2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress

Malicious Mail Filtering and Tracing System Based on KNN and Improved LSTM Algorithm

Da Xiao
School of Cyberspace Security
Beijing University of Posts and Telecommunications
Beijing, China
xiaoda99@bupt.edu.cn

Meiyi Jiang✉
School of Cyberspace Security
Beijing University of Posts and Telecommunications
Beijing, China
heqi867610481@bupt.edu.cn

***Abstract*—Spam and phishing emails are very troublesome problems for mailbox users. Many enterprises, departments and individuals are harmed by them. Moreover, the senders of these malicious emails are in a hidden position and occupy an initiative position. The existing mailbox services can only filter and shield some malicious mails, which is difficult to reverse the disadvantage of users. To solve these problems, we propose a secure mail system using k-nearest neighbor(KNN) algorithm and improved long short-term memory(LSTM) algorithm(Bi-LSTM-Attention algorithm). KNN classifier can effectively distinguish normal emails, spam and phishing emails, and has a high accuracy. Bi-LSTM-Attention classifier classifies phishing emails according to the similarity of the malicious mail text from the same attacker to some extent. By classifying and identifying the source of malicious emails, we can grasp the characteristics of the attacker, provide materials for further research, and improve the passive status of users. Experiments show that the classification results of attack sources reach 90%, which indicate the value of further research and promotion.**

***Keywords*-phishing mails; KNN; improved LSTM algorithm; classifier;**

I. Introduction

Spam emails consume a lot of bandwidth, waste resources, infringe the privacy and mailbox space of the recipient, and consume the time, energy and money of the recipient [1]. Phishing email not only brings a lot of troubles to people, but also is more likely to cheat people and produce economic losses. No matter for individuals, enterprises, government departments or other institutions, phishing email is a common means of attack. The existing email filtering system is effective in dealing with spam, but when dealing with phishing email, it often fails due to the similarity between phishing emails and normal emails. In addition, mailbox users have been in a passive position compared with attackers. In the face of mail filtering mechanism, attackers can easily bypass by changing IP and continue to spread spam or phishing emails. [2] However, a large number of cases show that malicious mails from the same source often have certain similarity in many headers and body features. Therefore, we can use this similarity to classify the malicious emails according to the attack sources, so as to describe the characteristics of the attackers, and provide effective feedback information such as the number of attack organizations and whether there are new attack sources. To sum up, users urgently need a secure email system, which can not only accurately identify phishing mail and spam, but also identify malicious mails from the same source, and change the passive status of users.

This paper describes an email filtering system that can effectively deal with phishing email attacks by machine learning and deep learning methods. The system can not only distinguish the malicious emails (i.e. spam and phishing emails) from the normal emails, but also distinguish the source of phishing emails according to the similarity between the mails from the same attacker. KNN classifier and Bi-LSTM-Attention classifier are mainly used in the system.

In this system, KNN classifier is used to distinguish normal mails, spam mails and phishing mails. This process includes two steps. Firstly, the phishing mails are detected from the data set . Secondly, the term frequency–inverse document frequency(TF-IDF) value of the words in the body of the mail is calculated by using the method of text classification to distinguish the spam mails from the normal mails.

Long short-term memory(LSTM) algorithm is very suitable for modeling temporal data, such as text data, because of its design characteristics. Bi-directional long short-term memory(Bi-LSTM) is a combination of forward LSTM and backward LSTM. Bi-LSTM-Attention is to add an attention layer to the model of Bi-LSTM. In Bi-LSTM-Attention algorithm, the weight of each time sequence is calculated first, then the vectors of all time sequences are weighted together as feature vectors, and then the softmax classification is carried out. In this system, the text of email is processed by Bi-LSTM-Attention classifier. Phishing emails are further classified according to the source of attack.

In the Section 2, we introduce the previous research results on email filtering methods, machine learning and deep learning related algorithms. In the Section 3, first of all, we show some examples of data set for LSTM classifier, compare the similarity of emails from the same source of attack, and explain the principle and feasibility of the module

978-1-7281-6609-4/20/$31.00 ©2020 IEEE
DOI 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00048

intuitively. Then we describe all aspects of the system design, including the selection and acquisition process of data sets, the workflow of the system and the principles of KNN and Bi-LSTM-Attention algorithm. In the Section 4, we focus on how the original sample is transformed into the feature vector which can be input to the classifier. In the Section 5, we introduce the evaluation indexes and results of classification effect.

The contributions of this paper are summarized as follows:

- We design a complete system including filtering phishing email, filtering spam, and classifying phishing emails by attack source. The KNN algorithm and the improved LSTM algorithm is applied in the system, and the satisfactory accuracy is obtained in the experiment. Existing results often focus on only filtering spam and phishing email [3], while our further study of phishing mail is relatively novel.
- This paper proposes the possibility of using the similarity between mails from the same source to trace the source of mails, which is of great significance when faced with more dangerous and targeted attacks, such as the advanced persistent threat faced by important government departments. Besides, our method can grasp the characteristics of attackers, provide materials for further research, and improve the passive status of users.
- We overcome the difficulty of small phishing email datasets. First of all, phishing email dataset itself is difficult to obtain, compared with ordinary spam. We implement the effective expansion of the data set through a data augmentation method combined with manual imitation. For most of the problems related to phishing mails, datasets have specific sources that are inconvenient to be disclosed. We point out a way to utilize the small public datasets.

II. Related Work

In 2006, Ian fette of Carnegie Mellon University and others first proposed a phishing email detection method based on machine learning [4], which mainly focuses on links and scripts. The commonly used classification algorithms of phishing email detection based on machine learning are k-nearest neighbor, support vector machine [5] and random forests [6].

Phishing email is closely related to social engineering [7], [8]. The detection methods of phishing email based on machine learning mostly use the general shallow machine learning classification algorithm, so the improvement of the detection methods of phishing email [9] mainly focuses on the improvement of features. On the basis of Ian fette, some researchers have put forward improved methods, which expand the existing features and also put forward new features. So far, the features used in phishing email detection are mainly divided into four categories: mail header feature, mail body feature, mail link feature and mail script feature. The existing detection of phishing email often focuses on the analysis of a single email, which can not judge the relationship between emails, the source of emails and other further analysis, and can not improve the passive status of victims. In the system designed in this paper, the detection method of phishing email appears as one of the modules.

It has become a common method to apply text classification to email filtering. The enterprise oriented secure e-mail system designed by Asaf Cidon et al. [10] makes use of text classification. They divide the classification problem into two parts: one is to analyze the title of e-mail, the other is to use natural language processing to detect the suspicious links in the phrases or the body of e-mail related to phishing attacks.

LSTM algorithm [11] is widely used in natural language processing. The Bi-LSTM-Attention proposed by Peng Zhou et al. [12] has been proved to be more effective. In our system, we apply Bi-LSTM-Attention classification algorithm to the body of mail to deal with the problem of mail classification by source.

III. System Design

A. Similarity

Figure 1. Example 1 of the similar spam mails.

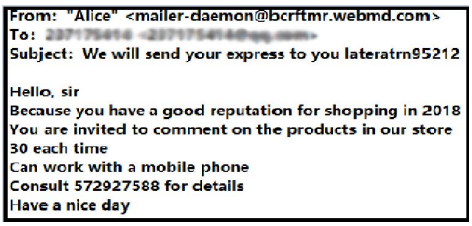

Figure 2. Example 2 of the similar spam mails.

Figures 1 and 2 shows two spam mails in the dataset. It's easy to see that the two emails are closely related and come from the same organization [13], [14]. Although there are subtle differences in the content of the two emails, the meaning and key information such as price and contact information are the same. In addition, the structure of the text is the same, for example, there are blessings at the end. This kind of similarity is the important basis of using the

improved LSTM algorithm to classify and find the common source of sender [15], [16].

For spam, it seems that this work is not so urgent, but for phishing email, it is much more important to identify the common attack source. The phishing email often involves criminal cases such as stealing secrets, high-value fraud, etc., or the next attack is brewing after the virus is implanted. The attackers behind phishing emails are also more cautious. They may try to change some words or expressions when sending phishing emails to different people, so that people don't pay attention to the connection between emails. And the more purposeful and harmful the attacker is, the more carefully he designs the email and hides himself. But how to write a mail is related to one's habits, and the connection between mails is not easy to cover up. We hope that the model described in this paper can make the attacker's efforts to hide the trace invalid according to the similarity that can't be concealed. If we can identify which emails come from the same source in many phishing emails, we have a basic understanding of the characteristics and patterns of the attack organization. On the one hand, we can filter emails targeted, on the other hand, we can lay a foundation for mastering the attack evidence of the attacker.

B. Dataset

Spam and normal mails are from "trec06p", a public English corpus provided by TREC. "Trec06p" includes 37822 emails, 12910 of which are normal emails and 24912 of which are spam emails. Because the dataset of phishing mail is difficult to obtain, and the number is relatively small, in order to maintain the balance of data quantity in the process of KNN classification, we randomly selected 1200 normal mails and 1200 spam mails from the "trec06p" dataset as the experimental dataset.

The dataset of phishing email is provided by Kaggle platform, which is a collection of more than 2,500 "Nigerian" Fraud Letters, dating from 1998 to 2007 [17], Nigerian scammers are an organized criminal network that has been operating practically all over the world since the beginning of the 80's and that hunts with financial fraud associated with money laundering from people. Phishing is a common method used by the organization.

First, we processed the phishing mail dataset. According to the sending time and "return-path" information of the mail, we filtered out the mail from the same person, and selected three types with more data. At this time, there were too few samples to support the process of deep learning. In order to solve this problem, we extended the dataset manually, and imitateed the author to generate new samples according to the samples we have. Finally, 1800 phishing mails are obtained, which are divided into three categories according to the author's differences, including 600 for each category.

C. System

The workflow of the system can be represented by Figure 3. First of all, preprocess the mail sample to separate the body and the header of the mail. Then for the message header, further extract the message ID, date, sender, recipient, subject and other information [18]–[20]. For the mail body, extract the plain text content of the body, the content in the the $\langle a \rangle$ tag and the content in the $\langle script \rangle$ tag.

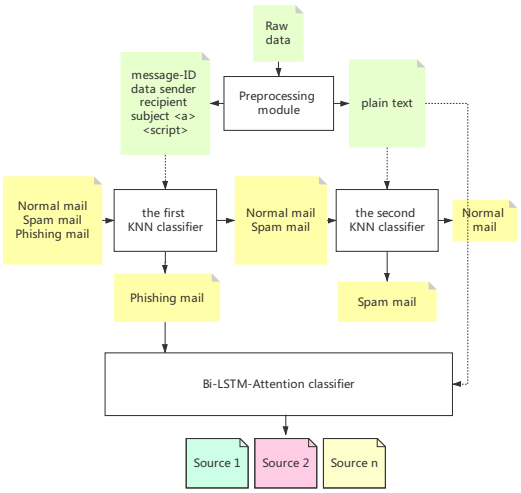

Figure 3. The workflow of the system.

The preprocessed samples are input into KNN classifier for model training. It includes two training processes with different purposes: first, in order to filter phishing mails, three types of samples are marked as phishing mails and non-phishing mails (including spam and normal mails). According to whether there are abnormal fishing links and other features in the text, the training is carried out. Next, in order to distinguish between spam and normal mails, we use the method of text classification to train the model.

In this process, 10% of the normal mail, spam and phishing mail samples were selected as the test set to evaluate the training results.

Next, phishing samples are classified by Bi-LSTM-Attention classifier. In the process, 10% of the samples are also reserved as the test set under each attack source label.

Finally, input the corresponding training set for each process to evaluate the results. The evaluation indexes include accuracy rate, recall rate, etc.

D. Correlation Algorithm

KNN algorithm and Bi-LSTM-Attention algorithm are mainly used in the system. The following briefly describes

the principle and important parameters of the two algorithms.

KNN algorithm. Generally speaking, the idea of KNN algorithm is to determine the data types of samples in a space by using the data types of K known samples closest to the unknown samples. First, calculate the distance between the points in the known category data set and the current point. Then select the k points with the smallest distance from the current point, and determine the occurrence probability of the category of the first k points. At last return the category with the highest occurrence frequency of the first k points as the prediction classification of the current point.

KNN algorithm has two key factors: the measurement of distance and similarity, and the value of K.

Measurement of distance and similarity. The specific steps are described as follows: firstly, according to the above-mentioned extraction features, the feature vector is formed. Then calculate the similarity between the test sample and each sample in the training set. The formula is as follows:

$$Sim(d_i, d_j) = \frac{\sum_{k=1}^{M} W_{ik} \times W_{jk}}{\sqrt{\sum_{k=1}^{M} W_{ik}^2} \sqrt{\sum_{k=1}^{M} W_{jk}^2}} \tag{1}$$

In the formula, d_i is the eigenvector of the i_{th} sample in the test set, d_j is the center vector of the j class in the training set, M represents the dimension of the eigenvector, W_k represents the value of the k dimension of the eigenvector. After calculating the sample similarity, k training set samples with the largest similarity to the test samples are selected. In these k neighbors, the weights of each class are calculated. Then, the weight of each class is compared, and the sample is determined as the class with the largest weight.

The value of k. The value of k has an important influence on the classification results. If the value of k is too small, the classification results are easy to be disturbed, and the influence of noise points will be amplified; if the value of k is too large, more points of other categories may be delimited as near neighbors, so as to intervene the results. There is an empirical rule that the value of k is generally lower than the square root of the number of training samples. It is worth emphasizing that we adopt a reasonable distance weighted average method in the final determination of categories. This kind of weighted average can also effectively reduce the impact of the setting of k value on the results.

Bi-LSTM-Attention algorithm. Bi-LSTM-Attention algorithm is an improved algorithm based on LSTM model. Long Short-Term Memory network (LSTM) is a kind of time recurrent neural network, which can learn long-term dependence. LSTM has three kinds of thresholds: input gate f_t, forgetting gate i_t and output gate o_t. The forgetting gate determines which information is to be discarded from the unit state. x_t represents the input word of t time; c_t represents the cell state; $\tilde{c_t}$ represents the temporary cell state; h_t represents the hidden layer state. The calculation process of LSTM can be summarized as: through forgetting and memorizing the new information in cell state, the useful information can be transferred to the subsequent calculation, while the useless information is discarded, and the hidden layer state will be output at each time step. Among them, forgetting, memory and output are controlled by forgetting gate, memory gate and output gate calculated from the hidden layer state of the last time and the current input. See Figure 4.

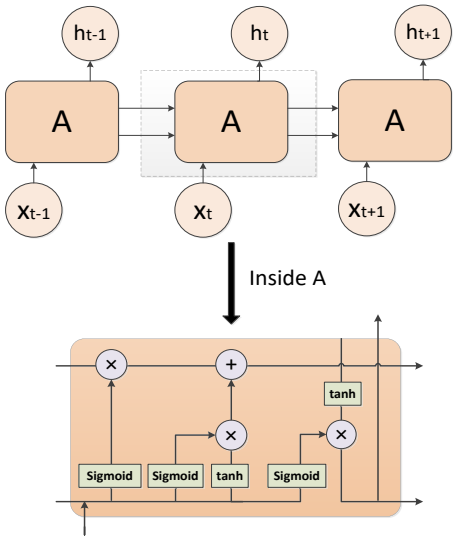

Figure 4. LSTM schematic diagram.

In the field of text classification, LSTM has a wide range of applications and excellent performance. However, there is still a problem for LSTM to model sentences, that is, it cannot encode the information from the back to the front. The forward LSTM is combined with the backward LSTM to form the Bi-LSTM. Bi-LSTM can better capture bi-directional semantic dependency. As shown in Figure 5, a set of vectors is obtained from the forward LSTM input positive order sentences, and a set of vectors is obtained from the backward LSTM input reverse order sentences. Finally, the two sets of vectors are spliced in turn. In this way, the final vector contains all the forward and backward information. In the Bi-LSTM model, the output vector of the last time sequence is used as the feature vector and input to the softmax layer. The process is formulated as follows. h_i represents the output of the i^{th} word.

$$h_i = [{h_i}^R \oplus {h_i}^L] \tag{2}$$

The Bi-LSTM-Attention model is improved on this basis: before output, the attention layer is introduced, the weight of each time sequence (denoted by W) is calculated first, then the vectors of all time sequences are weighted and summed as feature vectors, and then output to the softmax layer

for classification. In the experiment, adding attention did improve the results. Attention model is actually to simulate the attention of human brain. This kind of model can also simulate human brain more realistically. Let H represent a matrix consisting of output vectors$[h_1, h_2, ..., h_T]$ after two bidirectional vectors are combined, where T is the sentence length. The process above is formulated as follows.

$$\alpha = softmax(W^T tanh(H)) \tag{3}$$

$$h^* = tanh(H\alpha^T) \tag{4}$$

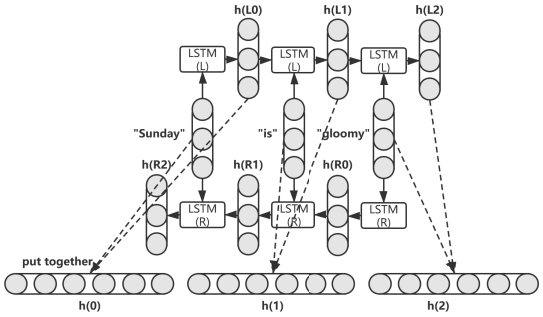

Figure 5. Bi-LSTM schematic diagram.

IV. Feature and Classifier

A. the Feature Selection of Phishing mail Filter

In the task of filtering out phishing mails, feature selection should be able to distinguish phishing mails from normal mails and spam mails. Because the phishing mails are made up by the attackers who imitate the normal mails carefully, it is difficult to distinguish the phishing mails from the normal mails only in semantics. The flaws of a phishing email often appear in the links that lure victims to click, the HTML text and JavaScript code implied in emails, etc. [21] Similarly, we believe that spam mails don't contain malicious code or malicious links that are attractive to click on. Therefore, the feature extraction method applied in this module focuses on the above parts rather than the text and semantics of the email. Based on the above explanation, this module extracts the following 14 features from the mail samples. If the i_{th} feature is expressed as A_i, then the email can use the feature vector $E = \{A_1, A_2, ..., A_{14}\}$.

These features are considered to be effective in determining whether an email is a phishing email in previous experiments. Most of these features are based on links that appear in the body of the message for the recipient to click on. Because the ultimate goal of phishing emails is to let the recipient click on the link, a suspicious link indicates that the email is suspicious.

In addition, if the message contains JavaScript code, it is possible to achieve the theft of user information, the theft of user cookies and so on. Well designed JavaScript sensitive code will not appear in normal mails. So if the message contains JavaScript code, it indicates that the message is suspicious.

Table I
Characteristic distribution of phishing mail

Num.	Features
A_1	Whether the website link is based on IP type
A_2	Whether the website link domain name registration time is too short
A_3	Does the website link contain a strong inductive language
A_4	Is the link domain name the same as the sender's mailbox domain name
A_5	Whether the display string of link domain name and web page in "href" matches
A_6	Whether there are too many separators in the link
A_7	Whether the "HTTP" protocol appears too many times in the link
A_8	Message ID domain name and sender domain name are inconsistent
A_9	Special characters in URL
A_{10}	The URL contains a non-standard port
A_{11}	Whether there are words that users pay more attention to in the subject or content of the message
A_{12}	Whether the message contains JavaScript code
A_{13}	The URL contains a non-standard port
A_{14}	Is the URL using a short address

The details in the email header can also provide a basis for judging whether the email is a phishing email. Message ID is the unique identifier of the mailbox sequence number, which exists in the message header. For example, "tencent_3226b9972e1539bc1e151623@QQ.com" is the message ID of an email sent by Tencent's QQ email administrator. Generally, the sender's email domain name of an enterprise with its own email server is the same as the domain name in the message ID. If the message ID domain name is inconsistent with the sender's mailbox domain name, it indicates that the message is suspicious.

B. Distinguish Spam and Normal Mail by Text Classification

The input in this process is plain text extracted from the mail sample. Before classification, we still need to process the text and transform it into feature vector. First, we remove the words with vague meaning in the sample according to the stoppage dictionary, such as "the", "as", etc. Then replace the words with different forms but the same meanings, such as "it s" and "it is".

After preprocessing the text, TF-IDF is used to represent the feature vector of the document. TF-IDF describes the ability of a word to distinguish document content attributes. The more widely a word appears in a document set, the less distinguishable it is. On the other hand, the more frequently a word appears in a document, the stronger its ability to distinguish the content attributes of the document.

C. The Feature Vector for the Bi-LSTM-Attention Classifier

Before classification, we need to process plain text and convert it into feature vector. The tool used in this step is

word2vec. Word2vec is a Google open-source tool for word vector computing. Word2vec completes the process of using a shallow neural network to map sparse words into a dense vector of n dimension (n is generally several hundred). We use the word2vec API of gensim, an open-source third-party Python toolkit, to do this.

The specific steps of processing plain text are as follows:

- Load plain text data, divide sentences into word representations, and remove low frequency words and stop words.
- The word vector of the project is generated and saved by using gensim's word2vec API. Then, we map the words and labels of the data set to index representation, build the "vocabulary index" mapping table and "label index" mapping table, and save them in the file, then directly load the processing data. For words that are not in the pre training word vector of word2vec, "unk" is used.
- The word vector is read from the pre-trained word vector model and input into the model as initialization value.

V. Evaluation

A. Evaluation Indicator

KNN classifier. For the two classification of KNN classifier, each time is a binary classification problem. In the first classification, spam and normal mail are regarded as one class according to their low harm to users, so the sample set is divided into phishing mail and non phishing mail in the first classification. The second classification separates spam and normal mail.

For binary classification, there are four common terms: false negative(FN), false positive(FP), true negative(TN) and true positive(TP).

- FN: indicating that the sample is determined to be a negative sample, but it is actually a positive sample.
- FP: indicating that the sample is determined to be a positive sample, but in fact it is a negative sample.
- TN: indicating that the sample is determined to be a negative sample, which is also a negative sample in fact.
- TP: indicating that the sample is determined to be a positive sample, in fact, it is also a positive sample.

In order to evaluate the effect of the classifier, we choose accuracy and false positive rate as the evaluation indexes.

- Accuracy
  Accuracy is defined as the percentage of predicted correct results in the total sample. In binary classification, the formula is expressed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

- False Positive Rate
  By definition, the false-positive rate represents the probability of predicting a negative sample as a positive one. The formula is as follows:

$$FalsePositiveRate = \frac{FP}{FP + TN} \tag{6}$$

Bi-LSTM-Attention classifier. For Bi-LSTM-Attention classifier, the further classification of phishing mails is actually multiple classification problems. Referring to the binary classification problem, we calculate the accuracy, precision, recall and f-beta values as the evaluation indexes of the classification effect of the classifier.

- Accuracy
  The calculation of accuracy rate is the same as that of binary classification, and it also predicts the percentage of correct results in the total sample
- Precision
  The proportion of correctly predicted positive samples to all predicted positive samples. In binary classification, the formula is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

  We use the method of average to extend the concept of precision rate in binary classification to multivariate classification, that is, for each type of label, it is regarded as a positive sample, while the rest are regarded as a negative sample. Under this definition, a set of precision rates corresponding to the tags are calculated. The average value is regarded as the precision rate of multivariate classification
- Recall Rate
  The proportion of correctly predicted positive samples to all actual positive samples. In binary classification, the formula is as follows:

$$RecallRate = \frac{TP}{TP + FN} \tag{8}$$

  The method of average is also used to get the recall rate of multivariate classification problem.
- f_β
  The physical meaning of f_β is to combine the two scores of precision and recall into one score. In the process of merging, the weight of recall rate is β times of accuracy rate. The formula is as follows:

$$f_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{9}$$

  We calculate the f_β value when the beta value is 1, that is, precision and recall are equally important. The calculation method is to calculate f_β for each pair of precision and recall, and then calculate the average value.

B. Classification Effect Evaluation

KNN classifier. The effect of KNN classifier on mail classification is shown in Table 2. The first line represents the process of filtering out phishing mails, and the second line represents the process of distinguishing normal mails and spam mails by text classification.

Table II
THE EFFECT OF KNN CLASSIFIER

	Accuracy	FP rate
filtering out phishing mails	95.27%	1.22%
distinguishing normal and spam mails	94.53%	2.58%

Bi-LSTM-Attention classifier. In the process of training the model, 390 iterations were made. Take 40 times as the step, and the results of each evaluation index are shown in Figure 6. The final result is stable around 90%. The results of the last ten iterations are particularly shown in Figure 7.

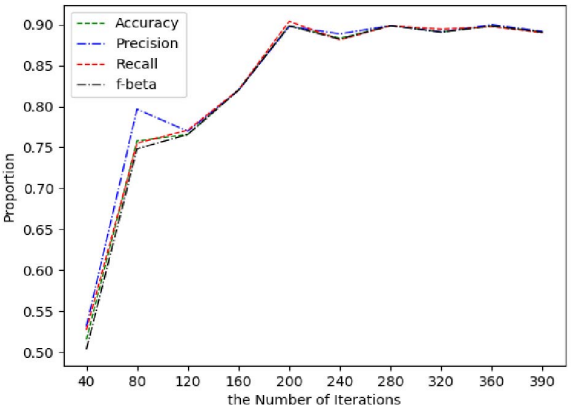

Figure 6. Evaluation index results in the iterative process.

Then the test set is used to evaluate the results, and the final classification results on the test set are shown in Table 3.

Table III
THE EFFECT OF BI-LSTM-ATTENTION CLASSIFIER

Accuracy	Precision
91.51%	91.75%
Recall rate	f_β
91.49%	91.58%

VI. SUMMARY

We propose a secure e-mail system using KNN algorithm and improved LSTM algorithm (Bi-LSTM-Attention). We hope that on the basis of traditional malicious e-mail filtering, we can more actively grasp the characteristics of e-mail from different attackers. KNN classifier is used to distinguish normal mail, spam and phishing mail, and Bi-LSTM-Attention classifier is used to classify phishing mails according to their attack sources. The result of classification is relatively good, which has the value of further research and application.

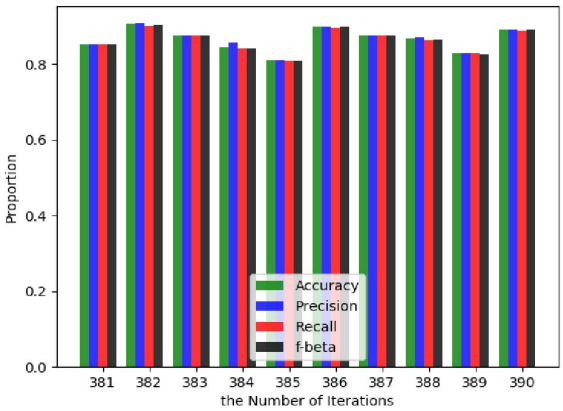

Figure 7. The results of the last ten iterations.

In the future, we will continue to study and make improvements in the following aspects:

- The current Bi-LSTM-Attention classifier has supervised learning and known classification categories when making classification. That is to say, the samples are manually arranged mails from different organizations without confusing the results. In the future, we hope to improve the algorithm to produce semi supervised learning effect [22], [23]. In this way, for new mail that does not belong to a known organization, the system can regard it as a new emerging organization. The improved classifier will be more conducive to the practical application of the system.
- In order to further put the system into application, we will study and use the API provided by the existing mailbox service to directly access and process users' mails [24].
- The feature vector of the current Bi-LSTM-Attention classifier is the processed message body text. In the future, other features will be considered according to the experimental results, such as email title, processed email image information, etc [25], [26].

ACKNOWLEDGMENT

The completion of the thesis is attributed to many people's support and encouragement.

First and foremost, I want to extend my heartfelt gratitude to my supervisor, Professor Cui Baojiang, and my senior, Dr. Yan Xiaodan, whose patient guidance, valuable suggestions and constant encouragement make me successfully complete

this thesis. My thanks also go to the authors whose books and articles have given me inspiration in the writing of this paper.

REFERENCES

[1] X. Zhou, F. C. Delicato, K. Wang and R. Huang, "Smart Computing and Cyber Technology for Cyberization," World Wide Web: Internet and Web Information Systems, available online in Jan. 2020. DOI: 10.1007/s11280-019-00773-y

[2] Y. Xu, G. Wang, J. Ren, and Y. Zhang, "An Adaptive and Configurable Protection Framework against Android Privilege Escalation Threats," Future Generation Computer Systems, vol. 92, pp: 210-224, 2019.

[3] A. Almomani, B. B. Gupta, S. Atawneh, et al. "A Survey of Phishing Email Filtering Techniques[J]". IEEE Communications Surveys and Tutorials, pp: 2070-2090, 2013.

[4] I. Fette , N. Sadeh, and A. Tomasic. "Learning to detect phishing emails." Proceedings of the 16th international conference on World Wide Web, pp: 649-656, 2007.

[5] J. A. Suykens, J. Vandewalle, Least squares support vector machine classifiers. Neural processing letters, 9(3), pp: 293-300, 1999.

[6] L. Breiman, Random Forests[J]. Machine Learning, 45(1), pp: 5-32, 2001.

[7] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "A Privacy-preserving Attribute-Based Access Control Scheme," 10th International Symposium on UbiSafe Computing (UbiSafe 2018), Springer, pp: 361-370, 2018.

[8] X. Zhou, B. Wu and Q. Jin, "Analysis of User Network and Correlation for Community Discovery Based on Topic-aware Similarity and Behavioral Influence," IEEE Transactions on Human-Machine Systems, vol. 48, no. 6, pp: 559-571, Dec. 2018. DOI: 10.1109/THMS.2017.2725341

[9] Y. Han, and Y. Shen. "Accurate spear phishing campaign attribution and early detection." Proceedings of the 31st Annual ACM Symposium on Applied Computing, pp: 2079-2086, 2016.

[10] A. Cidon, L. Gavish, I. Bleier, et al. "High precision detection of business email compromise." 28th USENIX Security Symposium (USENIX Security 19), pp: 1291-1307, 2019.

[11] S. Hochreiter, J. Schmidhuber, "Long short-term memory." Neural computation 9.8, pp: 1735-1780, 1997.

[12] P. Zhou, W. Shi, J. Tian, et al. "Attention-based bidirectional long short-term memory networks for relation classification." Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers), pp: 207-212, 2016.

[13] Y. Xu, W. Gao, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "FABAC: A Flexible Fuzzy Attribute-Based Access Control Mechanism," International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage (SpaCCS 2017), Springer, pp: 332-343, 2017.

[14] X. Yan, B. Cui, Y. Xu, P. Shi and Z. Wang, "A Method of Information Protection for Collaborative Deep Learning under GAN Model Attack," IEEE/ACM Transactions on Computational Biology and Bioinformatics, DOI: https://doi.org/10.1109/TCBB.2019.2940583

[15] D. G. Lowe, "Object recognition from local scale-invariant features." Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. Ieee, pp: 1150-1157, 1999.

[16] Y. Cheng, Mean shift, mode seeking, and clustering. IEEE transactions on pattern analysis and machine intelligence, 17(8), pp: 790-799, 1995.

[17] D. Radev,(2008), CLAIR collection of fraud email, ACL Data and Code Repository, ADCR2008T001, http://aclweb.org/aclwiki

[18] Y. Xu, W. Gao, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "A Feasible Fuzzy-Extended Attribute-Based Access Control Technique," Security and Communication Networks, Article ID 6476315, pp: 1-11, 2018.

[19] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain. A Real-Time Matching System for Large Fingerprint Databases, IEEE Transactions On Pattern Analysis And Machine Intelligence, VOL.18, NO.8, pp: 799-813, 1996.

[20] X. Zhou and Q. Jin, "A Heuristic Approach to Discovering User Correlations from Organized Social Stream Data," Multimedia Tools and Applications, vol. 76, no. 9, pp: 11487-11507, May 2017. DOI: 10.1007/s11042-014-2153-5

[21] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo and C. Li, "Learning URL Embedding for Malicious Website Detection," IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Informatics, vol. 16, no. 10, pp: 6673-6681, 2020.

[22] T. Scheffer, "Email answering assistance by semi-supervised text classification." intelligent data analysis (2004), pp: 481-493, 2004.

[23] Y. Meng, W. Li, L. Kwok, et al. "Enhancing email classification using data reduction and disagreement-based semi-supervised learning", international conference on communications, pp: 622-627, 2014.

[24] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "An Efficient Privacy-Enhanced Attribute-Based Access Control Mechanism," Concurrency and Computation: Practice and Experience, vol. 32, no. 5, pp: 1-10, 2020. DOI: https://doi.org/10.1002/cpe.5556

[25] X. Zhou, B. Wu and Q. Jin, "User Role Identification Based on Social Behavior and Networking Analysis for Information Dissemination," Future Generation Computer Systems, vol. 96, pp: 639-648, Jul. 2019. DOI: 10.1016/j.future.2017.04.043

[26] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo and T. Guo, "Trustworthy Network Anomaly Detection Based on an Adaptive Learning Rate and Momentum in IIoT," IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp: 6182-6192, 2020.