed by Google [10]. Word2vec can compress data while capturing contextual information, which includes two main types, a Continuous Bag Of Words(CBOW) [17] and skip-gram [34]. The goal of CBOW is to predict the probability of current words based on the context, while skip-gram is to predict the probability of the current context. Both methods use (shallow or double) artificial neural networks as their classification algorithms and attain the optimized vectors of each word in k-dimensional space, which simplifies the text processing in the vector space.

Training neural network models by using word vectors can assist the Word2vec model in accurately extracting the contextual information, the similar words in the vector space are used to calculate the semantic similarity. For example,

$$\begin{aligned}&\text{vector}('\text{Warszawa'}) \text{ - } \text{vector}('\text{Poland'})\\&+\text{vector}('\text{New Zealand'}) = ('\text{Wellington'})\\&\text{vector}('\text{Boy'}) \text{ - } \text{vector}('\text{Man'})\\&+\text{vector}('\text{Woman'}) = ('\text{Girl'})\end{aligned} \tag{1}$$

where “New Zealand” and “Girl” are resultant terms, respectively. Moreover, this semantic relationship is obtained via not using prior knowledge such as WordNet [25] but using purely statistical methods such as Huffman coding to avoid the heavy workload of manual constructions. Word2vec, in essence, is considered a distributed representation in vocabulary vectorization.

2.2 Deep belief network and privacy boundary

Deep belief networks are also widely used in the field of natural language. Wusuo Li [4] et al summarized several methods used in the field of natural language processing, such as the Hidden Markov Model(HMM), Maximum Entropy Markov Model(MEMM), etc., to maximize the conditional probability corresponding to the predicted object in the word semantic prediction of the text, rather than simply extracting the words and sentences in the text.

For the word prediction problem using a Deep Belief Network, Wusuo Li [4] et al proposed the improved Deep Belief Networks(DBN) model by adding the Part-Of-Speech (POS) node, which samples the words in the training text through the DBN model and predicts the association between words. Compared with the CRF method, the accuracy rate of his method improves by 1.47%. In privacy protection, for privacy-related words, privacy attackers use the prediction method of word association to associate from the same privacy word class to obtain private information. The DBN model proposed in Li's paper can precisely predict the sensitive words(Patient's name) and extract that private information.

For text classification, Meng Wang [49] et al. proposed a word classification algorithm Information Geometry Deep Belief Networks (IGDBN). This method solves the problem of sentiment word classification with a large number of label comments in the real world by training the DBN network to predict, associate, and classify labels and sentiment words. In terms of privacy, the user's own information mentioned in the post can also cause privacy leakage. Such posts will be used by privacy attackers to extract privacy-sensitive words and form users' privacy portraits.

2.3 Privacy preservation for support vector machine

Many researchers combine differential privacy with deep learning to protect the training model against differential privacy attacks. However, due to the high sensitivity of the network output to parameters, the direct application of random noise in the deep learning model will produce poor performance. Therefore, previous researchers proposed to add random noise to the training stage for the machine learning model [42] update and make it privatized. This protection method can completely eliminate the memory effect and reduce the exposure to privacy.

When analyzing the (ε, δ)-differential privacy, here several paper mention about security performance [27, 28, 32, 50] mentioned. When using Gaussian process to generate covariance kernel in Reproducing Kernel Hilbert Space (RKHS), then the correct noise level can be measured by RKHS norm function, and the "sensitivity" level of the function will impact machine learning result if learning approaches need to analysis the "sensitivity" level of added noise. In addition, it can use kernel space under Abstract Wiener space [38]. This paper will focus on stochastic analysis with Abstract Wiener under the stream if it is followed with Gaussian Distribution. For Abstract Wiener spaces, the Banach space $C_{\mathbb{R}}$ is replaced by any separable Banach space B and a certain densely embedded Hilbert Space $\mathbb{H}$. The measure ρ_1 on $\mathbb{B}$ is in general the centered Gaussian distribution and variance are 1. The Malliavin derivative D is defined by density of operator $L^2(\mathbb{B}, \rho_1)$, its value is $L^2(\mathbb{B}, \rho_1, \mathbb{H})$ and its Bochner square map on Hilbert space [7] as $f : \mathbb{B} - \mathbb{H}$:.

In the scheme proposed in this paper, after embedding the word vector to be processed, its semantics can be spatially mapped in the word vector to reflect the part-of-word-vector association between the word vectors. After the word vector is spatially mapped, its space can also be processed into Hilbert space by basis transformation. For Abstract Wiener spaces [5], the measure of ρ_1 on $\mathbb{B}$ can help to separate the sensitive classes from general word vectors.

3 Training for text classification with word embedding model

To improve the process performance efficiency, we need to classify input data before the coding stage. As shown in Fig. 1, prior to coding, the second step is chiefly working for classification. Text classification is a process of assigning labels that map an unclassified text to the existing classes by using a label. However, it implies that the classifiers will always classify a text into one class. The labeled classes do not cover the entire labeled classes, because only the sampling data has been labeled by the tester. The margins identified for the classifier are applicable to different classes from the training data. The boundaries are not applicable to the discrimination, so not all classes can be identified through the classifiers. The problem identifying the boundaries based on the training data-set is academically referred to as multi-class classification (MCC) [19].

It is difficult for a classifier to determine the discrimination between sensitive(privacy) and non-sensitive(non-privacy) classes from the margin of the data distribution if the definitions of the classes are vague in the labeling time [20]. To find out the sensitive(privacy) classes and classes difference, we need to calculate the independence degree of the input data, this stage we named Independence Calculation. Then, the class identifies the margin and features related to the labels of the input data.

3.1 Using deep belief network based independence degree calculation for privacy boundary

In our proposed solution, the pre-task of classifying the target word vector is to sample the data to be trained and generate the noise required by the privacy protection method according to the preset privacy level, this privacy level we called Privacy Budget(PB). The PB can be divided into 50 levels, and each level means the quantity of privacy noise. This stage is called the sampling stage; the main purpose is to determine the independence degree for input data sets and sampling from those data. Then, we can choose the sampled data and generate the noisy data within the noise boundary or so-called privacy boundary. The subsequent classification stage can distinguish the privacy level of the detected word vectors by the noise boundary, that is, the sensitivity of the word vectors.

In the sampling stage, we need to use the independence degree to seek the mutation for finding the boundary of data belonging to different classes, especially between sensitive and non-sensitive classes. Independence Degree [3] is a kind of measurement method which originally identifies the assigning labels that map an unclassified text to the existing classes and identify the boundary and features of the distribution for the input data set the degree of independence (support), which needs to be calculated from annotation data. So the degree of independence can get a difference between each class. The privacy boundary defines the scope for distinguishing sensitive and non-sensitive data. All sensitive and non-sensitive classes must be calculated with their respective degrees of independence. At the same time, the privacy boundary chooses to extract the independence of sensitive classes to form. The subsequent privacy noise can be obtained by judging whether the data conforms to the sensitive class.

Different from the tester has labeled only the sampling data; for word vectors, we propose the Deep Belief apply to calculate the Independence (support) Degree(ID), then use ID to measure the distribution feature among different word vectors. After measuring the slimiest data sets from ID, we can obtain the noise (privacy) boundary. This noise (privacy) border can help to extract and generate privacy noise for privacy-preserving propose.

In our solution, the function of Deep Belief Networks(DBN) is to obtain observable variables from input data sets; the range of the observable variables will be the degree. Because DBN can infer the state of unknown variables and adjust hidden states to reconstruct observable data as much as possible. In detail, the degree of independence can extract from the record of the frequency occurrence F_{w_i} of the vocabulary w_i, and f_{w_i} stands for the frequency of independent vocabulary w_i, and F_{w_i} is the sum of frequency occurrence. Obviously, for any vocabulary w_i, $\sum_{\text{contains } w_i} f_w = F_{w_i}$ holds. Based on the above, we use the data w_i to train the DBN network and use the model to observe the F_{w_i}, thereby obtaining the degree of independence of w_i. By analogy, the degree of independence of other word vectors can also be observed. In the noise generation stage, a certain word vector should be used by the noise for different word vectors. It should also be generated according to the principle of the same degree of independence of the word vector. In calculating the degree of independence, we assume that $\zeta(i, j)$ is the correlation coefficient of a set of words w_i and w_j, one means correlation, 0 means irrelevant. It has (3.1),

$$\zeta(i, j) = \begin{cases} 1, & w_j \supseteq w_i \\ 0, & w_j \supsetneq w_i \end{cases} \tag{2}$$

We construct a $N \times N$ matrix and the frequency of independence as

$$\begin{array}{l} A = \begin{bmatrix} \zeta(1,1) & \cdots & \zeta(1,N) \\ \vdots & \ddots & \vdots \\ \zeta(N,1) & \cdots & \zeta(N,N) \end{bmatrix}, \\ \vec{f} = \begin{bmatrix} f_{w_1} \\ \cdots \\ f_{w_N} \end{bmatrix}, \quad \vec{F} = \begin{bmatrix} F_{w_1} \\ \cdots \\ F_{w_N} \end{bmatrix} \end{array}. \tag{3}$$

We have $\mathbf{A}\vec{f} = \vec{F}$. And $\vec{f}$ stands for the vector of the class of (independent) vocabulary. Thus, the independent support of each class of vocabulary has

$$\vec{f} = \mathbf{A}^{-1}\vec{F}. \tag{4}$$

Since the rank of the matrix $\mathbf{A}$ is generally high, a fast solution usually does not directly use the matrix inversion to obtain the result from $\vec{f}$. Firstly, the class of the vocabulary $\vec{f}$ is sorted according to the length of the string from small to large, that is, in the case of guaranteeing $i < j$ that all are satisfied $length(w_i) \leq length(w_j)$. In this stage, the class of the vocabulary from the input label and labeled data sets. After this, the corresponding factors in the matrix A has

$$i < j \Rightarrow \zeta(i, j) = 0. \tag{5}$$

Thus, the matrix $\mathbf{A}$ is an upper triangular matrix. For the upper triangular matrix, the banding method can be used to greatly reduce the computational complexity of the solution. From (4), the key issue is that $\vec{f}$ exists a real solution.

Proposition 1 *A is a square matrix in (3). The sufficient and necessary condition for $A\vec{f} = \vec{F}$ to have a unique solution is that the square matrix $\mathbf{A}$ has the inverse matrix $\mathbf{A}^{-1}$ with full rank.*

In Proposition 1, $\vec{F}$ as the vector of the categories can be calculated via independence degree $\vec{f}$. Before the next stage, The result from this step needs to be utilized to generate the noise for the privacy protection proposed.

Then we can use the matrix $\mathbf{A}$ as a mask to judge whether any value in sampling target sets is the inside range of privacy(noise) budget or not. So in this aspect, the $\mathbf{A}$ can be a privacy(noise) boundary. After we obtain the privacy(noise) boundary, we can use the noise generation method in the following section to produce the privacy noise.

3.2 Privacy noise generation with renyi-differential privacy

Two main approaches for privacy protection are (ϵ, σ)-Differential Privacy(DP) [12] and Renyi-differential privacy (RDP) [40], those two methods which protect the privacy of personal information by adding noise. Differential privacy essentially keeps two distributions approximate, and differential privacy uses maximum entropy so-called α to measure the similarity of two distributions.

In an opposite way, the Renyi-differential privacy (RDP) [40] uses more range of Renyi entropy α, due to the α is hard to measure from origin data-sets and privacy noise sets, but α can be regarded as equivalent to privacy boundary, and privacy boundary can be calculated from Independence Calculation, so this noising generation approach will not break the coherence in the input data sets for classification. For instance, if an attacker seeks the privacy associated with an individual, the attacker's inquiry would lead to the 'same' result and they are not able to obtain any correct sensitive attribute value from the probability of sensitive attributes associated with the classified data. Our solution will use Renyi-differential privacy to generate privacy noise.

Among them, Renyi-differential privacy (RDP) needs to know the distribution characteristics of noise and the range of noise generation, that is, the privacy boundary generated by the degree of independence calculation. The noise sampling and generation method in this paper uses the standard Renyi-differential privacy generation method, so we will not repeat it here.

3.3 Word embedding process with support vector machine

In the text classification task, the text that needs to be trained for classification usually has four steps in the prepossessing stage,

- Word segmentation.
- Word vector establishment.
- One-hot encoding.
- Sequence alignment.

The simplest word embedding method in the third step is One-Hot Encoding, but this embedding method occupies a large space dimension and cannot reflect the relationship between words. Therefore, we need a way to map the One-Hot vector into a low-dimensional embedding space. Here this paper uses a parameter matrix A learned from the training data to convert the One-Hot Encoding to a low-dimensional vector. Different from previous methods, we map the word vectors through a matrix similar to the kernel space $\mathbb{H}$ (such as the parameter matrix A), the kernel space $\mathbb{H}$ is a type of Reproducing Kernel Hilbert Space(RKHS) in Support Vector Machine(SVM). Due to the poor performance of the original One-hot encoding, and the data classification involves privacy-related classes and irrelevant classes, the mutual exclusion between classes needs to be considered, that is, if all texts belonging to sensitive classes belong to other classes, It also needs to be processed according to the sensitive class during classification, so the original classification can be divided into multiple classifications with mutually exclusive classes. Through the above steps, the vector after word embedding processing can be input into the classifier for classification training.

Then, the next challenge is how to partition the original classification into the optimal combination of mutually exclusive multi-classification. The so-called 'optimal' classification refers to the separation of the original classification into a number of independent classes. Thus, the smallest number of training samples are contained in each class (Fig. 2).

In a support vector machine, given a string $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ as a training sample N, the first sample x_i is extracted as the feature vector, then the class number of the first sample is denoted as $y_i \in 1, 2, \ldots, M$, thus, the classification algorithm is to find a function in the hypothesis space $\mathbb{H} : X \to Y$, where X is the input space, Y is the output space. For a given scoring function $f : X \times Y \to R$, the function $\mathbb{H}(\cdot)$ returns the smallest value of the scoring function $J(\mathbb{H}) = R_{emp}(\mathbb{H}) + \lambda C(\mathbb{H})$. Since the input space $S_{(N\times N)} = [s_{(i,j)}]_{(N\times N)}$ represents the difference between the sample labeling class $s_{i,j}$ (a.k.a, fusion matrix), the classifier y_i can predict the input class based on a supervised learning algorithm y_j, we believe that the classified data can be used directly after reviewing the obtained matrix $s_{i,j}$ via classifier training. The matrix $s_{i,j}$ serves as the basis for the cluster optimization.

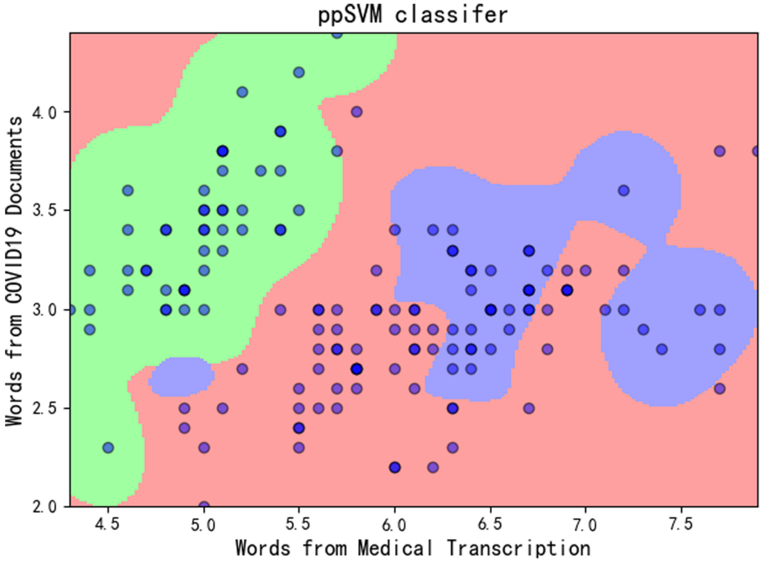


Fig. 2 Classification performance for Word Embedding Combined Privacy-preserving Support Vector Machine (WECPPSVM)

We expect the classification problem is: For the entire training set, the proportion of wrongly-classified samples to all the samples could be as small as possible. Then, after clustering, we set the actual optimization goal as follows: Let the fusion matrix $s_{i,j}$ be the number of training samples y_i that are labeled as a class with predicted γ by the K, given the classification algorithm $\gamma = U_{i=1}^{K}\gamma_i$ as the class $\gamma_\alpha \cap \gamma_\beta = \emptyset$. Then, the optimal clustering is to find a division of data-set and minimize the mis-classification rate is:

$$ER = \frac{W}{R+W} = \frac{\Sigma_{i=1}^{K}\Sigma_{i,j\in\gamma_i\&i\neq j}s_{i,j}}{\Sigma_{i=1}^{N}s_{i,i} + \Sigma_{i=1}^{K}\Sigma_{i,j\in\gamma_i\&i\neq j}s_{i,j}}. \tag{6}$$

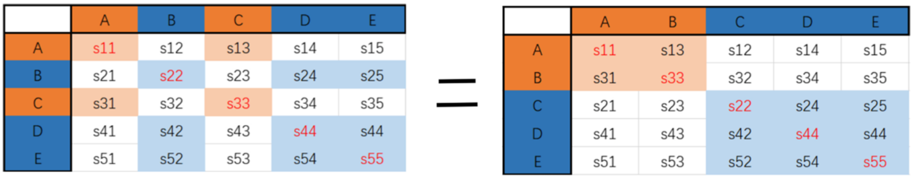
For example, Fig. 3 shows the fusion matrices of the five categories A, B, C, D, and E. Obviously, only diagonal elements are correctly classified, while other elements are mis-classified. Thus, the entire data set is mis-classified. The mis-classified rate out of $s_{1,1}, s_{2,2}, \ldots, s_{5,5}$ is $ER = 0.75$. Now, we consider the case partitioning all classes into two divisions. For example, $\{A, C\}$ and $\{B, D, E\}$ are two groups. If the first group $\{A, C\}$ are processed, we need to tackle all the elements within each cluster. In $\{A, C\}$, $s_{1,1} + s_{3,3}$ is the number of correctly classified samples, the number of wrongly-classified samples will be $s_{1,3} + s_{3,1}$. Similarly, within the group $\{B, D, E\}$, the number of samples $s_{2,2} + s_{4,4} + s_{5,5}$ is correctly classified, original clusters $s_{2,4} + s_{2,5} + s_{4,2} + s_{4,4} + s_{5,2} + s_{5,4}$ have the wrong number in the data-set. Then, according to this division, the mis-classified rate of the samples are $ER_{A,C} = 0.5$ and $ER_{B,D,E} = 2/3$, lower than 0.75. The goal of our optimization is to find an optimal division that minimizes the mis-classified rate in the word embedding model.

Definition 1 (Word embedding model combined privacy-preserving Support Vector Machine) In Proposition 1, we can add privacy preserving methods and create word embedding vector S, where S has the mapping $S \leftarrow W$, W belongs to Reproduce Kernel Hilbert Space(RKHS). Thus, Support Vector Machine(SVM) [45] is used to create a set. If privacy features are contained in the vector and can be attacked by using statistical methods or background knowledge.

In Definition 1 and Algorithm 1, we refine the Privacy-preserving Support Vector Machine(ppSVM) for the word embedding model and define the vector of the word frequency in our work, which will solve the mis-classification problem, and this approach is different from the previous privacy-preserving Support Vector Machine [39]. This is the main contribution of this paper.

For given classes, if a number of training samples are provided by the SVMs and each SVM only accepts two classes for each problem, the classification for each loop is calculated by using

$$\vec{f_{i.j}} = label\{\sum_{s\in S_{i,j}} a_s b_y (c_s' t + 1)^p + c_{i,j}\}, \tag{7}$$



	A	B	C	D	E
A	s11	s12	s13	s14	s15
B	s21	s22	s23	s24	s25
C	s31	s32	s33	s34	s35
D	s41	s42	s43	s44	s44
E	s51	s52	s53	s54	s55

=

	A	B	C	D	E
A	s11	s13	s12	s14	s15
B	s31	s33	s32	s34	s35
C	s21	s23	s22	s24	s25
D	s41	s43	s42	s44	s44
E	s51	s53	s52	s54	s55

Fig. 3 Clustering optimization for classification

where i and j are the classified for each sub-class respectively, and t is the classified samples which are able to be deducted [47] and constraint (8)

$$M_l = \arg \max_{i=1,\ldots,S_l} \{ \sum_{j=1, i \neq 1} \vec{f_{i,j}}(t) \}. \tag{8}$$

The work [39] treated the labeling training problem into the encrypted domain. Under the contexts of word processing, this paper needs to clarify the vector of words with frequency and other parameters, depending on the distributed bag of words(DBOW) [29] from input vectors.

In the word2vec Model, one question is that, if the input data as linear class occurs in kernel space, then the nonlinear classification can be transformed to linear classification through nonlinear classification. In addition, the dimension of a series of data in the linear Support Vector Machine(SVM) can be reduced from high-dimensional kernel space.

For example, there are two strings $\alpha =< \alpha_1, \alpha_2, \cdots ,\alpha_M >$ and $\beta =< \beta_1, \beta_2, \cdots, \beta_N >$. There exists such an integer k for any integer $i \in [1, N]$ that the relationship between two strings is $\beta_{k+i} = \alpha_i$. The kernel function as a nonlinear transformation represents the inner product between two spaces, for a function $K(\alpha, \beta)$, also named positive definite kernel, there is a mapping from the inner product space to the feature space $\phi(\alpha)$ for α, β in an input space,

$$K(\alpha, \beta) = \phi(\alpha) \bullet \phi(\beta) \tag{9}$$

According to (7), we obtain Multiple Non-linear Support Vector Machine(MNSVM) [46]. Then, if we need to obtain the privacy-preserving class, we calculate the classification with its labels. When we want to obtain the maximum values of $\sum \vec{A_{i,j}}(t)$, we seek how to minimize $\sum -\vec{A_{i,j}}(t)$ as follows,

$$\begin{aligned} &Max_l = \arg\max_{i=1,\ldots,S_l} \\ &\{\textstyle\sum_{j=1,i\neq 1} \vec{A}_{i,j}(\Theta_t, d_n)\} + N_{PrivacyNoise} \\ &Min_l = \arg\min_{i=1,\ldots,S_l} \\ &\{\textstyle\sum_{j=1,i\neq 1} -\vec{A}_{i,j}(\Theta_t, d_n)\} - N_{PrivacyNoise}. \end{aligned} \tag{10}$$

The algorithm for word embedding combined with the privacy-preserving Support Vector Machine algorithm is as follows:

The main function of the Algorithm 1 (short name as WECPPSVM) is to classify the privacy protection methods. This algorithm uses the support vector machine method. The input string S_{input} represents the number of the input vector. There is also the number of the support vector class of $S_{support}$. The support vector is the number performed in the calculation of the feature matching which class of the word vector, so the support vector can help to calculate the degree of independence of the word vector, then the model can compare with the existing Independence degree and decide the class it belongs to. $S_{feature}$ represents the support vector number of the word vector feature, $A_{sva}[S_{SupportVector}]$ represents the matrix A of frequency of independence, and $F_{in}[S_{supportDegree}]$ is the input vector phrase, and $b*$ represents the bias degree. The first step is to calculate the support degree according to the number of the input support vector mentioned above, and then update the target vector parameter F_{dist}, the F_{in} means the independence degree and $F_{in}[i].fe[k]$ is the means of irrelevant feature and $(A_{sva}[j].fe[k] - F_{in}[i].fe[k])^2$ is the new added target vector parameter with square 2 for add weight and ensure the value is absolute. In the iterative process, generate the privacy noise $N_{PrivacyNoise}$. When generating the decision matrix D, map the support vector to the Reproducing kernel Hilbert space, and then update the decision

Algorithm 1 Word Embedding Combined with Privacy-preserving Support Vector Machine(WECPPSVM) Algorithm

```
Input:
    $S_{input}$ (Input Vector Number) , $S_{ID}$ (Support Vector Number), $S_{feature}$ (The Feature Support Vector Number), $A_{sva}[S_{SupportVectorArray}]$ , $F_{in}[S_{ID}]$ (Input Vector Array), b* (bias)
Output :
    $D$ ( Decision Vector)

Calculated the Independence Degree of $F_{in}[S_{ID}]$
for E doach i in $S_{input}$
    Init: $D = 0$
    for E doach  j in $F_{dist} = 0$
        for E doachk in $S_{feature}$
            $F_{dist} += (A_{sva}[j].fe[k] - F_{in}[i].fe[k])^2$
        end for
        $\theta = \exp(-\lambda \times F_{dist})$
        $D = D + \arg\max_{j \in F_{dist}} A_{sva}[j] \times \theta + N_{PrivacyNoise}$
         $= D - \arg\min \sum_{j \in F_{dist}} (-A_{sva}[j] \times \theta - N_{PrivacyNoise})$
    end for
    $D = D + b*$
end for
return $D$
```

vector through $SVM(-\sum_j A(\Theta_t, d_n)) - N_{PrivacyNoise})$. Among them, $A(\Theta_t, d_n))$ is the kernel space we mentioned in the previous parts. The last stage will update the decision vector $D = D + b*$. It also records the existing word vector with its class. In addition, to guarantee the privacy level of vector and its sub-strings, we need to check whether the sub-strings in the vectors belong to the existing sensitive class, we will introduce a Privacy-preserving Distribution and Independent Frequent Sub-sequence Extraction Algorithm(PPDIFSEA) to find out this relationship. The sensitive class for privacy data sets is manually defined from original data sets.

4 Validation with PPDIFSEA

4.1 Verification based on gaussian distribution independent sub-sequence extraction

In the last stage, the main function of the verification process is to verify whether the results of the classification mentioned above are accurate. If the verification is performed only from the sensitive class and positive(correct classification) sample, the non-sensitive elements in the sensitive vector may not be removed, so here we need to observe the vectors classified as non-sensitive class(short as non-sensitive vectors or non-sensitive sample) in the sample data. Also mis-classified negative samples are very important, because if negative samples contain sensitive attributes but not be classified into the correct class(if non-sensitive class), they will not be protected. Lastly, how distinguishing the labeled data from non-sensitive vectors and extracting the non-sensitive samples is a complex challenge.

When the samples prepared for classifications may not belong to any one of the classes, the reason is that we might not know the probability distribution for this binary word vector, where the training sample can only provide positive samples for classification. The negative samples may not exist or are extremely difficult to be obtained. Then detecting the independence of each sample is an important basis for verifying categories, especially sensitive categories. Therefore, to find the independence degree for each vector more accurately, we compare and propose a method based on deep belief networks to distribute the input samples into different independent frequent sub-sequences, then separate and map the negative samples into the negative sub-sequences and extract the sub-sequence composed of these negative samples. Besides, we identify the probability distribution of those samples under the word vector.

For the method for the probability distribution of the document to vectors(Doc2Vec) [30], the normalized document vectors will be mainly distributed in a high-dimensional structure, the radius of this structure from the center constitutes a variance in each dimension, and it can be recognized as a discriminator for sub-sequences. Thickness can describe the dimension of its document vector. We further infer its radius and thickness from the statistical distribution. According to the nature of the chi-square test in statistics, for the degrees of freedom $k \to \infty$, there are

$$\sigma(b_{\mu}^{(k)} + W_{:,\mu}^{(k+1)^T} h^{(k+1)})^2 = \frac{\chi^2 - k}{\sqrt{2k} \to N(0,1)}. \tag{11}$$

Equation 11 is the degree of deviation between the actual observation value of the statistical sample and the theoretical inferred value. In this equation, the σ is the degree of deviation and $(b_{\mu}^{(k)} + W_{:,\mu}^{(k+1)^T} h^{(k+1)})^2$ is observation variable and k is degrees of freedom, W is the weight matrix and sampling parameter is h. When approaching the mean $\mu \to k$, the standard normal distribution of variance has $\sigma^2 \to 2k$. Since the number of Doc2Vec vector dimensions is very large (400+), we approximately estimate the maximum value of its density appears at the radius $r = k$. According to the law of distribution, before and after the mean, the range of density $r \in [-2\sqrt{2k}, 2\sqrt{2k}]$. Compared with the result from different distribution prediction approaches, the Deep Belief Network(DBN) [31] tends to have higher prediction accuracy, which covers 95% samples. The prediction work of DBN is expressed as follows.

$$\begin{gathered} P(h^{(l)}, h^{(l-1)}) \\ \propto \exp(b^{(l)^T} h^{(l)} + b^{(l-1)^T} h^{(l-1)} + h^{(l-1)^T} W^{(l)} h^{(l)} P(h_i^{(k)}) = 1 \mid h^{(k+1)}) \\ = \sigma(b_i^{(k)} + W_{:,i}^{(k+1)^T} h^{(k+1)}) \exists i, \exists k \in 1, \ldots, l - 2 P(v_i = 1 \mid h^{(1)}) \\ = \sigma(b_i^{(0)} + W_{:,i}^{(1)^T} h^(1)) \exists i \end{gathered} \tag{12}$$

where $P(h^{(l)}, h^{(l-1)})$ is probability of sample parameter h, $W^{(l)h(l)}$ is the weight matrix, and l is the index of the weight matrix. The second line is the interaction between different layers in the network. σ is the factor form of $\exp(b^{(k)} h^{(k)}) + h^{(k)})$.

4.2 Algorithm for PPDIFSEA

As Fig. 1 mentioned, this paper utilizes a deep belief network to predict the privacy boundary from the distribution of word vectors and then classify the word vectors with the Privacy-preserving Support Vector Machine(PPSVM as a classifier). The pseudo-code of privacy-preserving prediction and independent frequent sub-sequence extraction algorithm (PPDIFSEA) is provided in Algorithm 2.

In Algorithm 2, the input string $S = \{s_1, s_2, \cdots, s_N\}$, $s_i = < s_{i,1}, s_{i,2}, \cdots, s_{i,m} >$ is raw word vectors, ξ is support threshold which represents the output of vectors. And output result is F. After initializing the model node-set, the algorithm updates the weight function $h^{(l)}$ by establishing and updating the independence degree list (x) from deep belief network G in $G.UPDATE\_IC\_FROM\_DBN(x)$, that is, $h^{(l)} = \exp(b^{l^T} h^{(l)} + b^{l-1} h^{(l-1)} + h^{(l-1)} W^{(l)} h^{(l)})$, so as to use the updated weight function to obtain more effective results in classification training.In the second step of word embedding coding, the algorithm improves the nodes in the deep belief network by updating the relation tree composed of word vectors, so that the semantic connection between the classes can be established, so that the classes can be better updated from the original text. The set X is the node data of the relation tree, and the update process is $G.GET\_VERTEX(S_0)$, $X = \{x_1, x_2, \cdots, x_K\}$, after updating the relationship tree X, the node set Y of the weight network is also adjusted accordingly. Finally, through Algorithm 1, Word embedding combined with privacy preserving support vector machine algorithm (ppSVM) plus the previously obtained (X, Y) set is used to update the final result and generate an output result F and privacy class Pr_F.

Overall, PPDIFSEA allows the weight model of the DBN to be changed iteratively to predict the degree of independence of the data and obtain the privacy boundary. In the next section, we will test and compare the fusion experiments of the models and algorithms proposed in this paper and the comparison of related methods.

Algorithm 2 Privacy-preserving distribution prediction and independent frequent subsequence extraction algorithm (PPDIFSEA)

```
INPUT:
string set $S = \{s_1, s_2, \cdots, s_N\}$, $s_i = < s_{i,1}, s_{i,2}, \cdots, s_{i,m} >$
support threshold $\xi$
Output:
Independent frequent substring set $F$, privacy classes $Pr_F$

procedure INIT EMPTY DAG G
    for $b_i$ in $S$,l,j in range($S$,$L$,$\|s_i\| - l$ do
        $h^{(l)} = \exp(b^{l^T} h^{(l)} + b^{l-1} h^{(l-1)} + h^{(l-1)} W^{(l)} h^{(l)})$
        G.ADD_VERTEX_IF_NOT_EXIST (x),(p1, x),(p2, x)
        G.UPDATE_IC_FROM_DBN (x)
    end for
    SET $X$ =G.GET_VERTEX($S_0$), $X = \{x_1, x_2, \cdots, x_K\}$
SORT X BY $\|x_i\|$
    for each $x_i$ IN $X$ do
        if G then.GET_SUPPORT $(x_i) < \xi$
            G.REMOVE_SELECTED_NODE $(x_i)$
            G.REMOVE_VERTEX $(x_i)$
        end if
     end for
     for each $l = L$ to 1 and $x_i$ IN $X$ ON $\|x_i\| = l$ do
        SET $Y$ =G.UPDATE_IC_FROM_DBN $(x_i)$
        PPDIFSEA $(x_i) = \mathrm{D}_{\mathrm{ppSVM}}(x_i) + \sum_{x_j \in Y} PPDIFSEA(x_j)$
        if PPDIFSEA $(x_i) \geq \xi$ then
            $F.ADD(x_i)$
        end if
     end for
 return F,$Pr_F$
 end procedure
```

4.3 Stochastic gradient descent(SGD) approach for refining PPDIFSEA

The privacy boundary $Pb(b) = e^{-\frac{\omega}{b}}$ follows the symmetrical and exponential distribution, the standard deviation is $\sqrt{2}b$, under the condition of $b = \Delta\frac{f}{\theta}$. Thus, the probability density function of $p(x)$ is

$$p(x) = \frac{\theta}{2\Delta(f)} e^{-\omega\frac{\theta}{\Delta(f)}}. \tag{13}$$

The problem (13) can refine by the optimized function Stochastic Gradient Descent(SGD). The SGD has three steps:

1. **Step 1**: Label the input data and push those data into vectors, which can be calculated by using (10).
2. **Step 2**: Stochastic Gradient Descent (SGD) impact every stage. The model must compute gradient direction from random subsets and update the parameters systematically, then the gradient is estimated as $\vartheta_{\Theta_t}\mathcal{L}(\Theta_t, d_n)$.
   After that, we normalize the activation function during each iteration and compute the average value. The Gaussian noises are generated by using (13). The Gaussian noises can be added to the training sets as the differential process. Furthermore, in order to avoid the disclosure of confidential information, we only included the training methods and parameters during the training period to protect the training data if the data contains privacy.
3. **Step 3**: The third step is to submit the data into the cloud according to two requirements. The first requirement is that the framework should select the correct Θ_{t+1}, which is the maximum value of the SGD algorithm. The other requirement is that the framework should select the sample D_{n+1} and the sample value is less than the limitation of privacy budget G. The second choice may lead to lessening the convergence time. However, both conditions can be improved by using the kernel function with a positive kernel.

According to the definition of frequently-independent sub-strings, we need to find frequent sub-strings and their support (independence), then the PPDIFSEA obtain the inclusion relationship between frequent sub-strings as Algorithm 1 shows. Since the vectors have added noises (differential privacy approach) during SGD steps if the third parties want to obtain privacy via statistical attack, they need to find out the distance between noise data and processed data. In real conditions, the possibility of obtaining raw privacy information is little and the time cost is high [11] due to the project process being irreversible.

5 Experimental results

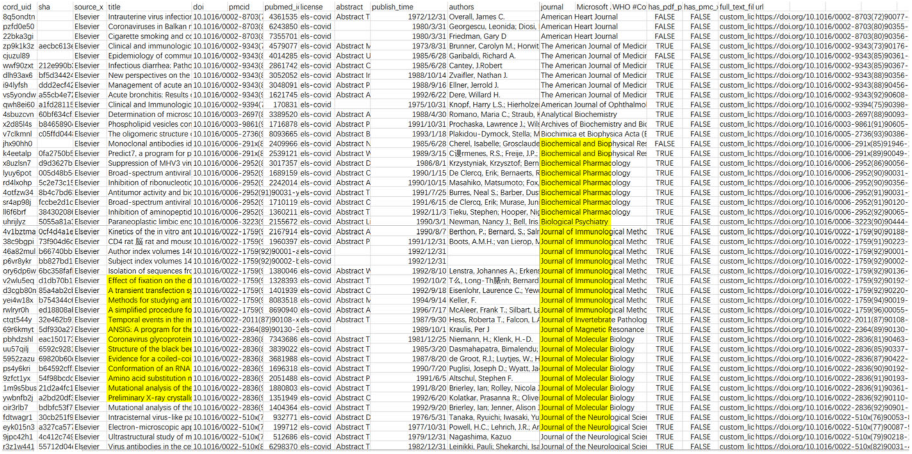
5.1 Implementation of data sets

In the actual data processing process, there may be several types in the text, such as the identity of the text subject, the author of the text or the whole text, keywords, or annotation labels. In the method in this paper, the sensitive class has been defined by the user. The method in this paper can associate the sensitive attribute word space after training to protect the word vectors in the sensitive category. Specifically, this paper tests the data set of COVID-19 [48] to explain the utility of the proposed model. From the aspect of accuracy rate with classification, we have applied the entire COVID-19 data-set [48] as the training data for WECPPSVM.

Since the accuracy of the existing frameworks of medical word segmentation [51] (e.g., the MD word segmentation) and the word segmentation are highly correlated, we have screened the Medical Transcription Corpus((MTC) [16] abbreviated medicine vocabulary entries and found new words in the text, which have resulted in a more complete user word list. Then, in the word embedding process we segment the medical word and added it to the user vocabulary to process the COVID-19 data-set [48]). During the word embedding process for Medical Transcription Corpus [16], the process is able to generate the vector corresponding to each word in the data sets. In detail, all vocabulary vectors are lodged into memory firstly, and the text is tokenized, normalized, and lemmatized. Then, the vocabulary is weighted and averaged to obtain the word vector.

The identity of the text topic, the author of the text, or the existence of the entire text (in the training set) are the key directions of the word embedding text classification studied by the algorithm in this paper. In order to better illustrate the relationship between classification tasks and privacy protection, we use the Classifier matrix for COVID-19 Data-sets to illustrate this relationship. In Fig. 4, we have analyzed the classifier matrix of the training set in COVID-19 [48] data-sets as a case. In this data-set, the data include passage titles, contents, authors, etc, Our work will extract the sensitive parts from the contents part and classify the useful information inside of content parts connect with other features, authors, or titles for example. The light-yellow cells (diagonal elements) in Fig. 4 show the counts of classified documents with passages content and keywords, etc. Then, it means, that under the current classification framework, the classifier can not effectively identify the boundaries of these classes, and these classes themselves coincide with others.

Due to the inconsistency of classification logic and the inherent diversity of content, and the extremely complex text structure, the dimension formed by the direct conversion of documents is extremely high. If word embedding processing is not performed, the classifier cannot obtain the classification consistent with the label data from the original data. In other words, due to the complex structure of training samples, each sample is only recognized as one of its actual categories, which leads to poor performance training of the classification model [2]. So it also shows the importance of word embedding processing.



cord_uid	sha	source_x	title	doi	pmcid	pubmed_i	license	abstract	publish_time	authors	journal	Microsoft .	WHO #Co	has_pdf_p	has_pmc_)	full_text_fil	url
8q5ondtn		Elsevier	Intrauterine virus infectior	10.1016/0002-8703(7		4361535	els-covid	Abstract T	1972/12/31	Overall, James C.	American Heart Journal			FALSE	FALSE	custom_lic	https://doi.org/10.1016/0002-8703(72)90077-
pzfd0e50		Elsevier	Coronaviruses in Balkan r	10.1016/0002-8703(8		6243850	els-covid		1980/3/31	Georgescu, Leonida; Diosi,	American Heart Journal			FALSE	FALSE	custom_lic	https://doi.org/10.1016/0002-8703(80)90355-
22bka3gi		Elsevier	Cigarette smoking and cc	10.1016/0002-8703(8		7355701	els-covid		1980/3/31	Friedman, Gary D	American Heart Journal			FALSE	FALSE	custom_lic	https://doi.org/10.1016/0002-8703(80)90356-
zp9k1k3z	aecbc613e	Elsevier	Clinical and immunologic	10.1016/0002-9343(7		4579077	els-covid	Abstract M	1973/8/31	Brunner, Carolyn M.; Horwit	The American Journal of Medicir			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0002-9343(73)90176-
cjuzul89		Elsevier	Epidemiology of commur	10.1016/0002-9343(8		4014285	els-covid	Abstract U	1985/6/28	Garibaldi, Richard A.	The American Journal of Medicir			FALSE	FALSE	custom_lic	https://doi.org/10.1016/0002-9343(85)90361-
wwf90zxt	212e990b:	Elsevier	Infectious diarrhea: Pathc	10.1016/0002-9343(8		2861742	els-covid	Abstract C	1985/6/28	Cantey, J.Robert	The American Journal of Medicir			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0002-9343(85)90367-
dlh93ax6	bf5d3442	Elsevier	New perspectives on the	10.1016/0002-9343(8		3052052	els-covid	Abstract Ir	1988/10/14	Zvaifler, Nathan J.	The American Journal of Medicir			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0002-9343(88)90356-
i94lyfsh	ddd2ecf42	Elsevier	Management of acute an	10.1016/0002-9343(8		3048091	els-covid	Abstract P	1988/9/16	Ellner, Jerrold J.	The American Journal of Medicir			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0002-9343(88)90456-
vs5yondw	a55cb4e7:	Elsevier	Acute bronchitis: Results	10.1016/0002-9343(9		1621745	els-covid	Abstract A	1992/6/22	Dere, Willard H.	The American Journal of Medicir			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0002-9343(92)90608-
qwh8ei60	a1fd2811	Elsevier	Clinical and Immunologic	10.1016/0002-9394(7		170831	els-covid		1975/10/31	Knopf, Harry L.S.; Hierholze	American Journal of Ophthalmol			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0002-9394(75)90398-
4sbuzcvn	60bf634cf	Elsevier	Determination of microsc	10.1016/0003-2697(8		3389520	els-covid	Abstract A	1988/4/30	Romano, Maria C.; Straub, I	Analytical Biochemistry			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0003-2697(88)90093-
x2d85l4s	b8465890	Elsevier	Phospholipid vesicles con	10.1016/0003-9861(9		1716878	els-covid	Abstract P	1991/10/31	Prochaska, Lawrence J.; Wils	Archives of Biochemistry and Bic			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0003-9861(91)90605-
v7clkmnl	c05ffd044	Elsevier	The oligomeric structure	10.1016/0005-2736(9		8093665	els-covid	Abstract B	1993/1/18	Plakidou-Dymock, Stella; M	Biochimica et Biophysica Acta (E			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0005-2736(93)90386-
jhx90hh0		Elsevier	Monoclonal antibodies id	10.1016/0006-291x(8		2409966	els-covid	Abstract N	1985/6/28	Cherel, Isabelle; Grosclaude	Biochemical and Biophysical Res			FALSE	FALSE	custom_lic	https://doi.org/10.1016/0006-291x(85)91946-
k4eetalp	0fa2750b5	Elsevier	Predict7, a program for p	10.1016/0006-291x(8		2539121	els-covid	Abstract W	1989/3/15	C谩rmenes, R.S.; Freije, J.P.;	Biochemical and Biophysical Res			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0006-291x(89)90049-
x8uzlsn7	d9d3627b	Elsevier	Suppression of MHV3 vir	10.1016/0006-2952(8		3017357	els-covid	Abstract D	1986/8/1	Krzystyniak, Krzysztof; Bern	Biochemical Pharmacology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0006-2952(86)90056-
lyuy6pot	005d48b5	Elsevier	Broad-spectrum antiviral	10.1016/0006-2952(9		1689159	els-covid	Abstract C	1990/1/15	De Clercq, Erik; Bernaerts, R	Biochemical Pharmacology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0006-2952(90)90031-
rd4lxohp	5c2e73c1	Elsevier	Inhibition of ribonucleotic	10.1016/0006-2952(9		2242014	els-covid	Abstract A	1990/10/15	Masahiko, Matsumoto; Fox,	Biochemical Pharmacology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0006-2952(90)90356-
4otfzw34	8b4c7bd6:	Elsevier	Antitumor activity and bic	10.1016/0006-2952(91)90031-y			els-covid	Abstract T	1991/7/25	Burres, Neal S.; Barber, Dus	Biochemical Pharmacology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0006-2952(91)90031-
sr4ap98j	fccbe2d1c	Elsevier	Broad-spectrum antiviral	10.1016/0006-2952(9		1710119	els-covid	Abstract C	1991/6/15	de Clercq, Erik; Murase, Juni	Biochemical Pharmacology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0006-2952(91)90120-
ll6f6brf	38430208	Elsevier	Inhibition of aminopeptid	10.1016/0006-2952(9		1360211	els-covid	Abstract T	1992/11/3	Tieku, Stephen; Hooper, Ni	Biochemical Pharmacology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0006-2952(92)90065-
uhrijlyz	5055a81a:	Elsevier	Paraneoplastic limbic enc	10.1016/0006-3223(9		2155672	els-covid	Abstract Li	1990/3/1	Newman, Nancy J.; Bell, Iris	Biological Psychiatry			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0006-3223(90)90444-
4v1bztma	0cf4d4a1e	Elsevier	Kinetics of the in vitro ant	10.1016/0022-1759(9		2167914	els-covid	Abstract A	1990/8/7	Berthon, P.; Bernard, S.; Salr	Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(90)90188-
38c9bgpi	73f904d6c	Elsevier	CD4 rat 脳 rat and mouse	10.1016/0022-1759(9		1960397	els-covid	Abstract P	1991/12/31	Boots, A.M.H.; van Lierop, M	Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(91)90223-
46a82mul	b66740bb	Elsevier	Author index volumes 14	10.1016/0022-1759(92)90001-a			els-covid		1992/12/31		Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(92)90001-
p6vr8ykr	bb827bd1	Elsevier	Subject index volumes 14	10.1016/0022-1759(92)90002-b			els-covid		1992/12/31		Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(92)90002-
ory6dp6w	6bc358faf	Elsevier	Isolation of sequences frc	10.1016/0022-1759(9		1380046	els-covid	Abstract W	1992/8/10	Lenstra, Johannes A.; Erkens	Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(92)90136-
v2wlu5eq	d1db70b1	Elsevier	Effect of fixation on the d	10.1016/0022-1759(9		1328393	els-covid	Abstract T	1992/10/2	T么, Long-Th膝nh; Bernard	Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(92)90192-
d3cgb80n	85a4ab2c	Elsevier	A transient transfection sy	10.1016/0022-1759(9		1401939	els-covid	Abstract C	1992/9/18	Eisenlohr, Laurence C.; Yew	Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(92)90220-
yei4w18x	b754344c	Elsevier	Methods for studying ant	10.1016/0022-1759(9		8083518	els-covid	Abstract M	1994/9/14	Keller, F.	Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(94)90019-
rwlryr0h	ed18808al	Elsevier	A simplified procedure fo	10.1016/0022-1759(9		8690940	els-covid	Abstract A	1996/7/17	McAleer, Frank T.; Silbart, La	Journal of Immunological Methc			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-1759(96)00055-
ctqt544y	32e462b9	Elsevier	Temporal events in the in	10.1016/0022-2011(87)90108-x			els-covid	Abstract T	1987/9/30	Hess, Roberta T.; Falcon, L.A	Journal of Invertebrate Patholog			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2011(87)90108-
69r6kmyt	5df930a27	Elsevier	ANSIG: A program for the	10.1016/0022-2364(89)90130-3			els-covid		1989/10/1	Kraulis, Per J	Journal of Magnetic Resonance			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2364(89)90130-
pbhdzshl	eac15017	Elsevier	Coronavirus glycoprotein	10.1016/0022-2836(8		7343686	els-covid	Abstract T	1981/12/25	Niemann, H.; Klenk, H.-D.	Journal of Molecular Biology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2836(81)90463-
uu57qilj	6592c928	Elsevier	Structure of the black bee	10.1016/0022-2836(8		3839022	els-covid	Abstract T	1985/3/20	Dasmahapatra, Bimalendu;	Journal of Molecular Biology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2836(85)90337-
5952zazu	69820b60	Elsevier	Evidence for a coiled-coi	10.1016/0022-2836(8		3681988	els-covid	Abstract T	1987/8/20	de Groot, R.J.; Luytjes, W.; H	Journal of Molecular Biology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2836(87)90422-
ps4y6kri	b64592cff	Elsevier	Conformation of an RNA	10.1016/0022-2836(9		1696318	els-covid	Abstract T	1990/7/20	Puglisi, Joseph D.; Wyatt, Ja	Journal of Molecular Biology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2836(90)90192-
9zfct1yx	54f98bcdc	Elsevier	Amino acid substitution n	10.1016/0022-2836(9		2051488	els-covid	Abstract P	1991/6/5	Altschul, Stephen F.	Journal of Molecular Biology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2836(91)90193-
1m9s5bus	21d2a4fc1	Elsevier	Mutational analysis of the	10.1016/0022-2836(9		1880803	els-covid	Abstract T	1991/8/20	Brierley, Ian; Rolley, Nicola J	Journal of Molecular Biology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2836(91)90361-
ywbnfb2j	a2bd20df	Elsevier	Preliminary X-ray crystall	10.1016/0022-2836(9		1351949	els-covid	Abstract C	1992/6/20	Kolatkar, Prasanna R.; Olive	Journal of Molecular Biology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2836(92)90110-
oir3rlb7	bdbfc53f7	Elsevier	Mutational analysis of the	10.1016/0022-2836(9		1404364	els-covid	Abstract T	1992/9/20	Brierley, Ian; Jenner, Alison J	Journal of Molecular Biology			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-2836(92)90901-
fdtwagr1	30cb251f9	Elsevier	Intracisternal virus-like pa	10.1016/0022-510x(7		932771	els-covid	Abstract D	1976/5/31	Tanaka, Ryuichi; Iwasaki, Yu	Journal of the Neurological Scier			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-510x(76)90053-
eyk015n3	a327ca57	Elsevier	Electron-microscopic apı	10.1016/0022-510x(7		199712	els-covid	Abstract T	1977/10/31	Powell, H.C.; Lehrich, J.R.; Ar	Journal of the Neurological Scier			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-510x(77)90087-
9jpc42h1	4c412c74	Elsevier	Ultrastructural study of m	10.1016/0022-510x(7		512686	els-covid	Abstract T	1979/12/31	Nagashima, Kazuo	Journal of the Neurological Scier			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-510x(79)90218-
r3z1w441	55712d04	Elsevier	Virus antibodies in the ce	10.1016/0022-510x(8		6298370	els-covid	Abstract T	1982/12/31	Leinikki, Pauli; Shekarchi, Isa	Journal of the Neurological Scier			TRUE	FALSE	custom_lic	https://doi.org/10.1016/0022-510x(82)90031-

Fig. 4 The text matrix with annotation labels for COVID-19 Data-sets [48]

5.2 Ablation experiments

From the ablation study aspects, in order to accurately evaluate the performance of the algorithm in this paper, the evaluation matrix we introduced not only includes the accuracy rate but also includes Root Mean Squared Error with different levels of privacy budget. Moreover, at the beginning of the paper, we introduced the privacy threat model. In order to verify the privacy threat model, we utilize the similarity of data-sets to mimic privacy attacks in Fig. 10. The core of this test is how similar the attacker's composition is to the existing data. We use the KL divergence method to compare, assuming the attack If the person already knows the data source, how similar is the data source to the existing protected data? Generally, the higher the similarity, the higher the KL value, and the worse the protection of the privacy protection algorithm.

In Fig. 5, the Root Mean Squared Error(rMSE) shows the comparison between the Word Embedding Combination Privacy-preserving Support Vector Machine(WECPPSVM) under the COVID 19 [48] datasets with and without using PPDIFSEA, the x-axis is the privacy budget from L1 to L50, L1 means less privacy budget and L50 means higher privacy budget, the higher privacy budget means the method need to predict more privacy boundary for independence degree calculation. And the y-axis is the training loss with Root Mean Squared Error. The higher of value the higher of error for the prediction of the privacy budget. The Radial Basis Function(RBF) and Sigmoid functions are the kernel function for the Support Vector Machine. When the predicted amount of privacy budget is less than L16, the rMSE of both kernel functions reduces significantly, but the error rates with and without PPDIFSEA under RBF functions are almost the same. It also can be seen that the WECPPSVM with PPDIFSEA with RBF is lower than the rest of the models when the amount of privacy budget is around L16 to L50. The PPDIFSEA algorithm shows its advantages. Compared with the model without the PPDIFSEA algorithm, whether the classification algorithm uses the Sigmoid or the RBF kernel function when the privacy noise is larger, the error rate using the PPDIFSEA algorithm can be controlled at a lower level. When the privacy budget is set to L50, the difference in the rMSE value can be more than 8.

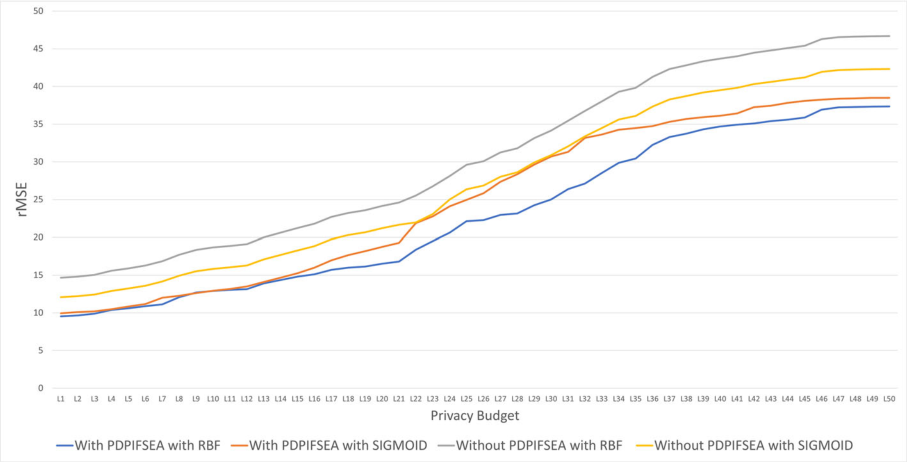


Fig. 5 Root Mean Squared Error(rMSE) of training data-sets using the WECPPSVM with and without PPDIFSEA under different privacy budget)

The Fig. 6 is the ablation study for proposed PPDIFSEA under the COVID 19 [48] datasets in different Optimize functions(SGD and AdamW separately) and with different activation functions(Leaky Relu and Relu). It can be seen from the Fig. 6 that when the training period of the deep belief network in PPDIFSEA increases, the accuracy of predicting the degree of independence also increases accordingly. For the same optimization function SGD, no matter which activation function is used, the accuracy of all cycles is higher than the optimization function of AdamW, especially after the training cycle is increased to more than 400 epochs, the accuracy advantage of using SGD in PPDIFSEA is even greater. Obviously, when the iteration period is 540 epochs, regardless of whether Relu or Leaky Relu activation function is used, the accuracy of the algorithm predicting the degree of independence can reach 87%. Using the AdamW optimization function, the accuracy of using Leaky Relu is about 4% higher than that of using Relu, reaching 68%. From a period of 50, using Leaky Relu with SGD is 8% higher than using Relu with AdamW function to 550 epochs, the difference reaches 24 percentage points. This shows that in the prediction of the privacy noise boundary, the SGD optimization function is more obvious than AdamW, and the Leaky Relu activation function has more obvious advantages than the Relu function.

This test is the acceptance probability of PPDIFSEA under different sampling rates. Here we use the chi-square test to determine the acceptance probability, that is, to determine whether the features obtained by sampling and predicting the data before and after reconstruction through Deep Belief Network(DBN) are statistically significant The acceptance probability is 1.0, indicating that the similarity is 100%, and the lower the acceptance probability, the more obvious the difference between the two, and 0 means that the two data are completely different in statistical characteristics. For a better comparison here, we use two datasets, COVID-19 [48] datasets and Medical Transcription Corpus(MTC) [16] datasets, where the sampling rate is the rate at which the dataset is sampled within itself, for example,

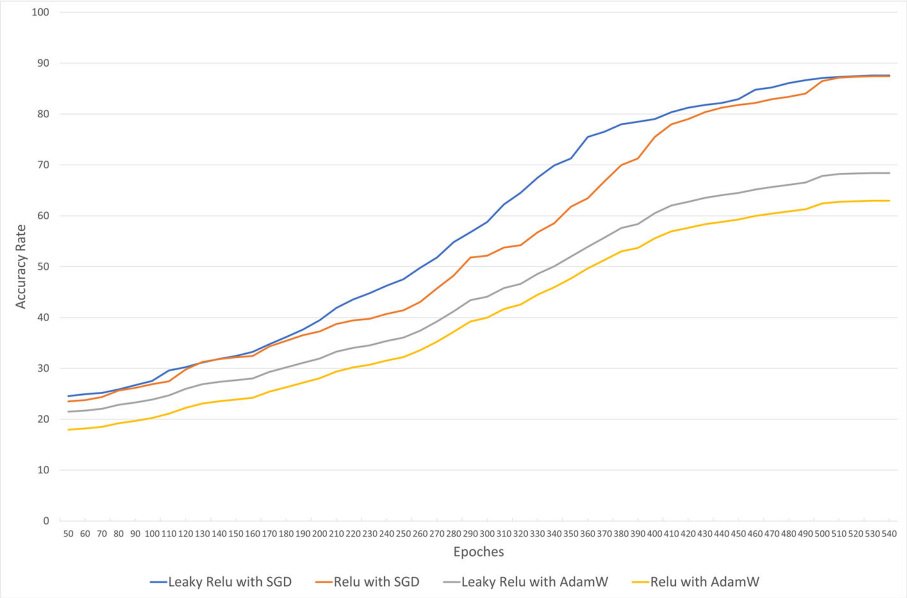


Fig. 6 Classification performance for Independence Degree with Deep Belief Network process

a sampling rate of 0.1 means removing 10% of the original data, and retain the remaining 90% of the data as the range of DBN sampling data.

As shown in Fig. 7, as the test sampling rate increases, its acceptance probability decreases accordingly, and the degree of reduction of the classification acceptance probability on different data sets is also different. For the trends in this figure, the COVID-19 corpus has a better classification acceptance rate than the medical vocabulary corpus. Due to the redundant or repeated data in the COVID-19 corpus, after removing a part of the MTC data, the statistical difference increases significantly. When 90% of the data is removed, in terms of statistical characteristics, the reconstructed data can hardly get any similar characteristics. The privacy noise it constitutes is also difficult to interfere with the background-knowledge-based privacy attack. Finally, judging from the results of these two typical corpora (COVID-19 and medical vocabulary words), the data set MTC with extremely low repetition and a large number of isolated words, the more sampling data PPDIFSEA needs. And the sample quantity cannot be reduced for MTC datasets in the sampling stage, because the characteristics of the text in MTC are obviously different. After being attacked by differential privacy, the possibility of guessing private information is greater. The weaknesses against the statistical characteristics of the data will be left to our future work to address (Fig. 8).

5.3 Performance of WECPPSVM and PPDIFSEA

In Fig. 9, the X-axis means WECPPSVM uses different kernel functions, the Y-axis is the classification accuracy of different data under different kernel functions, and the sampling rate ν is 0.1 means from the training labeled data set randomly remove out 10% of the labeled text for classification. When the training set classification accuracy of the COVID-19 corpus is 0.90, the gap between the validation set accuracy and the training set as corresponding samples is less than 0.02. And The Medical Transcription Corpus(MTC) data set is the normalization verification set, the accuracy for MTC has also reached between 0.50 and 0.60. Because firstly, COVID-19 data-sets and Medical Transcription Corpus contents are different. Secondly, the former (Covid-19) contains the latter (MTC) data. Thirdly, the WECPPSVM model is trained for the former data. The accuracy rate of 0.5 for MTC also shows that the WECPPSVM model has a relatively strong normalization ability. In the same data-set, compared to several

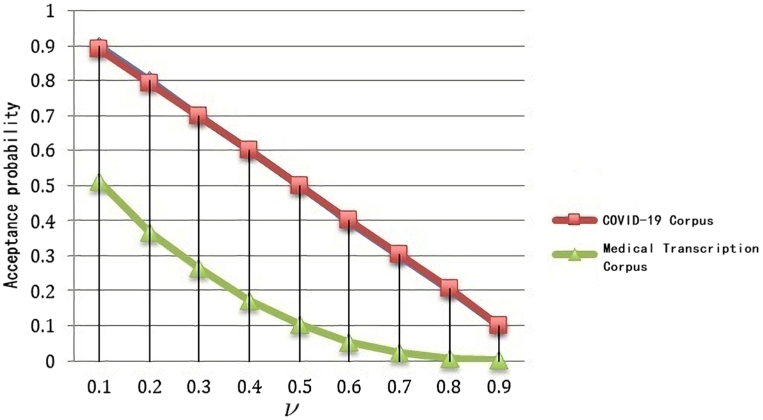


Fig. 7 Acceptance probability for PPDIFSEA under different sample rate ν with Chi-square test

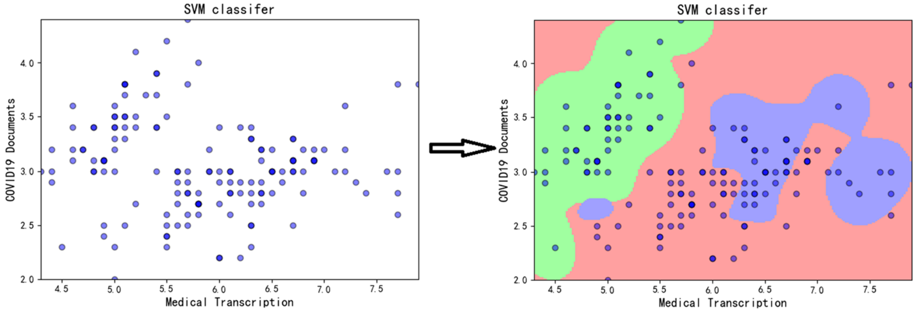


Fig. 8 Classification process of Word Embedding Combination Privacy-preserving Support Vector Machine (WECPPSVM)

functions, the radial basis function (RBF) has higher classification accuracy than the other three, and the other three accuracy values are not much different. Therefore, the scheme in this paper also uses Radial Basis Function (RBF) as the kernel function of our classifier.

In Table 1, the classification accuracy of three text classification models at different sampling rates is shown, and the last item is the maximum difference in accuracy among models. Because the results of several classifications can be very poor when the training data set is deleted in large numbers, only sampling rates ν between 0.01 and 0.09 are used for this comparative experiment. It means the removed data has not exceeded 10% of the total data in the test.

Among them, we compared the text classification accuracy of the WECPPSVM model with and without the PPDIFSEA algorithm proposed in this paper. It can be seen that the maximum gap value between WECPPSVM model including PPDIFSEA and the WECPPSVM model without PPDIFSEA can reach 0.056. This indicates that the deep belief network (DBN) of PPDIFSEA can help the previous classification model to achieve better classification accuracy compared with the WECPPSVM which privacy noise generated by random sampling(WECPPSVM model without PPDIFSEA). And for BERT which does not use any privacy protection method, the gap between the classification accuracy of the model proposed in this paper is also very low. When the sampling rate is 0.01-0.09, although the two

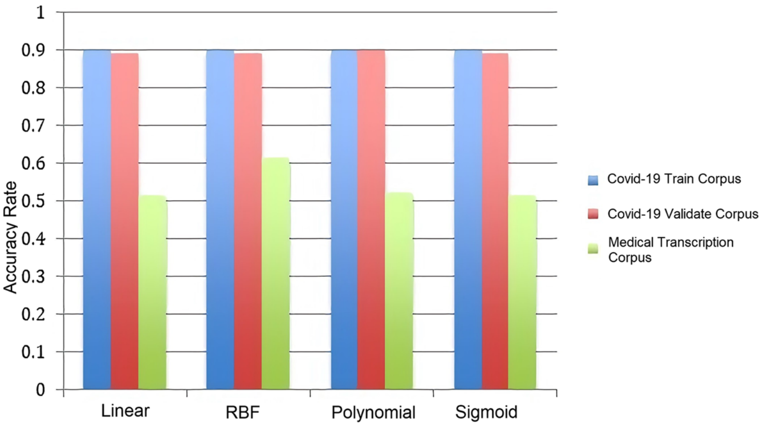


Fig. 9 Comparison of accuracy for WECPPSVM with different kernel functions

Table 1 The comparison of accuracy rate of the algorithms PPDIFSEA, BERT [14] (without privacy protection) and WECPPSVM (without PPDIFSEA) based on MIMIC [21] Clinical Data Sets

v	PPDIFSEA+WECPPSVM	BERT	only WECPPSVM	Maximum gap
0.01	0.8495	0.8932	0.8135	0.0797
0.02	0.8312	0.8715	0.7852	0.0863
0.03	0.8117	0.8514	0.7623	0.0891
0.04	0.7962	0.8385	0.7735	0.065
0.05	0.7716	0.8102	0.7437	0.0665
0.06	0.7595	0.7824	0.7161	0.0663
0.07	0.7378	0.7622	0.6951	0.0671
0.08	0.7002	0.7327	0.6566	0.0761
0.09	0.6718	0.7145	0.6312	0.0833

models always lag behind BERT, the maximum difference in accuracy rate value is always no excess than 0.1. But both WECPPSVM models(with and without PPDIFSEA) preserve privacy, while BERT does not. It can be shown that the WECPPSVM model(with PPDIFSEA) proposed in this paper can balance the strength of privacy protection and the accuracy of text classification.

5.4 Empirical privacy threat evaluation

Moreover, at the beginning of the paper, we introduced the privacy threat. Privacy attacks can be guessed by finding or constructing samples from a set of data sets B similar to the target data set A, that is, the attacker uses background knowledge attacks to query the target data set A to obtain himself What content in the known data set B is similar to the target data set A. Through the query, the attacker can obtain the privacy or sensitive data similar to or the same as A in the data set he knows, so as to obtain the privacy of A.

In order to verify the proposed solution for privacy threats, we utilize the similarity of datasets to generate a privacy test. As shown in Fig. 10, The figure is a privacy attack test under the background knowledge attack. The core of this test is how similar the attacker's composition is to the existing data. We use the KL divergence method to compare, assuming the attack If the person already knows the data source, how similar is the data source to the existing protected data? Generally, the higher the similarity, the higher the KL value, and the worse the protection of the privacy protection algorithm.

The datasets for this test, Wiki-meta [6], WordNet [13], Ca-hepTh [33],Ca-GrQc [9] are pairwise similar data sets, of which WordNet The data set in comes from part of the data set in Wiki-meta, and Ca-GrQc [33] is a collaboration network of Arxiv General Relativity category. Ca-Hepth [26] is a collaboration network of Arxiv High Energy Physics Theory category from 1993 to 2003. Ca-Hepth and Ca-GrQc have very similar word pieces, and Wiki-meta, WordNet has very similar contents. Among them, the X axis represents the KL divergence. This value is obtained by comparing another similar data set in the figure, such as the figure The X-axis in the data set titled Wiki-meta, Musae-twitch, Ca-GrQc, and Ca-Hepth represent the KL divergence compared to the data set sampled from the data set Wiki-meta, WordNet, Ca-GrQc, and Ca-Hepth separately. The Y axis is the corresponding value of ε. Among them, ε is the ε in ε-differential privacy. If the result of classifying one kind of data

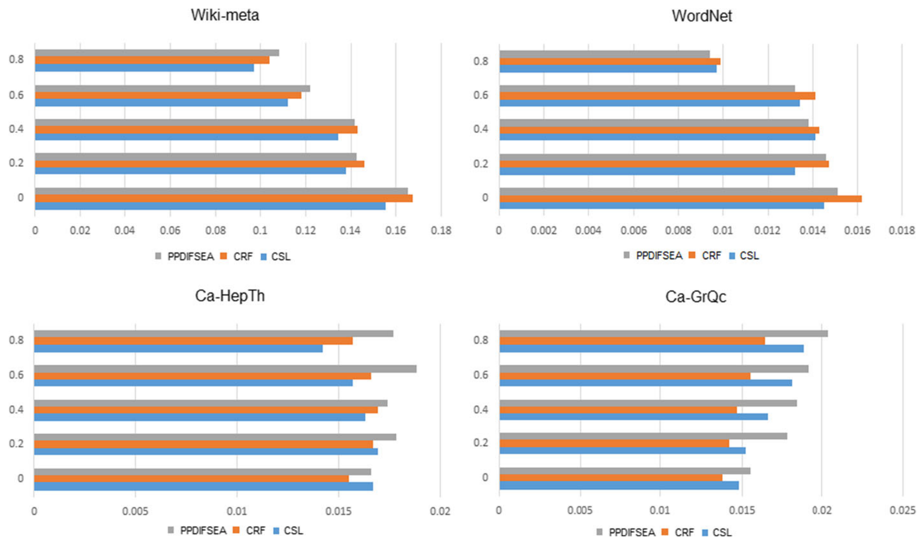


Fig. 10 knowledge-background-based privacy attack test under pairwise similar data sets

expresses the less similar the other result, that is, the higher the KL divergence value, the harder it is to guess the privacy through background knowledge attack.

It can be seen from the figure that PPDIFSEA has a relatively high KL divergence value among the different ε values of the four data sets, that is, the probability that the attacker obtains similar information through the background knowledge attack becomes lower. The other two methods, one is the Constrain Laplace Noise Methods(CSL) [22], and the other is so-called CRF(Conditional Random Field) [41] method. Although they both deal with the target data set and obfuscate, these two methods can allow the attacker to obtain sensitive information easier. It becomes more difficult to obtain private information via background knowledge with higher KL values. Through observation, it can be found that the KL values of these two methods on different types of data sets are random. For example, on the Wiki-Meta and Musae data sets, CRF gives less similar distributions on such data sets (the KL divergence means is higher than other methods). However, the CRF method has a lower mean KL divergence value on ca-HepTh and ca-GrQc data sets. Through the above experiments, it can be verified that the PPDIFSEA method in this paper is better than the above two methods in preventing privacy leakage based on background knowledge attacks.

6 Conclusion

In this paper, we propose Word Embedding Combination Privacy-preserving Support Vector Machines (WECPPSVM) to preserve text classification results. We have empirically evaluated and validated the algorithm on real datasets. Our proposed WECPPSVM allows pattern classification with high accuracy. We predict the privacy boundary and generate privacy noise by using the Privacy-preserving Distribution and Independent Frequent Sub-sequence Extraction Algorithm(PPDIFSEA) method of deep belief networks. In the privacy verification experiment, we also show that the method proposed in this paper can prevent privacy attacks based on background knowledge, thereby protecting privacy. Since this work is based on publicly available text embedding models, this work reveals a new direction for privacy-preserving text classification. In our future work, we will deeply explore the performance of our method in more textual scenarios. Especially for some data-sets with very independent statistical features and many rare words, the classification model of this paper is continuously optimized.

Data Availability No datasets were generated in this study. The datasets analyzed during the current study are available in the Kaggle, MTsamples, SNAP and WordNet repositories. These data-sets were derived from the following public domain resources:
https://www.kaggle.com/datasets/allen-institute-for-ai/CORD-19-research-challengeCOVID19 Data-sets
https://www.mtsamples.com/site/pages/browse.asp?type=6-Cardiovascular%20/%20PulmonaryMedical Transcription Corpus
https://snap.stanford.edu/data/ca-GrQc.htmlCa-GrQc Datasets
https://snap.stanford.edu/data/ca-HepTh.htmlCa-HepTh Datasets
https://snap.stanford.edu/data/wiki-meta.htmlWiki-meta Datasets
https://wordnet.princeton.edu/downloadWordNet Datasets

Declarations

Funding and/or Conflicts of interests/Competing interests Open Access funding enabled and organized by CAUL and its Member Institutions. The authors did not receive support from any organization for the submitted work. The authors have no financial or non-financial or proprietary interests in any material discussed in this article.

References

1. Abdalla M, Abdalla M, Hirst G, Rudzicz F (2020) Exploring the privacy-preserving properties of word embeddings: algorithmic validation study. Journal of medical Internet research 22(7):18055
2. Abdalla M, Abdalla M, Hirst G, Rudzicz F (2020) Exploring the privacy-preserving properties of word embeddings: algorithmic validation study. Journal of medical Internet research 22(7), 18055
3. Abe N, Kudo M, Toyama J, Shimbo M (2006) Classifier-independent feature selection on the basis of divergence criterion. Pattern analysis and applications 9(2–3):127–137
4. Abe N, Kudo M, Toyama J, Shimbo M (2006) Classifier-independent feature selection on the basis of divergence criterion. Pattern analysis and applications 9(2-3), 127–137
5. Ambrosio L, Miranda M Jr, Maniglia S, Pallara D (2010) Bv functions in abstract wiener spaces. Journal of Functional Analysis 258(3):785–813
6. Bartunov S, Kondrashkin D, Osokin A, Vetrov D (2016) Breaking sticks and ambiguities with adaptive skip-gram. In: Artificial Intelligence and Statistics, pp. 130–138
7. Chang YK, Zhao ZH (2011) N'Guérékata, GM (2011) Square-mean almost automorphic mild solutions to some stochastic differential equations in a hilbert space. Advances in Difference Equations 1:1–12
8. Chowdhury GG (2003) Natural language processing. Annual Review of Information Science and Technology 37(1):51–89
9. Church KW (2017) Word2vec. Natural Language Engineering 23(1), 155–162
10. Church KW (2017) Word2vec. Natural Language Engineering 23(1):155–162
11. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805

12. Dwork C, Roth A et al (2014) The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science 9(3–4):211–407
13. Fellbaum C (2010) Wordnet. Computer Applications, Theory and Applications of Ontology, pp 231–243
14. Fernandes N, Dras M, McIver A (2019) Generalised differential privacy for text document processing. In: International Conference on Principles of Security and Trust, pp. 123–148
15. Geng C, Huang Sj, Chen S (2020) Recent advances in open set recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
16. Ghosh S, Dasgupta A, Swetapadma A (2019) A study on support vector machine based linear and non-linear pattern classification. In: 2019 International Conference on Intelligent Sustainable Systems (ICISS), pp. 24–28 IEEE
17. Gupta D, Kose U, Le Nguyen B, Bhattacharyya S (2021) Artificial intelligence for data-driven medical diagnosis. Walter de Gruyter GmbH & Co KG
18. Hirsch C, Hosking J, Grundy J (2010) Vikibuilder: end-user specification and generation of visual wikis. In: Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, pp. 13–22
19. Huang CR, Ahrens K (2003) Individuals, kinds and events: classifier coercion of nouns. Language Sciences 25(4), 353–373
20. Huang CR, Ahrens K (2003) Individuals, kinds and events: classifier coercion of nouns. Language Sciences 25(4):353–373
21. Johnson AE, Pollard TJ, Shen L, Lehman LWH, Feng M, Ghassemi M, Moody B, Szolovits P, Anthony Celi L, Mark RG (2016) MIMIC-III: A freely accessible critical care database. Scientific Data 3(1):1–9
22. Kaggle COVID-19 Open Research Dataset Challenge from www.kaggle.com (2020). https://www.kaggle.com/datasets/allen-institute-for-ai/CORD-19-research-challenge
23. Kurnia R, Tangkuman Y, Girsang A (2020) Classification of user comment using word2vec and svm classifier. Int. J. Adv. Trends. Comput. Sci. Eng. 9(1):643–648
24. Lai S, Liu K, He S, Zhao J (2016) How to generate a good word embedding. IEEE Intelligent Systems 31(6):5–14
25. Lee H, Grosse R, Ranganath R, Ng AY (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616
26. Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: Densification and shrinking diameters. ACM transactions on Knowledge Discovery from Data (TKDD) 1(1):2
27. Liao X, Yu Y, Li B, Li Z, Qin Z (2019) A new payload partition strategy in color image steganography. IEEE Transactions on Circuits and Systems for Video Technology 30(3):685–696
28. Liao X, Li K, Zhu X, Liu KR (2020) Robust detection of image operator chain with two-stream convolutional neural network. IEEE Journal of Selected Topics in Signal Processing 14(5):955–968
29. Liao X, Li K, Zhu X, Liu KR (2020) Robust detection of image operator chain with two-stream convolutional neural network. IEEE Journal of Selected Topics in Signal Processing 14(5), 955–968
30. Liao X, Yins J, Chen M, Qin Z (2020) Adaptive payload distribution in multiple images steganography based on image texture features. IEEE Transactions on Dependable and Secure Computing
31. Liao X, Yu Y, Li B, Li Z, Qin Z (2019) A new payload partition strategy in color image steganography. IEEE Transactions on Circuits and Systems for Video Technology 30(3), 685–696
32. Li W, Han J, Pei J (2001) Cmar: Accurate and efficient classification based on multiple class-association rules. In: Proceedings 2001 IEEE International Conference on Data Mining, pp. 369–376 IEEE
33. Liu D, Jing Y, Zhao J, Wang W, Song G (2017) A fast and efficient algorithm for mining top-k nodes in complex networks. Scientific reports 7(1):1–8
34. Melis L, Song C, De Cristofaro E, Shmatikov V (2019) Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 691–706
35. Mironov I (2017) Rényi differential privacy. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF), pp. 263–275 IEEE
36. Mitra V, Wang CJ, Banerjee S (2007) Text classification: A least square support vector machine approach. Applied Soft Computing 7(3):908–914
37. Mnih A, Kavukcuoglu K (2013) Learning word embeddings efficiently with noise-contrastive estimation. Advances in Neural Information Processing Systems(NeurIPS 26:2265–2273
38. Osswald H (2003) Malliavin calculus in abstract wiener space using infinitesimals. Advances in Mathematics 176(1):1–37
39. Rahulamathavan Y, Phan RCW, Veluru S, Cumanan K, Rajarajan M (2013) Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud. IEEE Transactions on Dependable and Secure Computing 11(5):467–479

40. Rahulamathavan Y, Phan RCW, Veluru S, Cumanan K, Rajarajan M (2013) Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud. IEEE Transactions on Dependable and Secure Computing 11(5), 467–479
41. Ramanathan V, Wechsler H (2013) Phishing detection and impersonated entity discovery using conditional random field and latent dirichlet allocation. Computers & Security 34:123–139
42. Ramanathan V, Wechsler H (2013) Phishing detection and impersonated entity discovery using conditional random field and latent dirichlet allocation. Computers & Security 34, 123–139
43. Shen D, Wang e. Guoyin (2018) Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, pp. 440–450 Association for Computational Linguistics(ACL)
44. Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Processing Letters 9(3), 293–300
45. Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Processing Letters 9(3):293–300
46. Thomas A, Adelani DI, Davody A, Mogadala, A, Klakow D (2020) Investigating the impact of pre-trained word embeddings on memorization in neural networks. In: International Conference on Text, Speech, and Dialogue, pp. 273–281 Springer
47. Trèves F (1966) Linear partial differential equations with constant coefficients: existence, approximation, and regularity of solutions. CRC Press
48. Wang, M, Ning ZH, Xiao C, Li T (2018) Sentiment classification based on information geometry and deep belief networks. IEEE Access 6, 35206–35213
49. Wang M, Ning ZH, Xiao C, Li T (2018) Sentiment classification based on information geometry and deep belief networks. IEEE Access 6:35206–35213
50. Wang Q, Xu J, Chen H, He B (2017) Two improved continuous bag-of-word models. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2851–2856 IEEE
51. Yi K, Beheshti J (2009) A hidden markov model-based text classification of medical documents. Journal of Information Science 35(1):67–81
52. Zhang D, Xu H, Su Z, Xu Y (2015) Chinese comments sentiment classification based on word2vec and svmperf. Expert Systems with Applications 42(4):1857–1863

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.