Engineering Applications of Artificial Intelligence 75 (2018) 11–21

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Email security level classification of imbalanced data using artificial neural network: The real case in a world-leading enterprise

Jen-Wei Huang [a,*], Chia-Wen Chiang [b], Jia-Wei Chang [c]

[a] Department of Electrical Engineering, National Cheng Kung University, 70101, Tainan, Taiwan ROC
[b] Institute of Computer and Communication Engineering, National Cheng Kung University, 70101, Tainan, Taiwan ROC
[c] Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, 40401, Taichung, Taiwan ROC

ARTICLE INFO

Keywords:
E-mail
Classifier
Text mining
Artificial neural network

ABSTRACT

Email is far more convenient than traditional mail in the delivery of messages. However, it is susceptible to information leakage in business. This problem can be alleviated by classifying emails into different security levels using text mining and machine learning technology. In this research, we developed a scheme in which a neural network is used to extract information from emails to enable its transformation into a multidimensional vector. Email text data is processed using bi-gram to train the document vector, which then undergoes under-sampling to deal with the problem of data imbalance. Finally, the security label of emails is classified using an artificial neural network. The proposed system was evaluated in an actual corporate setting. The results show that the proposed feature extraction approach is more effective than existing methods for the representations of email data in true positive rates and F1-scores.

1. Introduction

Email is a common way to communicate with others nowadays, especially in a corporation and between companies. Employees usually receive and send lots of information via emails every day. However, the enormous amount of emails has brought some difficulties, such as spamming (Ismaila et al., 2015), privacy threat (Snchez and Batet, 2017) or information leakage. Especially, when employees send emails to people outside the company, they may accidentally or intentionally include some confidential information. If managers do not scrutinize the outbound emails, the sensitive information will be exposed to the public or the competitive company, which will result in significant loss of the corporate. However, it is a massive task for managers because the vast numbers of emails are required to be inspected within one day.

To address this urgent problem, this study proposes an effective system to assist in the security level classification of corporate emails using text mining and machine learning techniques. The proposed system can scrutinize lots of contents of emails and prevent sensitive information from leaking based on the emails security labels. In the corporation, all emails will be examined by the proposed email security classification system before being sent to other companies. If the classified security level does not match the label provided by email users, the email will be suspended from sending out and reported to the corresponding manager. In this way, the amount of email needed to be checked by managers can be reduced significantly.

However, there are practical issues needed to be solved: (1) Due to the privacy policy of the corporation (subject), we cannot use the meta-data of emails, including senders and receivers. The only available information to classify emails is text. Under the very limited information, Emails are required to be classified into different security levels according to the textual contents of the emails and the attachments. (2) The portion of sensitive email data in the real world is much less than those of usual email data, which results in significant differences between the amount of data in each class. Therefore, the proposed method must address the data imbalance problem. (3) A DNN model with excellent performance requires enormous amounts of training data. Dealing with such volumes of data imposes considerable difficulties concerning time and computational overhead. In our test case, there are more than 150 thousands of emails which are short and imbalanced.

To conquer three major problems mentioned above, we proposed the corresponding methods to them as follow: (1) We need to extract the meaningful textual features of the email body and the attachments. However, emails are usually short to have very limited information. Therefore, we improve the neural network based paragraph vector (Le and Mikolov, 2014) to represent processed emails. The traditional paragraph vector disregards the order of individual words in the text.

* Corresponding author.
E-mail address: jwhuang@mail.ncku.edu.tw (J.-W. Huang).

https://doi.org/10.1016/j.engappai.2018.07.010
Received 4 February 2018; Received in revised form 23 July 2018; Accepted 29 July 2018
Available online 7 August 2018

We regard that the same word with different neighboring words would have different meanings. The combination of consecutive words should be seen as a meaningful unit. Therefore, we use bi-grams as a basic unit instead of a single word to be input into the paragraph vector. In addition, using bi-grams enlarges the training corpus and includes more information of the order of words via bi-grams. The proposed Bi-PVDBOW method can represent more semantic meanings of the contents. The experimental results show that the proposed method outperforms original paragraph vectors. Finally, we use the bi-grams-based paragraph vector as an input for classifying the security level of emails by artificial neural network. (2) On the issue of imbalanced security labels, we compared different methods. To avoid overfitting and maintain the distribution of features of original data, we apply K-means clustering to do undersampling. The cluster-based majority undersampling can effectively avoid the critical information loss of majority class. (3) To make our service efficient, we build a distributed system when parsing emails and preprocessing text data on Spark (Zaharia et al., 2010) and Hadoop (White, 2012), which utilized the cloud computing technique on multiple machines.

In the experimental design, one real corporation voluntarily uses the system to scrutinize its vast numbers of emails. The goal of the experiments is to validate the performance of the proposed system and its practical value in the real world. The results show the proposed system achieves high true positive rate and F1-scores in predicting unseen emails' security levels and the average processing time is concise. All validation results are good enough, which implies that the proposed email classification system can effectively and expeditiously operate in the corporation and help to control the sensitive information in real time. Furthermore, the effects of undersampling were examined. The experimental results show that the sampled data can competently represent the original data in the majority class. Additionally, the classification results are worse without undersampling for minority class. Importantly, only less than 10% of emails are reported to managers on real corporate data. The results indicate the proposed system can help companies protecting sensitive information in emails in such an effective and efficient way.

The remainder of the paper is organized as follows. Section 2 outlines the previous work related to the techniques used in this study. Section 3 describes the proposed email classification system in detail. Section 4 shows the experiments aimed at evaluating the prediction results generated by the system. Finally, the conclusions are given in Section 5.

2. Related work

2.1. Security level classification for documents

Security-level labeling of a document is a document classification problem. The purpose of document classification is to assign predefined labels to a new material that is not classified (Joachims, 1998). We have to first transform the textual data into a relational and analytical form. Then the new representation of documents can be fed into a classifier. However, in the security-level classification problem, the amount of data of confidential class is much less than nonconfidential data. We have to deal with the imbalance data problem.

For security-level classifier, Alparslan and Bahsi (2009) used Support Vector Machine (SVM) and Naive Bayes (NB) to classify confidential documents, the SVM on 59 test documents achieved best overall accuracy, 89.83% (53/59). Alparslan et al. (2013) further proposed the SVM-Adaptive Neuro-Fuzzy Inference System that achieved the best overall accuracy, 96.67% (57/59), on 59 test documents. Shakir et al. (2016) proposed an ensemble approach that combines SVM, NB, Decision Trees, and K-Nearest Neighbor. The ensemble approach achieved the classification performance on legal document filtering around 90% Precision, Recall, and F-measure. However, previous works did not deal with the issue of imbalanced data. Therefore, we aim to design a robust classifier learnt from large-scale datasets with short texts and imbalanced classes.

2.2. Document representation

Classification performance of textual data is very relevant to the preprocessing tasks (Han and Kamber, 2006). Effective feature extraction can greatly facilitate machine learning. In this work, the only available information is text contents provided by the corporation. Textual data needs to be formatted in a relational and analytical form. In some works (Alparslan and Bahsi, 2009; Alparslan et al., 2013; Shakir et al., 2016), Term Frequency–Inverse Document Frequency (TF–IDF) is used to represent text-based contents. Using this representation, each of distinct term in the document set is a dimension of the TF–IDF representation. However, TF–IDF leads to a very high-dimensional representation. For example, the dataset used by Alparslan and Bahsi (2009) and Alparslan et al. (2013) has only 222 documents but includes over 2.5 million of words. Therefore, TF–IDF representation may not be appropriate to deal with emails.

This necessitates the formulation of a document representation to replace the article (Jain and Yu, 1998). Numerous methods have been devised for the retrieval of information from text. Word frequency, TF–IDF, is the most common way to retrieve intelligence from text (Salton and Buckley, 1988). Topic models, such as Latent Dirichlet Allocation, LDA (Blei et al., 2003), are statistical models used to discover the topic of a document by extracting latent topical information from the document to generate a document-topic distribution map. LDA is used to derive the topic distribution of a document. In some cases, a distributed vector can be used as a representation of the document.

However, those methods are hard to extract the semantic meaning of words in the document. The neural network is used to learn word vectors for the representation of documents and sentences, in a process referred to as word embedding (Mikolov et al., 2013a; Mikolov, 2012; Mikolov et al., 2013b). Word vectors were derived from the neural probability language model proposed in Bengio et al. (2003). In the neural probability language model, each input word is represented by a vector and then concatenated or averaged to predict subsequent words in the text. Probability prediction is transformed into a multi-class classification with the architecture of two-layer neural network.

Paragraph vectors (Le and Mikolov, 2014) is an extension of word vectors (Mikolov et al., 2013a) to construct embeddings from entire documents using a two-layer neural network, which differ from word vectors by the fact that the input includes a paragraph id. The Skipgram and DBOW of word vector model can be used to compute the paragraph vector by adding a paragraph id. The first architecture can be used to predict the word immediately after the training word. Following the training process, paragraph vectors and word vectors are unique. It operates like a record of missing contents, and is therefore referred to as Distributed Memory model of Paragraph Vector, PVDM (Le and Mikolov, 2014). The second architecture is the Distributed Bag of Words version of Paragraph Vector, PVDBOW (Le and Mikolov, 2014), which does not consider the order of the words. It aims to predict the presence of words in the document. In each training iteration, words from the text window are randomly sampled in order to formulate a classifier capable of predicting the words in a paragraph vector. Paragraph vectors can be used to deal with text of any length.

2.3. Preprocessing methods of imbalanced data

Machine learning-based classifiers learn the characteristics of data by minimizing the error rate. However, the results make sense only if each class of data is balanced. Sampling is one approach to the balancing of data. There are two types of sampling: over-sampling and under-sampling. The over-sampling approach involves increasing the amount of minority class data (Han et al., 2005). The authors oversampling the minority class until the amounts of data in each class balanced. However, over-sampling can lead to problems in the cross validation. Fig. 1 shows two problems of the cross validation due to over-sampling when conducting cross validation. Fig. 1(a) shows over-sampling prior

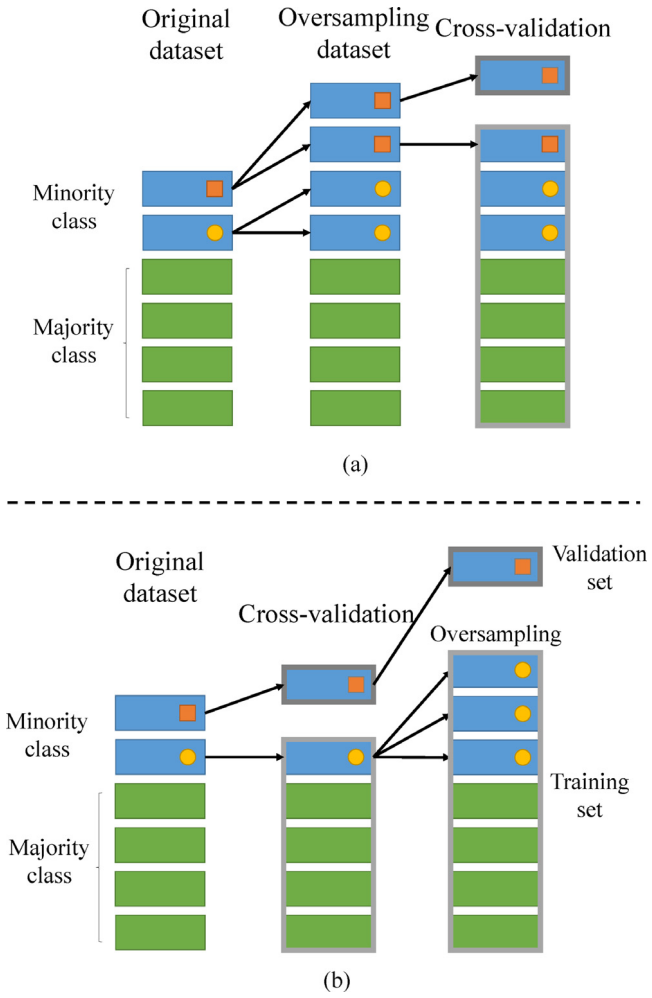

Fig. 1. Over-sampling and cross validation: (a) over-sampling before cross validation; (b) over-sampling after cross validation.

to the cross validation, which lead to the training dataset containing the same data in the validation dataset. Thereby, this way prevents the formulation of a fair classifier. As shown in Fig. 1(b), if the minority class is over-sampled after the cross validation, the minority class in the validation set will tend to be unique and under represented in the training data, which would undermine the ability of the classifier to make predictions.

On the other hand, Zhang and Mani (2003) examined four under-sampling methods of selecting majority samples from the class distribution. They proposed Near Miss-1, Near Miss-2, Near Miss-3 and the distance method. NearMiss methods select the points from the majority class that are close to all or some of the minority points. The distance method selects the majority points whose average distance are largest to three closest minority points. However, Near-Miss methods have significant dropping in Recall as the samples increases, and the distance method is overgeneralization of the minority class.

Another preprocessing method, the cost-sensitive learning, gives different costs for different classes which are considered as the importance of classes (Zhou and Liu, 2006; Zhang et al., 2010). However, it is difficult to define the cost for each class. Zhang et al. (2010) dealt with above problems by an unsupervised technique, K-means clustering (Hartigan and Wong, 1979). The cluster-based majority undersampling approach selects a representative subset from the majority class, and uses the representative subset and all samples of the minority class as the training data. Zhang et al. found that compared to other under-sampling methods, cluster-based majority under-sampling can effectively avoid the critical information loss of majority class. Therefore, we adopted K-means clustering to deal with the problem of imbalanced data.

2.4. Classification

For the classification tasks, there are many methods being devised, such as SVM (Suykens and Vandewalle, 1999), Decision Trees (Safavian and Landgrebe, 1991), and Neural Networks (Hecht-Nielsen, 1988). The neural network is a non-linear statistical model that inspired from information transfer in the human brain. The smallest unit of transfer information is called "neuron". Each neuron has its own weight and bias. In the neural network model, each layer consists of many interconnected neurons. The intelligence will feed forward in neural network layer by layer, and the prediction errors are calculated in the output layer. Then, errors are back-propagated to updates weights and bias of neurons so that the important information is learned in the network. The artificial neural network model might have a hidden layer or multiple hidden layers. In this work, we collected corporate email data that were marked with security labels to train an artificial neural network model. After the training process, unseen input data is fed forward to obtain the prediction result.

3. Email security classification system

In this section, we introduce the proposed email security classification system. The system flow is illustrated in Fig. 2. Classifying the text of emails involves parsing the body of emails as well as the attachments. Preprocessing involves word segmentation, removing stop words, and stemming. After having clean data, we combine words into bi-grams and train an email representation in order to extract features from the text. These features must represent the contents of emails to ensure that the classifier is able to derive the semantic meaning of emails. After obtaining the features of emails, we deal with imbalanced data using under-sampling and determine the security levels by Artificial Neural Network.

3.1. Extracting textual contents of emails

To accelerate email parsing and preprocessing, we implemented a distributed system on Spark and Hadoop. The eml files are first stored in HDFS. To parse all of the text data in the email, including the attachments, requires the loading of eml files from HDFS followed by the partitioning of jobs for each executor in the worker node. Each worker node then parses and stores the eml body and attachment contents in a distributed database, mongoDB (MongoDB, 2018), in parallel.

3.2. Preprocessing data

After all the textual contents are retrieved from eml files, workers receive a mongoDB dataframe, where each data point pertains to one email message containing the main body, the subject, and the security label. In this phase of the process, we segment the sentence, stem the words, and remove stop words. We then combine a consecutive series of two-word entities into a bi-gram, as shown in the example below.

The sentence is:

Amy likes to eat steak.

Using bi-gram to cut the sentence, we obtain the following:

Amy likes, likes to, to eat, eat steak

Combining words into a bi-gram greatly enlarges the email corpus and enhance the capability of representing the semantic meaning. In the representation of an email, we can describe the words in the paragraphs more precisely.

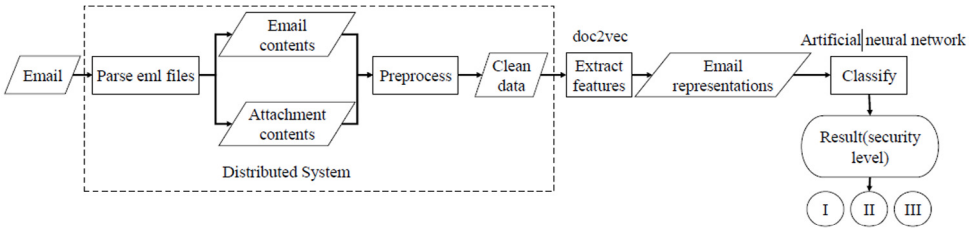

Fig. 2. Flowchart of email security classification.

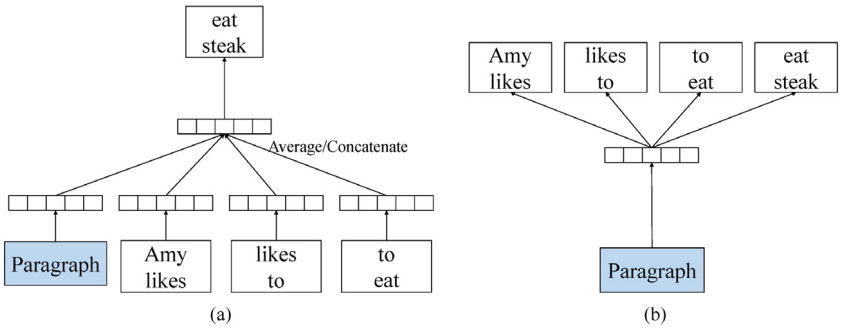

Fig. 3. Learning paragraph vector with bi-gram: (a) bi-PVDM; (b) bi-PVDBOW.

3.3. Feature extraction using paragraph vectors

The concept of paragraph vector is similar to the word vector. Each word in the paragraph is mapped into a unique vector. In the proposed system, we combine consecutive words into bi-grams and map bi-grams into a unique vector. The same word with different neighboring words would have different meanings. The combination of consecutive words will be seen as a new meaningful unit. Each pair of consecutive words will correspond to a unique vector.

To represent an email, we transform the clean data obtained in the previous phase to a high dimensional feature vector using PVDM and PVDBOW models. The input of PVDM are a paragraph vector and bi-grams. The output of PVDM is also in the form of a bi-gram. The combination of bi-grams and PVDM is referred to as bi-PVDM as shown in Fig. 3(a). We also combine PVDBOW with bi-grams, where the input is a paragraph vector used to predict a pair of words randomly sampled from the text. The outputs are bi-grams randomly sampled in the given paragraph. The model is called bi-PVDBOW as shown in Fig. 3(b).

In the original PVDBOW, predictions only consider the appearances of single words in the text, but disregard the order in which they appear. In bi-PVDBOW, the words in a pair must be in the correct order. Therefore, bi-PVDBOW enlarges the training corpus and includes more information of the order of words via bi-grams. Therefore, bi-PVDBOW can represent more semantic meanings. The experimental results show that bi-PVDBOW is more effective than bi-PVDM, original PVDBOW, and original PVDM.

3.4. Email security level classification

In the proposed system, an artificial neural network is used as the classifier that undergoes training to predict the security level of emails. The power of neural networks depends on the amount of available data. However, classification imposes a number of problems when dealing with real-world datasets. For example, not all datasets are balanced. Without appropriate process of data prior to training, the resulting classifier will tend to favor classes with a larger data sizes.

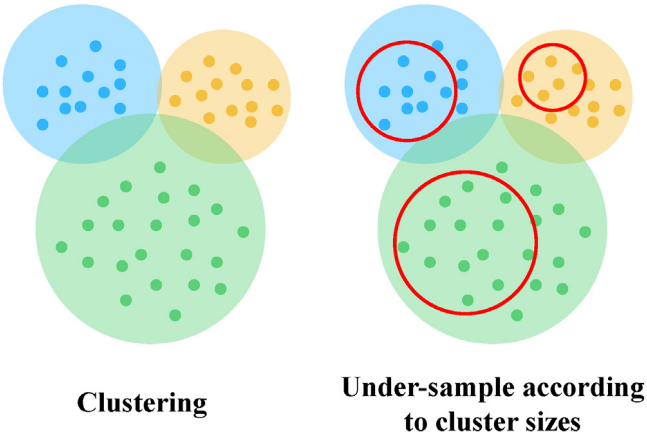

Fig. 4. Under-sampling with clustering.

3.4.1. Under-sampling using K-means clustering

K-means clustering (Hartigan and Wong, 1979) is an unsupervised method used to partition observation data into K clusters. K-means clustering involves minimizing the distance between each node in a cluster and the mean of the cluster. After K-means clustering, we can partition majority class into K parts. Emails with similar semantic meaning are closely distributed in the vector space due to the meanings represented in the paragraph vectors. As shown in Fig. 4, similar emails are grouped within the same clusters. Then we under-sample the majority class from each cluster according to the cluster size until the amounts of the data in the majority class similar to the amounts in the minority class. The number of data sampled in each cluster, N_k, is proportion to the size of the cluster, as calculated by Eq. (1). T is the target amount we want after under-sampling. N is the total number of the data in the majority class. Partition majority into K clusters and n_k is the number of the k_{th} cluster.

$$N_k = T \cdot \frac{n_k}{N} \tag{1}$$

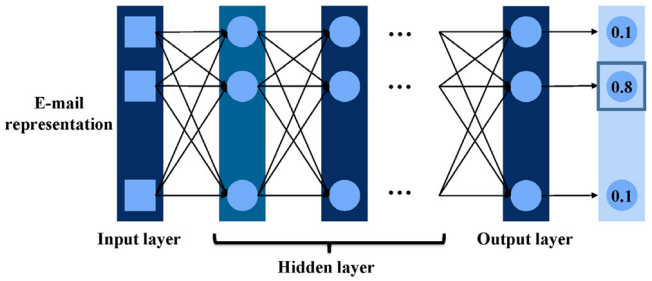

Fig. 5. Classification of emails using artificial neural network.

3.4.2. Training email classifiers by artificial neural network

Under-sampling makes it possible to balance majority class and minority class. The artificial neural network is then used as our classifier. The input of the neural network is the representation of the email, and the output layer has multiple units respectively corresponding to different security labels. Fig. 5 illustrates the process of using artificial neural network to predict the security level of an email. We input email representations and calculate the weights between each layer. We then select the largest output value as the final prediction.

The neural network learning process includes two phases, say a forward phase and a backward phase. In the forward phase, we feed the document vector, the features extracted in the previous step, and calculate the output. The backward phase involves comparing the differences between the output value o_i and the correct values y_i. Errors are calculated by Eq. (2) to adjust the neural network model. We then propagate the errors to the prior layer to update the weights using stochastic gradient descent. By partial differentiation to get the slope of weight, we can calculate the update of the weight using Eq. (3). η is the learning rate of the neural network, which can control the step size of each gradient descent. The weight is updated using Eq. (4). Classifier training involves updating the weight through several iterations, n. After training phrase, we can use the neural network to classify the emails.

$$E_{total} = \sum_{example\ x} \frac{1}{2} \sum_{i \in Output} (y_i - o_i)^2 \tag{2}$$

$$\Delta w_{ji}^{(n+1)} = w_{ji}^{(n)} + \eta \frac{\partial E_{total}}{\partial w_{ji}} \tag{3}$$

$$w_{ji}^{(n+1)} = w_{ji}^{(n)} + \Delta w_{ji}^{(n+1)} \tag{4}$$

4. Experiments

In this section, we present a series of experiments aimed at classifying email compiled by a real company.

4.1. Experiment setup

4.1.1. Data collection

Real email data are provided by a world-leading enterprise in Taiwan. The email data are collected from three departments in the company during December, 2016 to March, 2017. The emails are divided into three security level as follows: security level I, security level II and security level III. The security level I has the highest confidentiality. The security level is labeled in the subject line as well as in the main body. In some instances, the security labels in the subject line and main body do not match. For example, an email designated security level II includes a security level I attachment, such that it should be assigned to the higher security level, security level I. In some cases, it is observed that the definitions of the security levels differ from different departments. To ensure consistency in the training data, we filter out emails with discrepancies in the security designations. The dataset includes approximately 150 thousand emails. Table 1 shows the amounts of data in each security level.

Table 1
Email data security distribution after filtering.

Security level I	Security level II	Security level III
715	21132	136911

4.1.2. Network settings

To ensure a fair comparison, we limit the number of units in each hidden layer of the neural network to a consistent 300 neurons and use only one hidden layer. In the later experiment, we will investigate the effects of multiple hidden layers. The activation function of the hidden layer is ReLU. We employ Adam (Kingma and Ba, 2015) as the neural network gradient optimizer due to its excellent performance (Ruder, 2016). All training is conducted using the back propagation method based on the neural network devised by Keras (Chollet, 2015).

4.1.3. Evaluation methods

We evaluate various email representations using the same settings for the neural network in order to identify those with superior classification performance. The main purpose of the proposed system is to achieve accurate predictions for each security level of emails. To obtain an accurate indication of the performance of each method, we separately calculate true positive rates and F1-scores pertaining to each security level.

$$True\ positive\ (Class_n) = \frac{Correct\ prediction}{Total\ emails\ with\ Class_n}$$

$$Precision\ (Class_n) = \frac{Correct\ prediction}{Emails\ be\ predicted\ to\ Class_n}$$

$$F1 - score = \frac{Precision * Recall * 2}{Precision + Recall}$$

The true positive rate is the same as the recall in the F1-score equation. The previously assign security designations of the emails are used to judge the effectiveness. We use ten-fold cross validation in the classification phase to ensure that the model is not over-fitting to specific data, i.e., 90% of the data are used to train the classifier model and 10% are used for validation. The values shown in the following experiments are the average of ten-fold cross validation.

4.2. Classification results using different email representations

The effectiveness of representations is the primary factor influencing the classification accuracy. The experiments are meant to reveal the performance of each representation method when applied to different dimensions. We also compare the performance of bi-gram and uni-gram for each representation. All of the experiments in this section are based on the under-sampling of email data designated on the data of security level III.

4.2.1. Bi-BoW versus Uni-BoW

We use Bag-of-Words as the base line of the email representation and select the top frequent words by TF–IDF. The results in Tables 2–5 show uni-gram Bag-of-Words, Uni-BoW, can reach high true positive rate. One of the reasons is that there are few words in some emails. Therefore, using higher dimensions can represent more words in the emails. If we combine words into bi-grams, the words corpus would become much larger than origin. Therefore, the bi-gram Bag-of-Words, Bi-BoW, representations cannot contain enough words information in the limited dimensions. The performance of Bi-BoW representations is worse than Uni-BoW in the same dimensions.

Table 2
Uni-BoW true positive results for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.7916	0.83565	0.85838
150	0.85734	0.86248	0.88259
200	0.86433	0.88458	0.89373
250	0.88951	0.89049	0.8981
300	0.8937	0.89688	0.90433
350	0.8979	0.90133	0.90747

Table 3
Bi-BoW true positive results for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.14405	0.80588	0.44381
150	0.15804	0.81984	0.47468
200	0.44755	0.65365	0.6891
250	0.4923	0.62909	0.75515
300	0.50349	0.79949	0.61254
350	0.51888	0.67428	0.74823

Table 4
Uni-BoW F1-score for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.85562	0.84442	0.8475
150	0.89228	0.87086	0.87307
200	0.9035	0.8884	0.8887
250	0.91774	0.8937	0.89407
300	0.91678	0.90005	0.90004
350	0.91714	0.90424	0.904

Table 5
Bi-BoW F1-score for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.24177	0.67683	0.53948
150	0.26968	0.69158	0.57315
200	0.56737	0.66209	0.6757
250	0.60899	0.66882	0.70847
300	0.62499	0.72772	0.67478
350	0.63581	0.6977	0.71962

Table 6
Uni-LDA true positive results for each dimension.

Topics	Security level I	Security level II	Security level III
20	0	0.77844	0.35723
40	0	0.67669	0.52148
60	0	0.6836	0.63888
80	0	0.56591	0.67479
100	0.07832	0.61035	0.64247

4.2.2. Bi-LDA versus Uni-LDA

Bi-LDA takes the LDA topic-word distribution and reforms the distribution as a series of word pairs from the original uni-gram in LDA. The generative process is the same as that used for original LDA. LDA makes it difficult to retrieve any information other than the topics from emails. From our experimental results, we determine that topical information alone is insufficient for the prediction of security levels. As shown in Tables 6 and 8, the original LDA generates errors in the prediction of security level I. In Tables 7 and 9, we can find the performance of Bi-LDA is superior to the uni-LDA. However, it is unable to identify security level I emails accurately, either.

4.2.3. Bi-PVDM versus Uni-PVDM

We compare the classification results obtained by paragraph vectors with two types of architecture. We first compare the PVDM with bi-grams and the uni-gram. The true positive rates are shown in Tables 10 and 11. The F1-scores are shown in Tables 12 and 13. These results clearly illustrate the superiority of Bi-PVDM over the original PVDM, Uni-PVDM. The true positive rates of Bi-PVDM scoring 30% higher than

Table 7
Bi-LDA true positive results for each dimension.

Topics	Security level I	Security level II	Security level III
20	0	0.67944	0.63917
40	0	0.71133	0.73515
60	0.28251	0.73163	0.74087
80	0.35804	0.76476	0.76171
100	0.20419	0.74787	0.76437

Table 8
Uni-LDA F1-score for each dimension.

Topics	Security level I	Security level II	Security level III
20	0	0.63529	0.45145
40	0	0.62303	0.56031
60	0	0.66376	0.6473
80	0	0.58511	0.63303
100	0.14141	0.61394	0.62941

Table 9
Bi-LDA F1-score for each dimension.

Topics	Security level I	Security level II	Security level III
20	0	0.66048	0.64691
40	0	0.71056	0.72375
60	0.42126	0.7286	0.73578
80	0.51148	0.76058	0.75821
100	0.3337	0.74675	0.75562

Table 10
Uni-PVDM true positive results for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.54125	0.70991	0.71985
150	0.46293	0.71909	0.71327
200	0.50909	0.71105	0.71855
250	0.50209	0.70741	0.71276
300	0.52447	0.71138	0.71119
350	0.54125	0.7146	0.7212

Table 11
Bi-PVDM true positive results for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.8	0.84298	0.8726
150	0.8	0.85562	0.87283
200	0.80839	0.8638	0.86529
250	0.8	0.85912	0.86353
300	0.81678	0.86475	0.86123
350	0.81538	0.86489	0.86003

Table 12
Uni-PVDM F1-score for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.62519	0.71208	0.71448
150	0.56292	0.71446	0.71362
200	0.59721	0.71184	0.71428
250	0.589	0.70777	0.70893
300	0.60434	0.70965	0.70983
350	0.60753	0.71543	0.71775

Uni-PVDM in classifying security level I emails and 15% higher in classifying security level II and security level III emails. Both experiments reveal an increase in the true positive rates and F1-scores when the paragraph vector reached 350 dimensions.

4.2.4. Bi-PVDBOW versus Uni-PVDBOW

PVDBOW is another paragraph vector architecture in which the order of the words in the document is disregarded. From the experiment results, PVDBOW is better than PVDM for representing email data. This is a clear indication that the order of the words in a paragraph is unimportant in the corporate email data. As shown in Table 15, the

Table 13
Bi-PVDM F1-score for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.85757	0.85459	0.85904
150	0.84365	0.86198	0.865
200	0.85502	0.86351	0.86404
250	0.84992	0.8603	0.86068
300	0.86582	0.86222	0.86219
350	0.85861	0.86187	0.86162

Table 14
Uni-PVDBOW true positive results for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.92727	0.91926	0.91913
150	0.93426	0.92726	0.92253
200	0.94405	0.92968	0.92286
250	0.93706	0.93526	0.92761
300	0.92447	0.93673	0.92694
350	0.93986	0.93933	0.92793

Table 15
Bi-PVDBOW true positive results for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.97062	0.97818	0.97835
150	0.98181	0.97965	0.97735
200	0.97622	0.97846	0.9764
250	0.97762	0.98154	0.97749
300	0.98181	0.98126	0.97735
350	0.97902	0.98069	0.97901

Table 16
Uni-PVDBOW F1-score for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.93577	0.91906	0.91917
150	0.94018	0.92498	0.92467
200	0.93684	0.92663	0.92619
250	0.93771	0.93186	0.93109
300	0.92641	0.93229	0.93139
350	0.93593	0.93424	0.93321

Table 17
Bi-PVDBOW F1-score for each dimension.

Dimensions	Security level I	Security level II	Security level III
100	0.97814	0.97822	0.97823
150	0.98044	0.97856	0.9786
200	0.97827	0.97749	0.97739
250	0.97899	0.97966	0.97944
300	0.98112	0.97945	0.97927
350	0.97833	0.97994	0.97987

true positive rates of each class approached 97% and all the results are better than the original PVDBOW in Table 14. The true positive results reveal that Bi-PVDBOW is a better way to represent email data than the Uni-PVDBOW, which shows that the order of words in bi-grams is important. Combining bi-grams with PVDBOW is the best solution to represent emails. (See Tables 16 and 17.)

Fig. 6 presents a comparison of various approaches of the representations of emails. In this comparison, the dimensions of each representation method is set the same as 100. Bi-PVDBOW is shown to have the best performance, which also means that Bi-PVDBOW is best able to extract features from email data. Bi-PVDBOW performed very well in different three security level of emails. In the tests, the LDA model is proved too complicated for the representation of short text. As shown in Table 18, a large number of emails in the sample contain fewer than 100 words, i.e., they were deemed short. Due to the wide range of text lengths among emails, it is inevitable that many are too short for LDA. This problem is easily to be solved using the paragraph vector. The results show that Bi-PVDBOW is better than Uni-PVDBOW in all of four metrics.

Table 18
Email word count distribution.

Words count	<100	100~200	200~300	300~400	400~500	>500
email count	58108	31470	17004	10988	8125	33063

Table 19
True positive rates obtained without under-sampling.

Method	Security level I	Security level II	Security level III
Uni-BoW/100D	0.7972	0.68611	0.9776
Uni-LDA/100 topics	0.08111	0.09114	0.99464
Uni-PVDM/100D	0.43076	0.40904	0.94738
Uni-PVDBOW/100D	0.91188	0.8249	0.9783
Bi-BoW/100D	0.12867	0.10377	0.99512
Bi-LDA/100 topics	0.35524	0.2915	0.98167
Bi-PVDM/100D	0.77202	0.69605	0.97355
Bi-PVDBOW/100D	0.97902	0.94775	0.99409

4.2.5. Explore the effects of N-grams

According to the previous subsections, we find that PVDBOW is the best semantic representation method. To understand the effects of n-gram, we compare the performance of uni-grams to four-grams on PVDBOW by Recall, Precision, and F-Measure as shown in Fig. 7. The results of F-Measure show that Bi-PVDBOW outperforms Uni-PVDBOW, Tri-PVDBOW, and Four-PVDBOW. In addition, the results of Recall and Precision show that Bi-PVDBOW has more stable and better performance than others. Overall, Bi-PVDBOW is the best semantic representation setting for the proposed method. In addition, many previous approaches (Johnson et al., 2006; Bin et al., 2013; Jiang et al., 2013; Wu et al., 2015; Yoo et al., 2017) achieved excellent performance by adopting bi-gram to extract features from texts. Therefore, our findings are consistent with previous approaches.

4.3. Validate with all security level III email

Before training the classifier, we under-sample the security level III email data. In this experiment, we collect the rest of security level III data after under-sampling and validate the model with these data in order to see if the under-sampling method can well represent the whole class. The input dimension of each method is 100. The experimental results are shown in Fig. 8. Comparing with Fig. 6, the true positive rates of each model varies a little, which can be ignored. In our under-sampling method with clustering, the sampling data can represent the whole security level III email data well.

4.4. Validation without Under-sampling

We seek to elucidate the influence of imbalanced data by training the classifier without under-sampling. In the experiment results shown in Table 19, this leads to an increase in accuracy when predicting security level III emails due to the classifier leaning toward security level III data. Nonetheless, without under-sampling, the performance of most methods dropped considerably when classifying security level I and security level II emails. It is interesting to note that Bi-PVDBOW still achieved 95% true positive rates in each class. We indicate that the differences between each class are big enough for the classifier to recognize them. When features are represented effectively, classifiers are able to learn even from imbalanced data. (See Table 20.)

4.5. Performance comparison: Varying the number of layers

We compare the results obtained when varying the number of hidden layers in the neural network. In an artificial neural network, the hidden layer can be viewed as the features of the data that must be learned. Generally, increasing the number of parameters in a neural network renders a greater number of features to be learned. Fig. 9 shows that an increase in the number of hidden layers does not enhance the prediction significantly, due to the fact that the representation of the email is good enough to allow learning within the first layer.

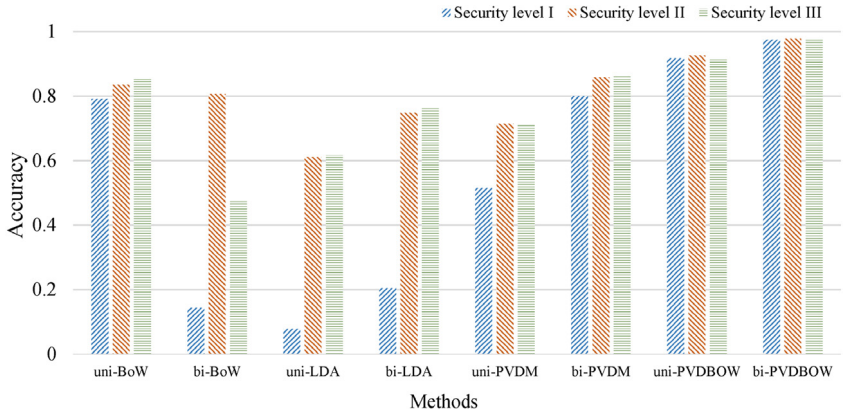

Fig. 6. Comparison of various email representations.

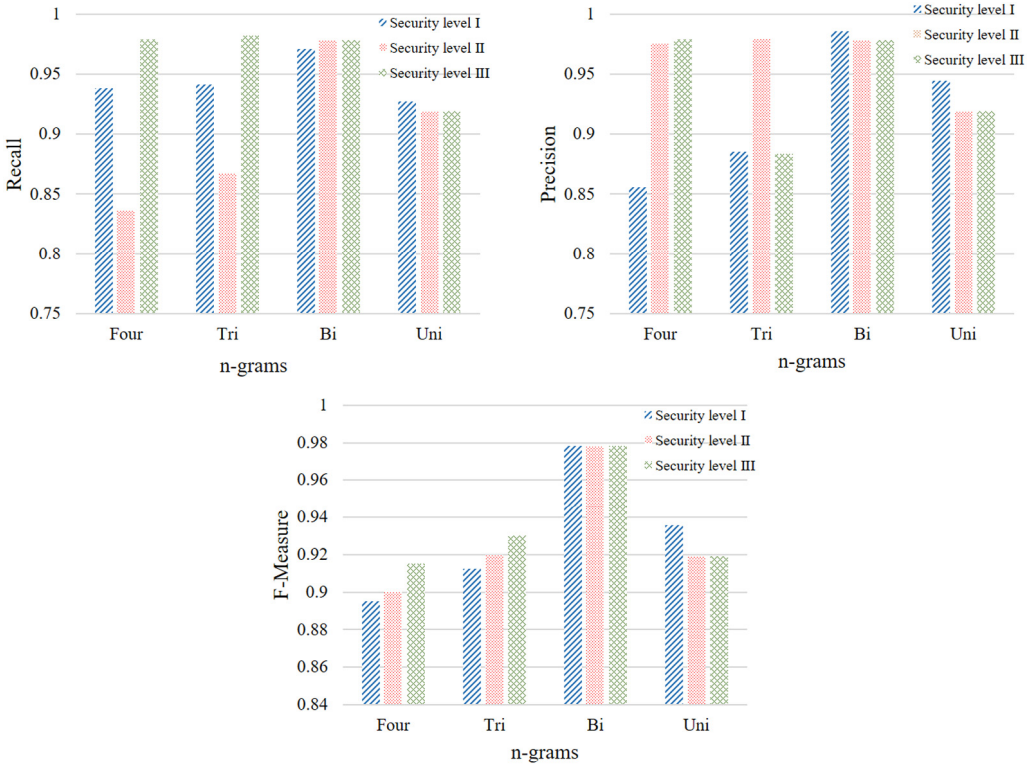

Fig. 7. Performance comparison of uni-gram to four-gram on PVDBOW.

Table 20
True positive rates obtained using under-sampling: security level III email.

Method	Security level I	Security level II	Security level III
Uni-BoW/100D	0.7916	0.83565	0.85838
Uni-LDA/100 topics	0.07832	0.61035	0.64247
Uni-PVDM/100D	0.51468	0.71379	0.71136
Uni-PVDBOW/100D	0.91748	0.92579	0.91263
Bi-BoW/100D	0.14405	0.80588	0.44381
Bi-LDA/100 topics	0.20419	0.74787	0.76437
Bi-PVDM/100D	0.8	0.85822	0.85963
Bi-PVDBOW/100D	0.97482	0.97846	0.97741

4.6. Performance validation using a combination of data segmentation methods

We also evaluate a combination of uni-gram and bi-gram segmented words. The results are presented in Fig. 10. In the combined method of PVDBOW, the training data includes uni-gram as well as bi-gram content. From the result, there is only a slight increase in true positive rates when predicting security level I emails. Therefore, it appears that bi-PVDBOW provides sufficient classification performance, and that bi-gram is the best email representation method.

4.7. One versus all classification method comparison

Besides the three classes classification methods, we also use one versus all classification method to predict the security level of the emails. In this method, we first classify the emails into two parts, the emails labeled with security level I and not security level I. Then we classify the emails which not belonged to security level I in the previous step into security level II and security level III. The results shown in Fig. 11 do not have much difference from the results of multi-classification because the email representation is the same as the previous experiments. The neural network can still learn the knowledge from the representation.

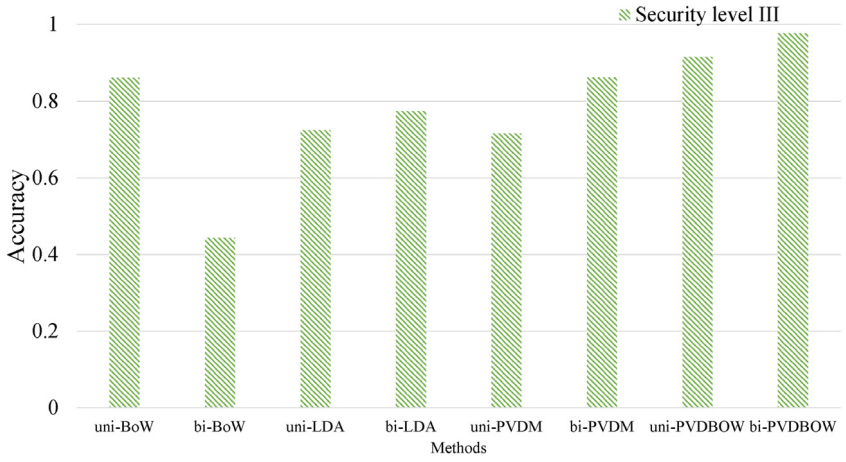

Fig. 8. Validate with all security level III data.

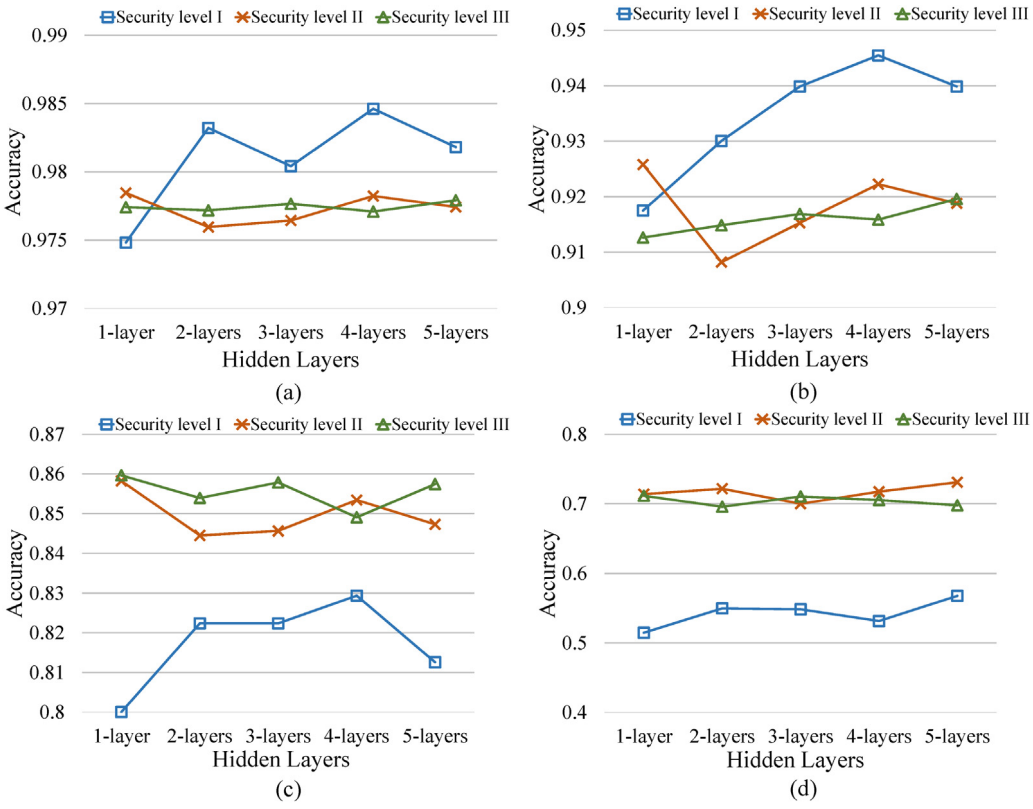

Fig. 9. Different layers of classifier. (a) Bi-PVDBOW (b) Uni-PVDBOW (c) Bi-PVDM (d) Uni-PVDM.

4.8. Comparison of other imbalanced data preprocessing methods

To understand the effects of using different preprocessing methods for imbalanced data, we compare with two methods, Han et al. (2005) for oversampling and Zhou and Liu (2006) for a cost-sensitive classifier. To have a baseline in the comparison, we include the raw data without imbalanced data preprocessing, None, in the comparison. In our approach, we use K-means clustering as an undersampling method. Above preprocessing methods are applied in the proposed ANN classifier.

Table 21 shows the accuracy of four methods. Our ANN classifier with K-means undersampling outperforms others in Security level I and II but has a little bit lower accuracy than Zhou et al. and None in Security level III. Nevertheless, using the raw data without imbalanced data preprocessing is better than Zhou et al. The result shows that we can achieve the best performance in Security level III without imbalanced data preprocessing. However, we care more about the classification performance of confidential classes, Security level I and

Table 21
Comparison of other imbalanced data preprocessing methods.

Label	Security level I	Security level II	Security level III
None	0.941	0.909	0.993
K-means	0.975	0.978	0.977
Han et al.	0.946	0.932	0.951
Zhou et al.	0.937	0.916	0.992

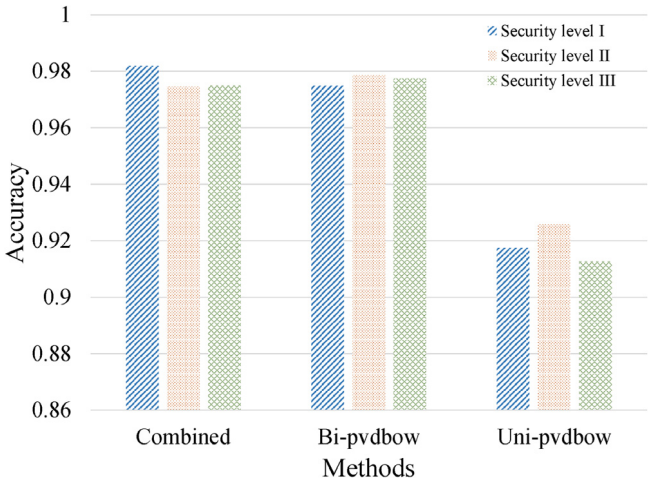

Fig. 10. Results obtained through combination of uni-gram and bi-gram.

Table 22
Classification accuracy validation by corporation.

	Security level I	Security level II	Security level III
Part A	0.837(2084/2491)	0.933(67865/72666)	0.916(52044/56843)
Part B	0.841(2094/2491)	0.966(70225/72666)	0.949(26931/29475)
Part C	0.839(2090/2491)	0.944(68599/72666)	0.931(53984/57961)

II. Classifying data in higher confidential levels to lower confidential levels is hazardous. Therefore, the proposed ANN classifier using K-means undersampling method is more ideal for our research problems.

4.9. Classification validation by corporation

After we design the model, the corporation validates the system with their own email data. Their data contain all factory email from March, 2017 to May, 2017. They depart email data into three parts and only use one month of security level III email data because of their limitation of hardware. The validation result is shown in Table 22. The true positive rates of each security level decrease a little because the diversity of email is bigger than our data. However, the results of the system are still great enough to be used online in the company.

Table 23
Average time of each phase when predicting an email.

Process	Time (seconds)
Textual content extraction	0.0403
Preprocessing data	0.0315
Email representation retrieval	0.0034
Security level prediction	0.00058
Total	0.0758

Table 23 shows the process time of an email. The average processing time of predicting an email is about 0.0758 s. Textual data extraction phase costs 0.0403 s and only needs 0.0315 s when preprocessing an email. It only needs 0.0034 s to retrieve email representation and 0.00058 s to predict security level through neural network. In the online predicting phase there is no need to use distributed system because the processing time is short enough.

5. Conclusions

This study presented an effective email security level classification framework in which sentences are segmented into bi-grams prior to training paragraph vectors. This approach helped to increase the size of the vocabulary set and includes the information of the order of the words. In experiments, bi-PVDBOW outperformed all other methods with regard to true positive rates and F1-scores. The proposed method also used K-means clustering to undersample the majority class that would otherwise be over-represented. The aim was to prevent the classifier from favoring the class with a larger dataset. The results also demonstrate the importance of deriving a good representation of email when dealing with the problem of data imbalance. The proposed email security level classification system proved highly accurate, while the response time remains very short. This makes it a real online system for the detection of sensitive information in corporate outbound emails. The significant contributions are that the proposed system can benefit the enterprises to protect the sensitive information in real time and reduce the human efforts of managers

Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no. MOST 105-2221-E-006-212-MY2.

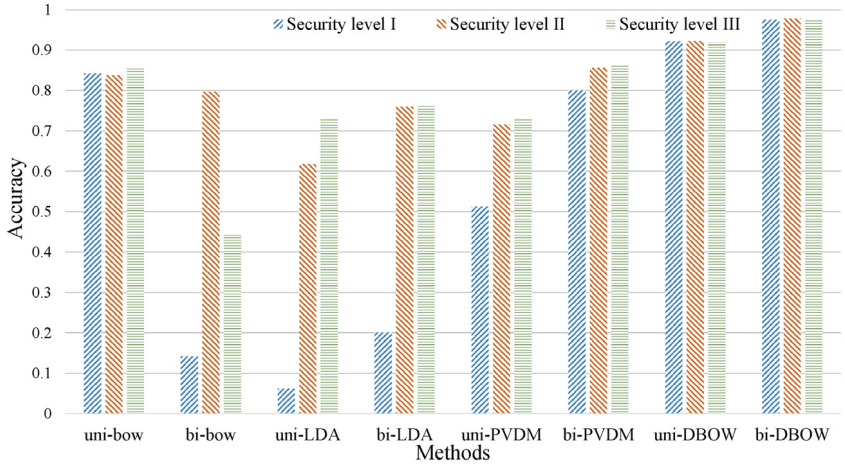

Fig. 11. Validate with one versus all classification method.

References

Alparslan, E., Bahsi, H., 2009. Security level classification of confidential documents written in Turkish. In: International Conference on User Centric Media. Springer, pp. 329–334.

Alparslan, E., Karahoca, A., Bahsi, H., 2013. Security-level classification for confidential documents by using adaptive neuro-fuzzy inference systems. Expert Syst. 30 (3), 233–242.

Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155.

Bin, Y., Cunlin, P., Dan, L., 2013. Chinese text feature extraction method based on bi-gram. In: Communications, Circuits and Systems, ICCCAS, 2013 International Conference on, vol. 2, IEEE, pp. 342–346.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3 (Jan), 993–1022.

Chollet, F., 2015. Keras, https://github.com/fchollet/keras.

Han, H., Wang, W.-Y., Mao, B.-H., 2005. Borderline-smote: A new over-sampling method in imbalanced data sets learning. Adv. Intell. Comput. 878–887.

Han, J., Kamber, M., 2006. Data Mining Concept and Techniques, second ed.

Hartigan, J.A., Wong, M.A., 1979. Algorithm as 136: A K-means clustering algorithm. J. R. Stat. Soc. Ser. C. Appl. Stat. 28 (1), 100–108.

Hecht-Nielsen, R., 1988. Theory of the backpropagation neural network. Neural Netw. 1 (Supplement-1), 445–448.

Ismaila, I., Ali, S., Ngoc, T.N., Sigeru, O., Ondrej, K., Kamil, K., Marek, P., 2015. A combined negative selection algorithm particle swarm optimization for an email spam detection system. Eng. Appl. Artif. Intell. 39, 33–44.

Jain, A.K., Yu, B., 1998. Document representation and its application to page decomposition. IEEE Trans. Pattern Anal. Mach. Intell. 20 (3), 294–308.

Jiang, S., Wang, X., Zhu, H., 2013. Learning pairwise comparisons of items with bi-gram content features for recommending. In: Computer Science and Network Technology, ICCSNT, 2013 3rd International Conference on. IEEE, pp. 446–449.

Joachims, T., 1998. Text categorization with support vector machines: learning with many relevant features. In: European Conference on Machine Learning. Springer, pp. 137–142.

Johnson, D., Malhotra, V., Vamplew, P., 2006. More effective web search using bi-grams and tri-grams. Webology 3 (4), 35.

Kingma, D., Ba, J., Adam: A method for stochastic optimization, 2015, arXiv.

Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning, ICML-14, pp. 1188–1196.

Mikolov, T., Statistical language models based on neural networks, Presentation at Google, Mountain View, 2nd April, 2012.

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013b. Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. 3111–3119.

MongoDB, 2018, URL https://docs.mongodb.com/.

Ruder, S., An overview of gradient descent optimization algorithms, 2016, arXiv.

Safavian, S.R., Landgrebe, D., 1991. A survey of decision tree classifier methodology. IEEE Trans. Syst. Man Cybern. 21 (3), 660–674.

Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24 (5), 513–523.

Shakir, M., Abubakar, A., Yousoff, Y., Waseem, M., Al-Emran, M., 2016. Model of security level classification for data in hybrid cloud computing. J. Theor. Appl. Inf. Technol. 94 (1), 133.

Snchez, D., Batet, M., 2017. Toward sensitive document release with privacy guarantees. Eng. Appl. Artif. Intell. 59, 23–34.

Suykens, J.A., Vandewalle, J., 1999. Least squares support vector machine classifiers. Neural Process. Lett. 9 (3), 293–300.

White, T., 2012. Hadoop: The Definitive Guide.

Wu, Y., Heng, Z., Xu, B., 2015. Tr-lda: a cascaded key-bi-gram extractor for microblog summarization. Int. J. Mach. Learn. Comput. 5 (3), 172–178.

Yoo, D., Ko, Y., Seo, J., 2017. Speech-act classification using a convolutional neural network based on pos tag and dependency-relation bi-gram embedding. IEICE Trans. Inf. Sys. 100 (12), 3081–3084.

Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I., 2010. Spark: cluster computing with working sets. HotCloud 10 (10–10), 95.

Zhang, J., Mani, I., 2003. knn approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of Workshop on Learning from Imbalanced Datasets, vol. 126.

Zhang, Y.-P., Zhang, L.-N., Wang, Y.-C., 2010. Cluster-based majority undersampling approaches for class imbalance learning. In: Information and Financial Engineering, ICIFE, 2010 2nd IEEE International Conference on. IEEE, pp. 400–404.

Zhou, Z.-H., Liu, X.-Y., 2006. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans. Knowl. Data Eng. 18 (1), 63–77.