Zhao *et al. Journal of Cloud Computing* (2024) 13:96
https://doi.org/10.1186/s13677-024-00662-4

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

A mobile edge computing-focused transferable sensitive data identification method based on product quantization

Xinjian Zhao[1], Guoquan Yuan[1], Shuhan Qiu[2*], Chenwei Xu[1] and Shanming Wei[2]

Abstract

Sensitive data identification represents the initial and crucial step in safeguarding sensitive information. With the ongoing evolution of the industrial internet, including its interconnectivity across various sectors like the electric power industry, the potential for sensitive data to traverse different domains increases, thereby altering the composition of sensitive data. Consequently, traditional approaches reliant on sensitive vocabularies struggle to adequately address the challenges posed by identifying sensitive data in the era of information abundance. Drawing inspiration from advancements in natural language processing within the realm of deep learning, we propose a transferable **S**ensitive **D**ata **I**dentification method based on **P**roduct **Q**uantization, named **PQ-SDI**. This innovative approach harnesses both the composition and contextual cues within textual data to accurately pinpoint sensitive information within the context of Mobile Edge Computing (MEC). Notably, PQ-SDI exhibits proficiency not only within a singular domain but also demonstrates adaptability to new domains following training on heterogeneous datasets. Moreover, the method autonomously identifies sensitive data throughout the entire process, eliminating the necessity for human upkeep of sensitive vocabularies. Extensive experimentation with the PQ-SDI model across four real-world datasets, resulting in performance improvements ranging from 2% to 5% over the baseline model and achieves an accuracy of up to 94.41%. In cross-domain trials, PQ-SDI achieved comparable accuracy to training and identification within the same domain. Furthermore, our experiments showcased the product quantization technique significantly reduces the parameter size by tens of times for the subsequent sensitive data identification phase, particularly beneficial for resource-constrained environments characteristic of MEC scenarios. This inherent advantage not only bolsters sensitive data protection but also mitigates the risk of data leakage during transmission, thus enhancing overall security measures in MEC environments.

Keywords Sensitive data identification, Mobile edge computing, Industrial internet

Introduction

With the onset of the big data era, data transmission and collaboration have become widespread, driving significant advancements in information technology research [1–4]. In the context of Mobile Edge Computing (MEC), the emergence of the Industrial Internet heralds the inception of profound integration between the new generation of information technology and the industrial sector [5–9]. Consequently, data within industrial production have undergone rapid aggregation, integration, and processing. The Industrial Internet has been widely adopted across various industries [10], including key sectors such as electric power, petroleum, and chemical industries, with MEC infrastructure facilitating real-time data processing and analysis at the edge [11–13].

*Correspondence:
Shuhan Qiu
shqiu1207@njust.edu.cn
[1] State Grid Jiangsu Electric Power Co., Ltd. Information & Telecommunication Branch, Nanjing, China
[2] Nanjing University of Science and Technology, Nanjing, China

Sensitive data are frequently generated in actual industrial production [14]. These sensitive data in industrial production typically encompass product information, sales volume, production equipment details, and more. In the MEC environment, sensitive data may encompass the user's personal and financial information, which is reflected in the power system may be the electricity bill, electricity consumption, etc. The leakage or theft of such sensitive information can have far-reaching consequences, posing significant risks to both industrial production and individual privacy. Such breaches not only jeopardize the integrity of industrial operations but also pose serious privacy and security threats [15]. Electric power data, serving as a core element in production, has emerged as a pivotal strategic resource for propelling the digital transformation of energy production and establishing a novel type of power system. So, the urgency to bolster data security measures becomes increasingly pronounced in MEC scenarios, where data integrity and confidentiality are paramount. Therefore, identifying sensitive information in the big data environment stands as the initial step in safeguarding sensitive data and constitutes a crucial aspect of the entire process within MEC ecosystems.

Traditional methods of identifying sensitive data typically involve building a sensitive dictionary, extracting data features, and conducting similarity queries. In the power system domain, it is even more common to manually review each data table and data stream to identify sensitive data, albeit in MEC environments, automated processes leveraging edge computing capabilities are becoming more prevalent. These traditional approaches necessitate human intervention, posing two major challenges within MEC ecosystems. Firstly, manual operations often lead to inefficiencies in the overall process, and due to the diversity of production data, the definition of sensitive data will be affected by subjective factors. Secondly, as industrial operations scale up and data volumes rapidly increase, the efficiency of manual intervention in sensitive data identification struggles to keep pace with the growth rate of data in MEC environments. Consequently, it may gradually lose accuracy amidst the deluge of big data, highlighting the need for automated and efficient sensitive data identification techniques tailored for MEC architectures.

Given the challenges outlined above, there arises an urgent necessity for the development of more intelligent and automated sensitive data identification technology tailored for big data environments. This technology should be capable of comprehensively, swiftly, and accurately identifying sensitive data within the vast quantities of data generated in industrial production. Such advancements are crucial to ensuring the effectiveness of subsequent security measures, including protection, desensitization, and control of sensitive data. In recent years, researchers have been exploring smarter sensitive identification techniques leveraging artificial intelligence [16–18]. However, many of these techniques are tailored to specific application scenarios, such as medical data [19, 20], social information [21], or mobile application platforms [22]. Given the variability of semantic information across different domains, it's worth noting that some sensitive words may not be transferable across domains. Even if an existing sensitive data identification model demonstrates high accuracy within one domain, it may require retraining when migrating to other scenarios, consuming a lot of time and computing resources.

In response to the aforementioned challenges, we propose a transferable sensitive data identification model, PQ-SDI, by integrating advanced techniques from natural language processing and text embedding, specifically based on product quantization. The proposed model operates as follows: Firstly, it converts text data into high-dimensional representation vectors using the pretrained language model BERT. Subsequently, these representation vectors are quantized into low-dimensional vectors using product quantization. Finally, the quantized text representation vectors are fed into a feed-forward network for identification, determining whether the text data is sensitive or not. There are several key advantages to our approach. Firstly, leveraging pretrained language models provides a foundational level of knowledge, aiding the sensitive data identification task. Secondly, the use of multiplicative quantization removes irrelevant semantic information, focusing solely on key information pertinent to sensitive data identification, thereby enhancing distinguishability. Moreover, PQ-SDI can effectively identify sensitive data in target domains by leveraging pre-trained models from other domains when training samples in the target domain are limited.

In summary, the primary contributions of this paper can be outlined as follows:

- This paper proposed a sensitive data identification model, named PQ-SDI, based on product quantization, which can automatically and accurately identify sensitive data based on semantic information of text data. PQ-SDI achieves an impressive accuracy of up to 94.41% on real-world datasets, surpassing the performance of all baseline models.
- The PQ-SDI model proposed in the paper is able to identify sensitive data in a new domain after training on a mixed domain dataset. The identification accuracy of the sensitive data can be the same as the model trained on the same domain.

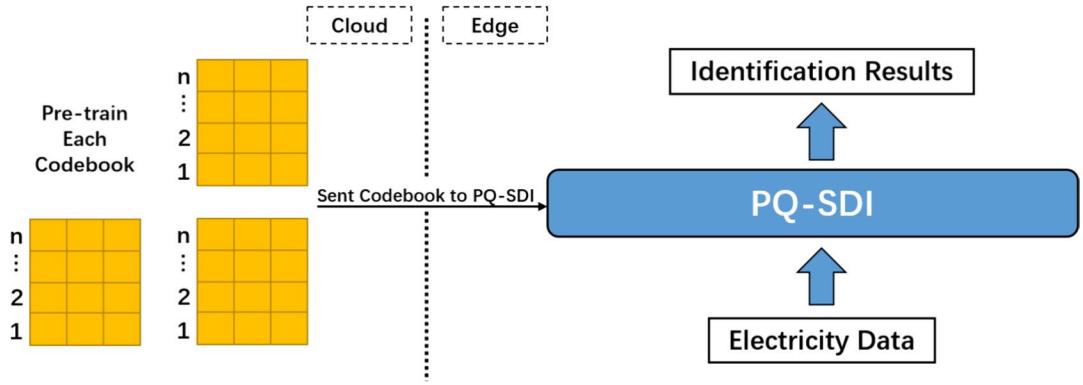


Fig. 1 Illustration of deploying PQ-SDI in a Cloud-Edge environment

- In our experiments, we simulated a mobile edge computing scenario, demonstrating that PQ-SDI effectively reduces parameter scale and minimizes computational resources required for sensitive data identification. This capability enables the sensitive data identification on small mobile terminals by utilizing codebooks sent from the cloud, consequently preventing sensitive data leakage during data transmission. Figure 1 illustrates the schematic diagram of the model in the cloud-edge environment.

Paper overview. In "Related work" section, We give a brief overview of the current state of research on sensitive data identification techniques and the current state of research on natural language processing techniques related to PQ-SDI. In "Methodology" section, We describe in detail the composition of PQ-SDI and the process of identifying sensitive data. In "Experiments" section, We show the details of our experiments on PQ-SDI, the results of the experiments, and the analysis of the results. Finally, we place our conclusion in "Conclusion" section.

Related work

Sensitive data identification and protection

Currently there have been many scholars studying the methods of sensitive data identification in many fields such as healthcare, industry and so on. Bi et al. [14] mentioned that the identification of sensitive information in the industrial internet needs to meet the requirements of timely identification of sensitive data; the ability to support secure decision-making, analysis and sharing; and the enhanced ability to protect data security. Mahendran et al. [23] reviewed the research in the area of data privacy protection, which to the best of our knowledge is the first interdisciplinary review that discusses privacy protection in the technical context of natural language processing.The paper focuses on the application of natural language processing to data security in four sections: healthcare data privacy, privacy protection in technical domains, analysis of privacy preserving strategies, and detection of privacy breaches in textual representations. Yang et al. [22] proposed a method for mobile applications to automatically identify sensitive data, called S3. S3 integrates semantic, syntactic, and lexical information to identify sensitive data mainly through the semantics of descriptive text, and experiments on more than 18,000 apps in the GooglePlay store have achieved good performance, which is a pioneering work in the direction of sensitive data identification. Xu et al. [24] proposed a method for topic identification of sensitive information on the web based on a weighted potential Dirichlet allocation model of sensitive words. The method first generates an embedded representation of sensitive words from manually collected sensitive vocabularies, and then embeds the sensitive vocabularies into the LDA model, thus improving the semantic understanding and identification of sensitive words by LDA. Garcia-Pablos et al. [19] used a pre-trained language model based on BERT to detect and classify sensitive data on several Spanish-language clinical datasets, and the results show that the BERT-based model does not need to be fine-tuned for specific domains in order to have better performance than the other baseline models, and has better stability in the face of insufficient training data. In the mobile edge computing scenario, paper [25] propose a data security mechanism called Fine-Grained Access Control (FGAC), which can ensure data security when accessing data in MEC, overcoming the disadvantages of existing methods such as not considering network attacks. Furthermore, paper [26] propose a privacy preserving data aggregation scheme for MEC-assisted IoT applications, which not only ensures terminal device's

data privacy, but also provides source verification and integrity. Paper [27] also takes into account the computational resource constraints on the device and proposes a multiuser resource allocation and computation offloading model with data security. They also introduced an ANS cryptographic technique as a security layer to protect sensitive data from cyber attacks.

Natural language processing and text classification

With the development of deep learning techniques, the field of natural language processing has also been enhanced. One of the most significant milestones is the proposal of the Transformer [28] architecture, which has driven research in various fields, including the field of natural language processing.The Transformer architecture is a deep learning architecture designed entirely based on the attention mechanism, which is different from previous architectures based on convolutional neural networks (CNN) or recurrent neural networks (RNN).The original proposal of the The Transformer architecture was motivated by the task of machine translation. In recent years, the proposal and development of large language models based on the Transformer architecture has achieved even more impressive results, improving almost all tasks related to text processing.

The most representative large language models are the GPT [29] model proposed by OpenAI, the Llama [30] model proposed by Meta, and the BERT [31] model proposed by Google. The BERT model is the most widely used pre-trained language model, which is a bi-directional encoder based on the Transformer architecture, and the main training method is to pretrain an unsupervised language model on a large amount of unlabeled text data by using the Masked Language Model (MLM) and the Next Sentence Prediction (NSP) task. The pre-trained model can be fine-tuned in downstream applications according to the needs of the scenario to meet the specific task requirements. Up to now, BERT has been successfully applied to many NLP-related tasks by many practitioners [32].

Different from the traditional text classification task in machine learning, deep learning based text classification task requires a large amount of data to allow the model to understand the semantic information of the text, and relies heavily on the quality of the dataset during training. Current deep learning-based text classification methods can be divided into CNN-based methods [33], RNN-based methods [34], GNN-based methods [35], and Attention-based methods [36]. The main application scenarios include topic labeling, sentiment analysis, short text classification and sensitive data identification.

Methodology

In this section, we will propose a transferable method for **S**ensitive **D**ata **I**dentification based on **P**roduct **Q**uantization, named **PQ-SDI**.

Approach overview

In this paper, we consider sensitive data identification as a binary classification task, aiming to determine whether textual information contains sensitive content. Unlike traditional methods, our approach goes beyond simply identifying specific sensitive words, as we analyze the semantic information embedded within the text. To train our model, we consider data from multiple domains, aiming to evaluate the migration ability of PQ-SDI across both familiar and unfamiliar domains. It should be noted that the mixed dataset we use in the training stage is not splicing the samples from different datasets into a single sample, but rather mixing the data from different datasets into a single dataset. For example: there are two datasets $[[a_1, a_2, a_3], [b_1, b_2, b_3]]$ and $[[c_1, c_2, c_3], [d_1, d_2, d_3]]$, we mix them into one dataset $[[a_1, a_2, a_3], [b_1, b_2, b_3], [c_1, c_2, c_3], [d_1, d_2, d_3]]$, instead of splicing them into something like $[[a_1, a_2, a_3, c_1, c_2, c_3], [b_1, b_2, b_3, d_1, d_2, d_3]]$.

To enhance the semantic understanding of text information within our sensitive data identification model, we employ a pre-trained language model based on the Transformer architecture. This architecture, leveraging the attention mechanism, has gained widespread adoption across various research fields in recent years. PQ-SDI initially utilizes the pre-trained language model to encode text information into high-dimensional vectors within the semantic space. However, encountering large volumes of data poses two challenges: high-dimensional vectors consume a lot of computational resources and more storage space is required to store model parameters. To address these challenges, we employ the product quantization technique to compress the embedded representations produced by the pre-trained language model. This compression reduces the computational demands for sensitive data identification while focusing on the semantic information crucial for the task at hand. Finally, the compressed representations are fed into a feed-forward network to perform the identification of sensitive data. The structure of the PQ-SDI model is shown in Fig. 2 and our notation is summarized in Table 1.

Text representation based on product quantization

As previously discussed, the process of embedding text involves two primary steps. Initially, the text undergoes encoding into high-dimensional vectors using a pre-trained language model, effectively embedding it into the feature space. Subsequently, the high-dimensional vectors are reduced to low-dimensional vectors through the

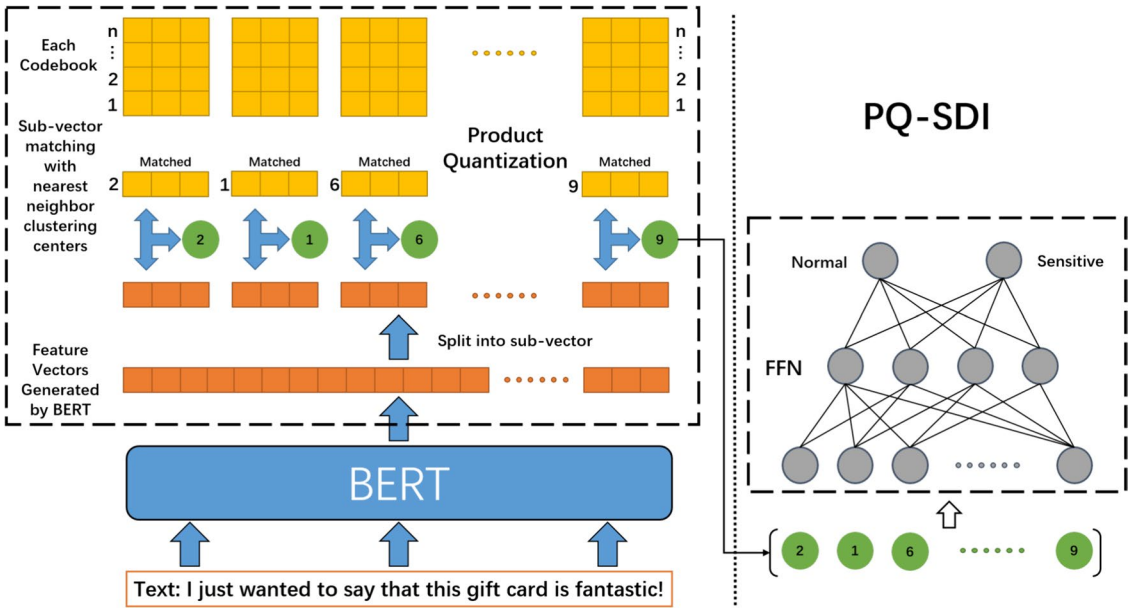


Fig. 2 The overall framework of the proposed transferable sensitive data identification method based on product quantization. It should be noted in particular that the codebook in the figure is from the trained PQ-SDI model, which in this paper is obtained using the K-Means algorithm on the training set

Table 1 Notations

Notation	Description
w_i	Each token in raw data
[CLS]	Special token at the beginning of data
$\boldsymbol{x}_i$	Vector generated by pre-trained language model
d	Vector dimensions generated by pre-trained language model
k	Number of subvectors divided in product quantization
c_i	Indexing of clustering centre in product quantization
j	Number of clustering centres in product quantization
$\boldsymbol{h}_i$	The feature vector processed by product quantization
$\boldsymbol{y}_i$	Identification results of FFN output

application of product quantization techniques, which serve to encapsulate the semantic information contained within the text. The subsequent subsections offer a comprehensive description of the implementation of this process.

Text representations generated by BERT

In this paper, the pre-trained language model we have chosen is the BERT [31] model, which is widely used in academia. The BERT model uses the encoder part of the Transformer to bidirectional encoding the context. However, unlike the original Transformer encoder, BERT uses learnable positional embeddings. The encoder layer consist of multiple Transformer encoders, each of which includes two sublayers: a multi-head self-attention layer and a feed-forward neural network layer. In this case, self-attention score is calculated in the multi-head self-attention layer using the scaled dot product attention formula:

$$score = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{1}$$

The feed-forward neural network layer comprises two fully connected layers aimed at augmenting the network's parameter count and enhancing the encoder's learning capability. In this setup, the ReLU activation function is employed:

$$ReLU(x) = \max(0, x) \tag{2}$$

Each sub-layer is accompanied by a residual connection and a normalization layer. The residual connection serves to diminish the variance between inputs and outputs, mitigating issues like gradient explosion and vanishing. This, in turn, expedites model convergence, reduces training time, and facilitates the addition of more layers to the model. Layer normalization further enhances the process by scaling both inputs and outputs, ensuring that each layer maintains consistent distribution patterns. The formula for layer normalization is:

$$Layer\ Norm(x) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{3}$$

γ and β are scaling and offset factors, μ and σ^2 are the mean and variance computed alone the last dimension, ϵ is a small positive number used to prevent division by 0. The encoder layer can encode the input text sequence and learn the relationship between tokens in the sequence to extract the contextual information of the sequence.

Before inputting text into the model for encoding, we need to preprocess the text with a special token *[CLS]* at the beginning of the text. In adddition, since the BERT model has a limit on the length of input text, we also need to fix the length of each text. We fix the length of text to n, for shorter text we use the special token *[PAD]* to fill the text to length n, and for the text which length greater than n, we truncate the part over length n. So generating vectors with the BERT model can be expressed as:

$$\boldsymbol{x}_i = BERT([CLS], w_1, w_2, ..., w_n) \tag{4}$$

Where $(w_1, w_2, ..., w_n)$ represents a text sequence of length n. *[CLS]* also means classify, which is often used in text classification tasks in the field of natural language processing, because the feature vector of *[CLS]* contains the semantic information of the whole text in the BERT model. In this paper, we also use the vectors characterizing *[CLS]* as the basis of classification for sensitive data identification too, denoted as $\boldsymbol{x}_i \in \mathbb{R}^d$ and d is the vector dimension of the BERT model. It's important to emphasize that the BERT model in PQ-SDI is interchangeable, and careful consideration should be given to the maximum length of the data in the dataset when selecting a pre-trained language model for encoding. Failure to do so may result in the truncation of sensitive information. Since the majority of data lengths in our dataset are less than the input limit of the BERT model, the set length $n$is primarily utilized to standardize data of varying lengths by employing a special token [*pad*] prior to encoding. In essence, n is determined by the longest data in the dataset.

Dimensionality reduction of vectors via product quantization

Next we will utilize Product Quantization (PQ) [37] to quantize the vector output from the BERT model. We split each vector equally into k sub-vectors, the dimension of each sub-vector is d/k. The split operation can be expressed formulaically as $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, ..., x_{i,k}]$. PQ first clusters each subclass in all data samples, and then, for each sub-vector, PQ will search for clustering centers with sub-vector's nearest neighbors and turn it to the index of the clustering center. For example, the sub-vector $x_{i,1}$ of vector $\boldsymbol{x}_i$ is nearest to the second clustering center, so $x_{i,1}$ can be written as the index of the second clustering center 2. Noting all transformed sub-vectors as $\boldsymbol{h}_i$, the formal description can be written as:

$$h_{i,k} = \arg\min_n \|x_{i,k} - c_n\|^2 \in \{1, 2, ..., j\} \tag{5}$$

$$\boldsymbol{h}_i = [h_{i,1}, h_{i,2}, ..., h_{i,k}] \tag{6}$$

where c_n denotes the n-th clustering center and there are a total of j clustering centres. In the process of quantization operation on vectors, the set of numerous cluster centers generated after clustering is called *codebook* and has the shape: $\mathbb{R}^{k \times j \times (d/k)}$. It should be noted that on the cross-domain experiments, the *codebook* is obtained in the model entirely from the training data, and then the data in the target domain generates a vector of down-scaled textual representations based on the *codebook*, thus demonstrating the transferability of the model. The schematic diagram illustrating the learning of the codebook is presented in Fig. 3.

Sensitive data identification

After embedded representation of the input textual information, we next identify the sensitive textual. In PQ-SDI, we take a feed-forward network (FFN) consisting of two fully connected layers for the identification of sensitive data:

$$FFN(\boldsymbol{h}_i) = W_2 \times sigmoid(W_1 \boldsymbol{h}_i + b_1) + b_2 \tag{7}$$

$$sigmoid(\cdot) = \frac{1}{1 + \exp(\cdot)} \tag{8}$$

where $\boldsymbol{h}_i$ is the previously mentioned text embedding vector after product quantization, which is used here as input to the FFN.W_1, W_2 is learnable parameter, The activation function between the two fully connected layers is the sigmoid function commonly used in classification tasks. the output of the FFN can be denoted as $\boldsymbol{y}_i$:

$$\boldsymbol{y}_i = FFN(\boldsymbol{h}_i) \tag{9}$$

where $\boldsymbol{y}_i$ is a two-dimensional vector representing the probability distribution of whether the text data is sensitive or not.

Model training and optimization

Finally, we describe the loss function and optimization methods used in training PQ-SDI. Since the sensitive data identification task solved by the model is regarded as a binary classification task, the loss function employed in training the model is the widely used Cross-Entropy Loss [38] function:

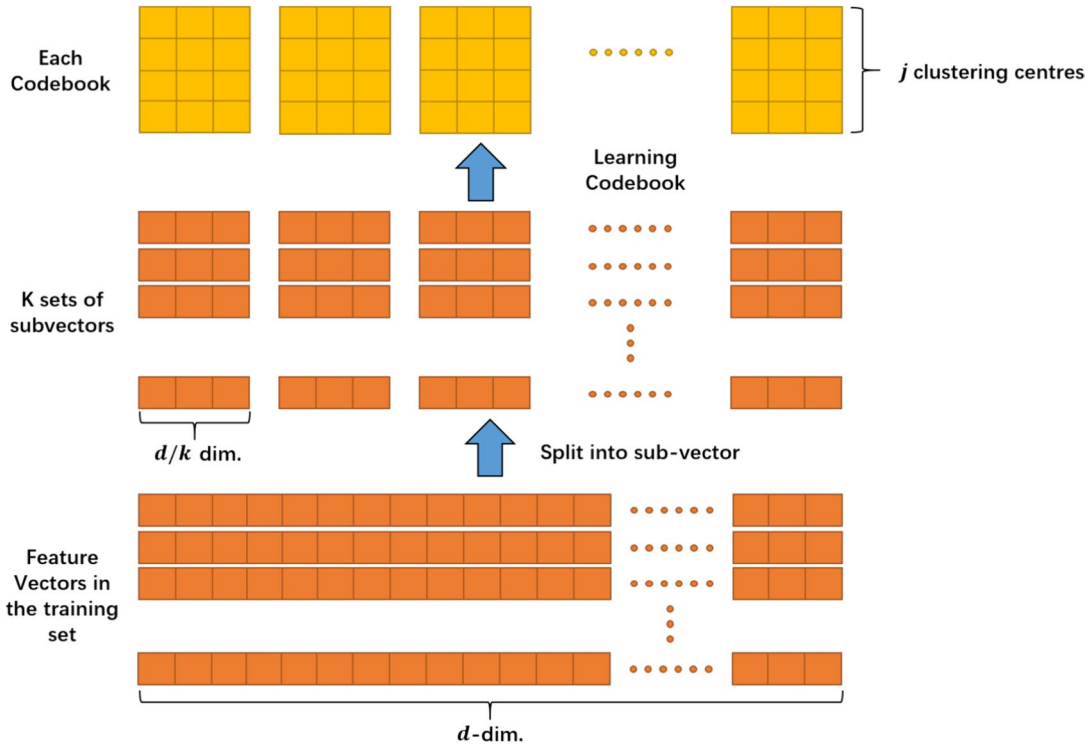


Fig. 3 Schematic for learning codebook in product quantization

$$Loss_{CrossEntropy} = -\frac{1}{m}\sum_{j=1}^{m}(r_{j1}\log y_{j1} + r_{j2}\log y_{j2}) \tag{10}$$

where m is the number of samples entered into the model for training each time, $r_j = (r_{j1}, r_{j2})$ is the ground truth of the sample, generally (0, 1) or (1, 0). y_{j1} and y_{j2} comes from the PQ-SDI, the probabilistic prediction whether the sample is sensitive or not, obviously $y_{j1} + y_{j2} = 1$.

The optimizer used in training is the Adam [39] optimizer, which is widely used in the current deep learning field, and is a variant algorithm derived from the stochastic gradient descent algorithm with adaptive moment estimation. We can present it formulaically:

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t \tag{11}$$

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2 \tag{12}$$

$$\hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t} \tag{13}$$

$$\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1-\beta_2^t} \tag{14}$$

$$\theta_{t+1} = \theta_t - \alpha\frac{\hat{\boldsymbol{m}}_t}{\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon} \tag{15}$$

In the above equation, $\boldsymbol{g}_t$ is the gradient at step t, $\boldsymbol{m}_t$ and $\boldsymbol{v}_t$ are estimate of the first-order moments and second-order moments at step t, θ_t is the parameter at step t, α is learning rate, β_1 and β_2 are decay rates of the first-order moments and the second-order moments, ϵ is a small positive number used to prevent division by 0.

Experiments

In this section, we will organize our experiments around the following Research Questions (RQs) to demonstrate the sophistication of our proposed model PQ-SDI:

- **RQ1**: How does the model perform on datasets from different domains?
- **RQ2**: How well the model performs in the new domain after training on the mixed training set?
- **RQ3**: Is the quantization operation on the embedding vectors generated by BERT necessary for the sensitive data identification task?

Table 2 Statistical information on datasets after data preprocessing

Datasets	Review amount	Average review length	Normal	Sensitive amount	Sensitive rate
All Beauty	341406	206.9	281896	59510	17.43%
Gift Cards	144039	95.1	135231	8808	6.12%
Magazine Subscriptions	82687	238.05	66436	16251	19.65%
Software	419976	418.7	285989	133987	31.90%

Experimental setup

In this section, we present the dataset we chose, the evaluation metrics used in the experiment and the implementation details of PQ-SDI.

Dataset

We will initially evaluate the model's performance in identifying sensitive data across four real-world public datasets from various domains: All Beauty, Gift Cards, Magazine Subscriptions, and Software datasets sourced from the Amazon Review Dataset [40]. These datasets comprise user reviews of diverse products on the Amazon platform as of 2018. Subsequently, we will curate a subset of data from these three datasets to construct a mixed training set for model training. We will then assess the model's capability to identify sensitive data within target domains that were not included in the training set. The statistical information for the four datasets is presented in Table 2.

As previously mentioned, the definition of sensitive data varies across different domains and is subject to constant evolution in real-world production settings. So, it is difficult to have a unified standard for defining sensitive data. In this paper, what we need to ensure is that there is the same definition of sensitive on the data in different domains. Therefore, in our experiments, we define user-generated negative reviews as sensitive data. Across various domains, users may articulate their dissatisfaction with products differently, but typically, such dissatisfaction is reflected in low ratings. Hence, we uniformly classify reviews associated with low ratings as negative reviews. The primary task of PQ-SDI is to swiftly and accurately identify sensitive data on new datasets after being trained on mixed datasets.

Evaluation metrics

For the sensitive data identification task, the primary metric for assessing the effectiveness of a model lies in its ability to accurately distinguish between normal text and sensitive text. Therefore, in this paper, we utilize Accuracy as the key metric to evaluate the model's effectiveness in identifying sensitive data.

Implementation details

As introduced to the model in "Related work" section, the identification of sensitive data by PQ-SDI contains two main steps: one is to embed the text data into the feature space, and the other is to identify sensitive text data by a feed-forward network.The BERT model used in the experiments is from Huggingface 'models–bert-base-uncased', which is available from Huggingface[1]. The text is passed through the BERT model and the output is a 768-dimensional vector, which is input into the product quantization module and then split the 768-dimensional vector into 32 sub-vectors of 48 dimensions each. When clustering codebook with K-Means algorithm, due to the limitation of computational resources, the samples used on a single NVIDIA GeForce RTX3090 are about 70,000-80,000 each time, and the clustering center of each group of subvectors is set to 256, which can be said that each group of subvectors is divided into 256 classes, and the theoretical number of samples that can be characterized reaches $256^{32} = 2^{256}$, which is far more than that of the experimental samples, and it almost doesn't produce the problem that the different texts have the same embedding representations. Since the feature vector after product quantization has 32 dimensions, the dimensions of the feed-forward network used to identify the sensitive data are set to 32,16,2 respectively, so the parameter matrices in the model are 32×16 and 16×2 respectively, and the learning rate when training the feed-forward network is set to *1e-4*. In our experiments, we also discovered that the embedding vector's lower dimensionality resulting from product quantization leads to a reduced demand for computational resources during the training of the feed-forward network, to a certain extent.

Cloud edge setup

To evaluate the practicality of deploying PQ-SDI in cloud-edge environment, we divided the deployment process into two stages. Firstly, we conducted the pre-training of the codebook on a cloud server. Next, we deployed the feed-forward neural network for

[1] https://huggingface.co/bert-base-uncased

Table 3 Performance of different models in identifying sensitive data on various datasets. The data with the highest accuracy on each dataset has been denoted in bold

Model	All beauty	Gift cards	Magazine subscriptions	Software
TF-IDF-RF	0.8226	0.9224	0.805	0.7579
BERT-RF	0.774	0.899	0.779	0.74
PQ-SDI	**0.8251**	**0.9441**	**0.8226**	**0.7669**

sensitive data identification on a local personal computer equipped with an NVIDIA GeForce RTX 2060 graphics card, offering 13.9 GB of available graphics memory. Since our experiments involved many trials, we ran the experiments related to RQ2 and RQ3 on cloud-edge environment.

Performance on identify sensitive data (RQ1)

To address the first problem, we conducted experiments and analysis. Specifically, we selected several sensitive data identification methods based on machine learning as baseline models for comparison with the PQ-SDI model proposed in this paper. The experiments were conducted across four real-world datasets to evaluate the effectiveness of these methods for the sensitive data identification task.

Baselines

We compare the PQ-SDI with the following baseline models:

- **TF-IDF-RF**: Sensitive data identification techniques integrating TF-IDF [41] algorithm and Random Forest [42] classification algorithm outside the deep learning domain are selected.
- **BERT-RF**: We replace the TF-IDF algorithm with the pre-trained language model BERT used in PQ-SDI to encode the text, which is then fed into the Random Forest algorithm to identify the sensitive data.

Performance on identify sensitive data

Based on the previously mentioned dataset, evaluation metrics, experimental details, and baseline models, we conducted comprehensive experiments and analyzed the results. The outcomes of these experiments are presented in the Table 3.

To maintain fairness, we employed identical data preprocessing methods across all experiments, ensuring consistency in the data used for analysis. Our findings indicate that PQ-SDI outperforms all baseline models across diverse datasets in identifying sensitive data effectively. Additionally, we observed that, for the Random Forest algorithm, the text representations generated by the BERT model resulted in diminished performance in sensitive data identification compared to the TF-IDF algorithm. This underscores the necessity for quantizing the representation vectors produced by the BERT model.

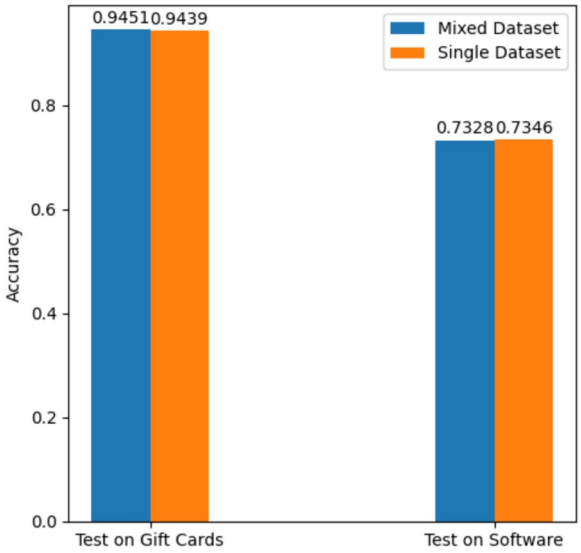
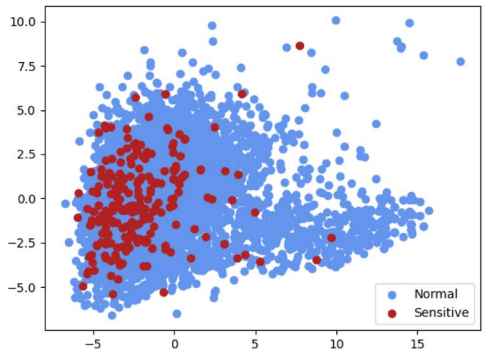


Fig. 4 Performance of PQ-SDI in cross domain experiments

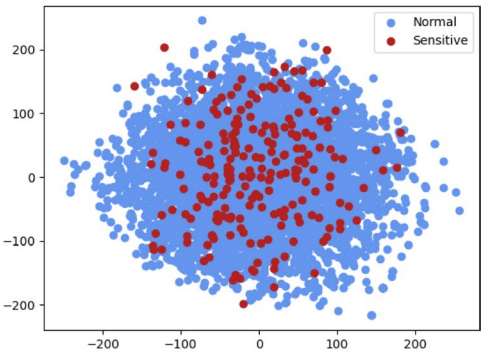
Cross domain experiments (RQ2)

Next, we conducted experiments and analysis to evaluate the transferability of PQ-SDI. Our approach involved selecting a portion of data from each of the three datasets to create a mixed dataset for training the model, primarily for generating the codebook in PQ-SDI. Subsequently, we deployed the trained model into a new domain for sensitive data identification and observed its performance in identifying sensitive data within the new domain. To assess the transferability of PQ-SDI, we compared the model trained on the mixed dataset with the model trained specifically in the target domain. If the model trained on the mixed dataset performs comparably to the model trained in the target domain in terms of identifying sensitive data, then we can conclude that PQ-SDI exhibits good transferability. The specific experimental results are presented in the Fig. 4.

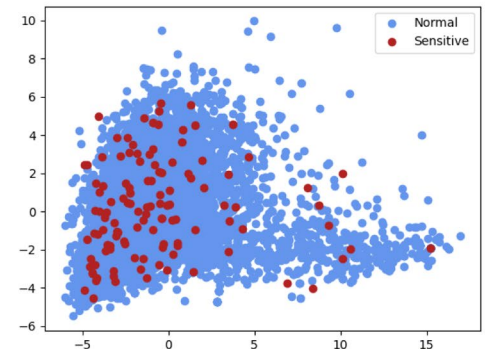
We conducted two main transferability experiments. In the first experiment, we trained PQ-SDI on a mixed dataset comprising the Software, All Beauty, and Magazine Subscriptions datasets. Subsequently, we evaluated its performance in sensitive data identification on the Gift Cards dataset. Our findings revealed that the model's ability to identify sensitive data in the new domain after training on the mixed-domain dataset was slightly superior to training



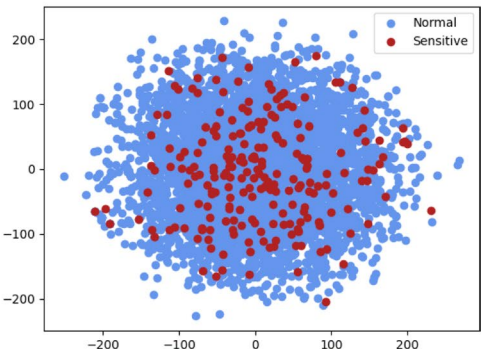
(a) Distribution of embedding vectors in space generated by the BERT model on the Gift Cards dataset.



(b) Distribution of embedding vectors in space after product quantization on the Gift Cards dataset.



(c) Distribution of embedding vectors in space generated by the BERT model on the All Beauty dataset.



(d) Distribution of embedding vectors in space after product quantization on the All Beauty dataset.

Fig. 5 Comparison plot after product quantization

on the original dataset and testing it. In the second experiment, we observed that the sensitive data identification performance in the new domain, Software, after training on the mixed dataset consisting of Gift Cards, All Beauty, and Magazine Subscriptions, was marginally weaker compared to training and testing on the original dataset. However, the difference was not statistically significant. Overall, based on these two experiments, we can conclude that PQ-SDI exhibits some degree of transferability and can swiftly adapt to sensitive data identification tasks in new scenarios.

Ablation study: effect after adding product quantization (RQ3)

In this section, we investigate the extent to which the incorporation of the product quantization technique enhances the model's ability to identify sensitive data. To begin, we compare the distribution of vectors in the feature space after the product quantization process with the distribution of unprocessed vectors. For this purpose, we randomly selected 5,000 data points from two datasets, Gift Card and All Beauty, for visualization. The results of this comparison are presented in Fig. 5

We observed that the product quantization technique effectively clusters irregularly distributed samples in the feature space towards the central region, irrespective of the dataset. This clustering phenomenon reduces semantic gaps between domains, thereby enhancing the effectiveness of sensitive data identification and improving the model's transferability. For comparison, we devised a variant of the PQ-SDI model

Table 4 Ablation study: comparison of sensitive data identification effect after removing product quantization

Model	All beauty	Gift cards	Magazine subscriptions
BERT-SDI	0.7962	0.9339	0.8013
PQ-SDI	**0.8251**	**0.9441**	**0.8226**

called BERT-SDI, where we removed the product quantization technique. Instead, we directly input the embedding vectors generated by BERT into the feed-forward neural network. In BERT-SDI, the number of neurons per layer in the feed-forward network was set to 768, 384, and 2, and the size of the learnable parameter matrices in the model was adjusted accordingly 768×384 and 384×2. In the subsequent table, we compare the performance of BERT-SDI and PQ-SDI in identifying sensitive data across the three datasets.

From the results presented in Table 4, we observed that the PQ-SDI model, augmented with the product quantization technique, demonstrates improvements across all sensitive data identification tasks. Notably, PQ-SDI exhibits the smallest enhancement in the Gift Card dataset, primarily due to the lower rate of sensitive data instances within this dataset. Conversely, the other two datasets exhibit comparable rates of sensitive data instances, resulting in similar degrees of enhancement brought about by PQ-SDI. Based on these findings, we can conclude that PQ-SDI consistently performs better, particularly when confronted with datasets containing higher proportions of sensitive data instances. Indeed, during the process of product quantization, the compression and clustering of vectors inevitably result in some information loss. However, this loss does not compromise the accuracy of sensitive data identification. This observation indicates the presence of redundant information in the original high-dimensional vectors that is unrelated to the task of sensitive data identification. The product quantization technique effectively filters out this redundant information, thereby conserving resources required for downstream tasks.

The results of the ablation study show the significance of PQ-SDI in MEC scenarios. While the parameters of BERT-SDI are too large to deploy in the cloud-edge environment established previously, making the comparison in the ablation study somewhat "unfair", the identification performance of PQ-SDI on sensitive data in the cloud-edge environment surpasses that of BERT-SDI deployed in the cloud. This observation highlights the performance of our model in MEC scenarios.

Conclusion

In this paper, we propose a novel sensitive data identification method named PQ-SDI, tailored specifically for the Mobile Edge Computing environment. Leveraging a combination of pre-trained language models and product quantization, PQ-SDI offers a robust solution for identifying sensitive data in real-time mobile edge scenarios. Initially, PQ-SDI utilizes a pre-trained language model BERT, to generate embedding representations of data. Subsequently, it employs a product quantization technique to compress the high-dimensional embedding vectors into lower-dimensional representations, facilitating sensitive data identification. Finally, these representations are inputted into a feed-forward network for further analysis and identification, enabling efficient processing and classification of sensitive data at the edge. Through our experiments, we demonstrate that PQ-SDI excels in identifying sensitive data and exhibits the capability to generalize to new datasets after training on a mixed dataset. Additionally, the incorporation of vector quantization reduces the computational resource requirements during the sensitive data identification phase, addressing the challenges posed by the era of massive data growth in mobile edge scenarios. Furthermore, our approach holds promise for extension to small devices within power information networks, serving as a proactive measure to prevent sensitive data leakage resulting from data transmission processes in MEC environments.

Our proposed method is designed to evolve alongside advancements in natural language processing technology within the realm of deep learning, specifically tailored for MEC applications. This adaptability allows for the flexible selection of various pre-trained language models to suit different data environments in MEC scenarios. Moving forward, we aim to further enhance the effectiveness of our approach by refining the feed-forward network used in the sensitive data identification phase. This ongoing research will focus on developing more efficient techniques for sensitive data identification, ultimately improving the overall performance of our method in MEC environments.

Acknowledgements

This work was supported by the Science and Technology Project of State Grid Jiangsu Electric Power Company Ltd., under Grant J2023179.

Authors' contributions

Zhao gave the idea of PQ-SDI first and conducted the main experiments. Yuan and Qiu helped finish experiments. Zhao, Yuan, Qiu, Xu and Wei wrote the main manuscript. Qiu and Wei prepared the figures and tables. All authors reviewed the manuscript.

Funding

Science and Technology Project of State Grid Jiangsu Electric Power Company Ltd., under Grant J2023179.

Availability of data and materials
No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 19 March 2024 Accepted: 29 April 2024
Published online: 08 May 2024

References

1. Wang F, Wang L, Li G et al (2022) Edge-cloud-enabled matrix factorization for diversified apis recommendation in mashup creation. World Wide Web 25(5):1809–1829
2. Qi L, Xu X, Wu X et al (2023) Digital-twin-enabled 6g mobile network video streaming using mobile crowdsourcing. IEEE J Sel Areas Commun 41(10):3161–3174. https://doi.org/10.1109/JSAC.2023.3310077
3. Gu R, Chen Y, Liu S et al (2022) Liquid: Intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed gpu clusters. IEEE Trans Parallel Distrib Syst 33(11):2808–2820. https://doi.org/10.1109/TPDS.2021.3138825
4. Wang F, Zhu H, Srivastava G et al (2022) Robust collaborative filtering recommendation with user-item-trust records. IEEE Trans Comput Soc Syst 9(4):986–996. https://doi.org/10.1109/TCSS.2021.3064213
5. Xu X, Tang S, Zhou X et al (2023) Cnn partitioning and offloading for vehicular edge networks in web3. IEEE Commun Mag 61(8):36–42. https://doi.org/10.1109/MCOM.002.2200424
6. Dai H, Wang X, Lie A et al (2023) Omnidirectional chargability with directional antennas. IEEE Trans Mob Comput. https://doi.org/10.1109/TMC.2023.3294391
7. Li Z, Xu X, Hang T et al (2022) A knowledge-driven anomaly detection framework for social production system. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/TCSS.2022.3217790
8. Dai H, Xu Y, Chen G et al (2022) Rose: Robustly safe charging for wireless power transfer. IEEE Trans Mob Comput 21(6):2180–2197
9. Dai H, Wang X, Lin X et al (2023) Placing wireless chargers with limited mobility. IEEE Trans Mob Comput 22(06):3589–3603. https://doi.org/10.1109/TMC.2021.3136967
10. Xu X, Gu J, Yan H et al (2023) Reputation-aware supplier assessment for blockchain-enabled supply chain in industry 4.0. IEEE Trans Ind Inform 19(4):5485–5494
11. Xu X, Li H, Li Z et al (2023) Safe: Synergic data filtering for federated learning in cloud-edge computing. IEEE Trans Ind Inform 19(2):1655–1665
12. Yang C, Xu X, Zhou X, et al (2022) Deep q network-driven task offloading for efficient multimedia data analysis in edge computing–assisted iov. ACM Trans Multimedia Comput Commun Appl 18(2s):1–24
13. Gu R, Zhang K, Xu Z, et al (2022) Fluid: Dataset abstraction and elastic acceleration for cloud-native deep learning training jobs. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). https://doi.org/10.1109/ICDE53745.2022.00209
14. Bi T, Li J, Chen X (2020) Research on industrial internet sensitive data identification technology based on traffic analysis. In: 2020 Chinese Automation Congress (CAC), IEEE, pp 1021–1023
15. Kong L, Wang L, Gong W et al (2022) Lsh-aware multitype health data prediction with privacy preservation in edge environment. World Wide Web 25(5):1793–1808
16. Senavirathne N, Torra V (2020) On the role of data anonymization in machine learning privacy. In: 2020 IEEE 19th International conference on trust, security and privacy in computing and communications (TrustCom), IEEE, pp 664–675
17. Nikoletos S, Vlachos S, Zaragkas E et al (2023) Rog §: A pipeline for automated sensitive data identification and anonymisation. In: 2023 IEEE International Conference on Cyber Security and Resilience (CSR), IEEE, pp 484–489
18. Jie S, Cui S, Chen F, et al (2023) Sensitive data discovery technology based on artificial intelligence. In: Proceedings of the 2nd International Conference on Information Economy, Data Modeling and Cloud Computing, ICIDC 2023, June 2–4, 2023, Nanchang
19. García-Pablos A, Perez N, Cuadros M (2020) Sensitive data detection and classification in spanish clinical text: Experiments with bert. arXiv preprint arXiv:2003.03106
20. Kong L, Li G, Rafique W et al (2022) Time-aware missing healthcare data prediction based on arima model. IEEE/ACM Trans Comput Biol Bioinform. https://doi.org/10.1109/TCBB.2022.3205064
21. Perikos I, Michael L (2023). Sensitive content recognition in social interaction messages. https://doi.org/10.4108/EAI.2-6-2023.2334615
22. Yang Z, Liang Z (2018) Automated identification of sensitive data from implicit user specification. Cybersecurity 1:1–15
23. Mahendran D, Luo C, Mcinnes B (2021) Privacy-preservation in the context of natural language processing. IEEE Access 9:147600–147612
24. Xu G, Wu X, Yao H et al (2019) Research on topic recognition of network sensitive information based on sw-lda model. IEEE Access 7:21527–21538
25. Hou Y, Garg S, Hui L, Jayakody DNK, Jin R, Hossain MS (2020) A data security enhanced access control mechanism in mobile edge computing. IEEE Access 8:136119–136130
26. Li X, Liu S, Wu F, Kumari S, Rodrigues JJ (2018) Privacy preserving data aggregation scheme for mobile edge computing assisted iot applications. IEEE Internet Things J 6(3):4755–4763
27. Elgendy IA, Zhang W, Tian Y, Li K (2019) Resource allocation and computation offloading with data security for mobile edge computing. Futur Gener Comput Syst 100:531–541
28. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Adv Neural Inf Process Syst 30:5998–6008
29. Brown T, Mann B, Ryder N et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901
30. Touvron H, Lavril T, Izacard G, et al (2023) Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971
31. Jacob D, Chang M, Lee K, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
32. Liu Y, Lapata M (2019) Text summarization with pretrained encoders. https://doi.org/10.18653/v1/d19-1387. arXiv preprint arXiv:1908.08345
33. Wang X, Kim H (2018) Text categorization with improved deep learning methods. J Inf Commun Converg Eng 16(2):106–113
34. DING F, SUN X (2022) Negative-emotion opinion target extraction based on attention and bilstm-crf. Comput Sci 49:223–230
35. Li R, Chen H, Feng F, et al (2021) Dual graph convolutional networks for aspect-based sentiment analysis. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, pp 6319–6329
36. Prabhakar S, Won D (2021) Medical text classification using hybrid deep learning models with multihead attention. Comput Intell Neurosci. https://doi.org/10.1155/2021/9425655
37. Jegou H, Douze M, Schmid C (2010) Product quantization for nearest neighbor search. IEEE Trans Pattern Anal Mach Intel 33(1):117–128
38. Zhang Z, Sabuncu M (2018) Generalized cross entropy loss for training deep neural networks with noisy labels. Adv Neural Inf Process Syst 31:8778–8788
39. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
40. Ni J, Li J, McAuley J (2019) Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, pp 188–197
41. Aizawa A (2003) An information-theoretic perspective of tf-idf measures. Inf Process Manag 39(1):45–65
42. Breiman L (2001) Random forests. Mach Learn 45:5–32

Publisher's Note