Computers, Materials & Continua
DOI: 10.32604/cmc.2023.029135
Article

Tech Science Press

Attack Behavior Extraction Based on Heterogeneous Cyberthreat Intelligence and Graph Convolutional Networks

Binhui Tang[1,3], Junfeng Wang[2,*], Huanran Qiu[3], Jian Yu[2], Zhongkun Yu[2] and Shijia Liu[2,4]

[1]School of Cyber Science and Engineering, Sichuan University, Chengdu, 610065, China
[2]College of Computer Science, Sichuan University, Chengdu, 610065, China
[3]Jincheng College of Sichuan University, Chengdu, 610065, China
[4]Institute for Infocomm Research, A∗STAR Singapore, Singapore
*Corresponding Author: Junfeng Wang. Email: wangjf@scu.edu.cn

Received: 26 February 2022; Accepted: 08 June 2022

Abstract: The continuous improvement of the cyber threat intelligence sharing mechanism provides new ideas to deal with Advanced Persistent Threats (APT). Extracting attack behaviors, i.e., Tactics, Techniques, Procedures (TTP) from Cyber Threat Intelligence (CTI) can facilitate APT actors' profiling for an immediate response. However, it is difficult for traditional manual methods to analyze attack behaviors from cyber threat intelligence due to its heterogeneous nature. Based on the Adversarial Tactics, Techniques and Common Knowledge (ATT&CK) of threat behavior description, this paper proposes a threat behavioral knowledge extraction framework that integrates Heterogeneous Text Network (HTN) and Graph Convolutional Network (GCN) to solve this issue. It leverages the hierarchical correlation relationships of attack techniques and tactics in the ATT&CK to construct a text network of heterogeneous cyber threat intelligence. With the help of the Bidirectional Encoder Representation from Transformers (BERT) pretraining model to analyze the contextual semantics of cyber threat intelligence, the task of threat behavior identification is transformed into a text classification task, which automatically extracts attack behavior in CTI, then identifies the malware and advanced threat actors. The experimental results show that F1 achieve 94.86% and 92.15% for the multi-label classification tasks of tactics and techniques. Extend the experiment to verify the method's effectiveness in identifying the malware and threat actors in APT attacks. The F1 for malware and advanced threat actors identification task reached 98.45% and 99.48%, which are better than the benchmark model in the experiment and achieve state of the art. The model can effectively model threat intelligence text data and acquire knowledge and experience migration by correlating implied features with a priori knowledge to compensate for insufficient sample data and improve the classification performance and recognition ability of threat behavior in text.

Keywords: Attack behavior extraction; cyber threat intelligence (CTI); graph convolutional network (GCN); heterogeneous textual network (HTN)

1 Introduction

Cyberattacks are a critical factor affecting national political, financial, and social security as the most intense, covert, and frequent form of inter-state conflict. The Advanced Persistent Threat (APT) attacks are a severe threat to cyberspace, bringing significant challenges to security defense [1]. Extracting attack Tactics, Techniques, and Procedures (TTP) from threat intelligence can model advanced threat behavior profiling and respond to advanced persistent threats in time. Through the study of APT attacks, it is found that this type of attack has a tendency to be organized and weaponized and is prone to use covert channels to cause sensitive data leakage or physical equipment damage. The attack characteristics are three aspects: (1) well-prepared before the attack; (2) covert behavior during the attack; (3) difficult to obtain evidence after the attack. Advanced threat actors are based on the actual situation of the attack target and adjustment of attack techniques timely to achieve maximum attack effect. At present, the main difficulty in defending against APT is the lack of effective methods to analyze the behavior patterns of advanced threat actors. With the help of the Cyber Threat Intelligence (CTI) sharing mechanism [2], most threat event reports can be shared to provide a reference for analyzing the threat behavior profiling of advanced threat actors. Threat event reports are a type of CTI containing rich TTP information [3] in attack events, attack behavior of malware, etc., which are critical for threat hunting and attack attribution analysis. However, threat event reports are described through natural language texts. How to automatically, accurately, and quickly extract threat behavior from the massive amount of threat intelligence text has become a significant concern in academia and industry [4]. Understanding complex behavioral relationships in the technical text was a recognized challenge in the Natural Language Process (NLP). Existing word vector representation methods have the shortcoming of "one word, one meaning," which cannot provide context-dependent word vectors for information extraction tasks. The interdependencies between contextual semantics are not effectively mined, the correlations between threat behaviors cannot be effectively understood, and implementing the threat behaviors identification task is even more difficult.

In recent years, deep language representation models have become the mainstay of natural language processing techniques and the basis for knowledge extraction tasks. To address the challenges of threat behavior extraction in cyber threat intelligence, we propose a method for threat behavior extraction, which leveraged the Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) to explain the semantic relationship between attack behavior and construct the heterogeneous textual network [5]. At the same time, the heterogeneous textual fused graph convolutional network and deep semantic representation model for tactics and techniques identification and extraction. The framework uses domain corpus to retrain the pretraining model of word embedding and address the problem of inaccurate text feature representation. It can compensate for the shortcomings of insufficient training samples and incomplete semantic and syntactic information in the text. Firstly, the word embedding representation method is based on the Bidirectional Encoder Representation from Transformers (BERT) pretraining model to improve the semantic representation. Secondly, the Bi-directional short-term memory network (BiLSTM) [6,7] is constructed to extract semantic features of the context. Thirdly, the Graph Convolutional Network (GCN) [8] is used to fuse the contextual semantic features with threat behavior relationships of the text and obtain more accurate results in the task of threat behavior extraction. The paper utilizes the ATT&CK framework [9] to improve the hierarchical relationship of techniques and tactics in threat intelligence text and extend the text dataset with malware and advanced threat actors labels. Therefore, the method can also be effectively applied to threat hunting to enhance malware's recognition ability and advanced threat actors. The experimental results show that the method achieves the F1 of 94.86% and 92.15% for threat behavioral tactics and techniques extraction tasks. The F1 of the malware and advanced threat actors recognition task reached 98.57% and 99.4%, which are better than the experiment's benchmark model and achieve the best results.

The main contributions of this paper are as follows:

(1) This paper collects more than 57,000 APT threat event reports, blogs, and other articles to build a corpus in the field of cybersecurity. At the same time, we annotate 6512 textual description sentences of attack techniques, tactics, attack tools, and the malware used by advanced threat actors. We utilize the ATT&CK threat behavior description framework, including 14 tactics and 247 root techniques and sub-techniques, 233 malware, and 83 advanced threat actors to construct the Heterogeneous textual relationship network to train the model of attack behavior extraction.

(2) This paper addresses the existing challenges of threat behavior knowledge extraction and proposes a fused neural network framework to extract attack behaviors in cyber threat intelligence. The deep learning framework HT-GCN based on Bert+ BiLSTM +GCN improves the word embedding representation of cyber threat intelligence text, solves the unbalanced distribution of threat behavior data in the text, and reduces the semantic gap between text descriptions and attack behaviors (i.e., TTPs). The method can improve the results of threat behavior extraction.

(3) This paper can help security analysts better understand the attack behaviors of malware and advanced threat actors in the cyber threat intelligence, form practical evidence for identifying malware and attack threat actors and improve the results of threat hunting.

2 Related Work

2.1 Threat Behavior Description Framework

As the complexity of APT attacks has increased and the adversarial has grown, the threat behavior description framework has been continuously adapted and improved. The ATT&CK model [10] is MITRE corporation based on Lockheed Martin's KillChain model [11] for describing advanced threat behaviors as a more fine-grained and easily shared knowledge, which serves as a comprehensive knowledge base for APT attacks by understanding all phases of the attack lifecycle and covers a wide range of threat behaviors performed by advanced threat actors. The framework is a structured threat behavior description metric for TTP that can effectively react to the attack impact of threat events throughout the attack lifecycle. Currently, the framework is continuously adapted and updated with the use of security vendors and enterprises, which has become the most comprehensive knowledge framework for describing APT threat behaviors and has become the authority and benchmark in the cyber security field. From the attacker's perspective, ATT&CK classifies the threat behaviors into tactical and technical categories. The tactical category describes the attacker's attack intent, and the technical category describes the attacker's attack method. They can help security experts to understand the attacking intent and predict the consequences of behavior. In practice, the ATT&CK knowledge framework can also be used to simulate the threat implementation process of attackers to assess the integrity of defense techniques and generalize the threat behavior patterns of attackers to identify the attackers from the overlap of the attack chain, thus enabling threat hunting and attribution analysis. The literature [12] proposes a security threat modeling language based on the ATT&CK knowledge framework, designed using a meta-attack language framework and focusing on describing the attack steps and defense methods associated with system assets. Legoy et al. [13] also tried to use Support Vector Machines (SVM) to identify technical and tactical in the threat intelligence texts. But the results and performance were poor, the best outcome for tactical was 65.38%, and the best result for technical was only 35.02%. It is worth noting that the far difference in the results of tactics and techniques is the significant difference in the number of tactical and technical categories datasets. In the latest version of ATT&CK, there are 14 tactical categories, while there are more than 300 technical categories, so the average number of data for technical types training is much lower than that of tactics in the same datasets, leading to poorer classification results. This paper considers using semi-supervised graph convolutional neural networks to enhance the accuracy of technique classification by using the hierarchical relationship of techniques and tactics on ATT&CK.

2.2 Text Classification Methods

As a fundamental task in natural language processing, text classification is essential for entity recognition, relationship extraction, and knowledge graph construction. Traditional classification methods are based on the manual extraction of text features with machine learning, such as Bayesian classifiers, Decision Trees, SVM [14] and the Hidden Markov Model (HMM) [15], which are widely used in text classification tasks. At present, deep learning-based methods have been applied to text classification tasks by training different neural network models, such as Convolutional Neural Network (CNN) [16], Recurrent Neural Network (RNN) [17], and Bidirectional Long Short-Term Memory (BiLSTM) [18], etc. They have obtained good results. Text classification mainly includes three steps: feature representation, feature extraction, and classifier training. Many studies have been conducted on text classification tasks, mainly focusing on representing text features and optimizing classifier models. In the study of text feature representation, the primary purpose is to improve the word embedding model, and the optimization of the model is mainly used to improve the accuracy of feature extraction and classification.

2.2.1 Text Representation Methods

Text representation uses word vectors to represent text features, and their performance has an essential impact on downstream tasks. Word vectors are also among the most common text feature representations and are widely used as input features for text in various natural language processing tasks. Early Word Vector Representation (WVR) methods are primarily context-independent independent features that strip the association between words in a text, such as one-hot representation, also known as independent word representation. This method can easily distinguish different words, but it cuts the correlation between words and can easily cause a "dimensional explosion." Some scholars have also proposed word embedding techniques, which use neural network models to learn the co-occurrence of words and obtain a low-dimensional vector of semantic information through unsupervised learning, such as Word to vector (Word2Vec) [19], Global Vectors for Word Representation (GloVe) [20]. They have achieved good results better than One-hot. Traditional word embedding models are word-based, which provide a word vector representation for each word that appears in the corpus and cannot be vectorized when the word is not in the training corpus. To solve this problem, FastText [19] uses character-level n-grams to represent a word, such that a word consists of a finer-grained set of familiar n-gram characters. Generally speaking, words with similar n-gram characters have identical semantics and are in a similar vector space. Therefore, FastText can capture the similarity of words, which is more beneficial for word vector representation of low-frequency words. However, these static word vector representation methods only correspond to a one-word vector and cannot solve the multiple meanings. These models extract context-independent independent features, stripping the association between words in the text, and cannot fully utilize the contextual information to dynamically represent the semantics, making it difficult to obtain better results in NLP tasks.

With the further development of deep learning techniques, the pretraining language models have been breakthroughs in many NLP tasks. The word vectors of pretraining models contain multiple semantic and syntactic information dimensions. Many pretraining models improve the representation of word vectors through different masking strategies in pretraining tasks, leading to differences in their word vectors. Devlin et al. proposed the Bert model [21], which contains an encoder layer with a self-attentive mechanism, and is one of the most successful deep neural network models in recent years. The model uses bidirectional representation and a self-attentive tool to train different text and obtain rich semantic features in context, leading to significant improvements for various natural language processing tasks. With the help of pretraining tasks and attention mechanisms, Bert can better understand the meaning of the utterance and use contextual information to solve the problem that the traditional word vector representation cannot have multiple definitions for the word [22].

2.2.2 Text Feature Extraction and Classification Methods

Text feature extraction is a fundamental task in text classification, which is divided into two methods: machine learning and deep learning. The machine learning method was usually adopted word frequency statistics, e.g., Term Frequency-Inverse Document Frequency (TF-IDF) [23] can characterize text features by considering the word frequency and inverse document frequency, which can better classify the texts. The machine learning methods cannot mine more profound into the contextual semantic information of text and do not perform well on text classification tasks with high abstraction, such as the threat behavior recognition task in this paper. Many researchers started to study deep learning methods to solve this problem and improve text classification performance. Deep learning automatically extracts high-dimensional text features by stacking multi-layer neural network models, thus enabling better text classification. Convolutional Neural Networks (CNN) model captures the local information by convolving the hierarchical text structure and changing convolutional kernel parameters to complete the extraction of high-dimensional features and achieve classification. The literature [24] proposes a Convolutional Neural Network model for processing text sequences with excellent performance in text classification tasks—Text Convolutional Neural Networks (TextCNN). TextCNN uses convolution kernels of different sizes to extract the critical information in text, especially by "one-dimensional convolution" to obtain the N-gram text feature, which can well capture the local relevance of context and achieve good results in text classification. With the success of TextCNN on text classification tasks, Very Deep Convolutional Networks (VDCNN), Deep Pyramid Convolutional Neural Networks (DPCNN), Deep Bilinear Convolutional Neural Networks (DBCNN), A Tree-Based Convolutional Neural Networks (TBCNN), and other CNN models for different text classification scenarios were born. Unlike CNN, Recurrent Neural Networks (RNN) perform recurrent recursive processing of text implicit layers that enable better capturing of context and mining of the text's sequence features.

Compared with TextCNN, TextRNN is more suitable for the classification task of long text, which is Recurrent Neural Network (RNN) for text classification. However, TextRNN suffers from gradient disappearance or explosion problems, so variants of recurrent neural networks such as BiLSTM and Gated Recurrent Unit (GRU) have been proposed for text classification tasks. In addition, the deep learning methods with graph-structured data have been a significant development. The Graph Convolutional Network (GCN) can deal with graph data as a Semi-supervised model to achieve end-to-end learning of node features and structural features, which has received widespread attention from researchers. Yao et al. [25] have proposed TextGCN, a text graph convolutional neural network, which is a text graph constructed based on word co-occurrence and document-word relations and uses a graph convolutional network for text classification. The existing research shows that TextGCN is more robust than before in text classification tasks with less training data. Graph convolutional neural networks can build graph models by passing information between nodes. At the same time, some methods have proposed a heterogeneous attention mechanism with TextCNN to improve the accuracy of text classification [26]. Initial GCNs cannot capture both short-term and long-term contextual associations, which can only be solved by increasing the number of GCN layers to capture contextual associations. Still, a multi-layer GCN algorithm for text classification tasks incurs a high space complexity. Increasing the number of network layers may lead to excessive smoothing of node features and convergence of local features to similar values, resulting in degraded classification performance.

2.3 Threat Behavior Recognition Methods

With the development of Natural Language Processing (NLP) techniques, the automatic recognition of threat behavior from cyber threat intelligence has been better achieved. Knowledge extraction is a fundamental task in NLP, which is significant for text classification, topic identification, and knowledge graph construction. Knowledge extraction includes Named Entity Recognition, Relationship Extraction, Event Extraction, and other tasks with rule-based methods, feature-based machine learning methods, and deep learning methods for different text structures. Rule-based knowledge extraction methods mainly deal with structured text with similar design and simple syntax, such as IP, Email, Domain URL, CVEs, etc. Li et al. [27] used a set of regular expressions (Regex) extracted from IOCharterms to find sentences containing assumed IoC tags such as IP, MD5, and other strings. Machine learning methods have been widely used to process semi-structured textual data. Husari et al. [28] proposed the TTPDrill to extract TTP in cyber threat intelligence, using word frequency statistics and support vector machines for text classification. The method also uses the hierarchical relationship between TTP to improve the effectiveness of tactics and techniques extraction by voting to calculate the confidence level. Structured data limit rule-based and machine learning methods. However, cyber threat intelligence contains unstructured data with different grammatical and irregular text structures, and unstructured data mainly adopts deep learning methods to represent text and discover implicit threat behavior features [29]. The literature [30] proposed that the threat entities extraction model is designed with RNNs and CRFs, which uses multi-granularity BiLstm to mine the threat intelligence contextual relationships, extracts implicit features, performs state transformation on the features by CRF, and achieves the extraction of IoC entities. The literature [31] uses heterogeneous graph convolutional neural networks to model threat intelligence and mine the deeper implicit relationships between IoCs. It is found that using a single CNN or RNN feature extractor is used to extract IOC information and cannot effectively extract deep features of threat behavior, such as TTP. The Attention mechanism is gradually being used in natural language processing because it enhances the ability to mine textual feature information. The literature [32] introduced the attention mechanism in Long Short Term Memory (LSTM) networks to increase the weight of keywords, thus improving the text classification ability. CNN extracts the text features, and the attention layer performs multi-granularity weighting on the feature map, thus improving the entity classification ability of the model [33].

3 Methodology

The above research lays the foundation for the task in this paper. It can be seen from the related work that most of the natural language processing tasks use pretraining models for word vector training and ensure the generalization performance of the models [34]. Many studies use generic corpora, which differ significantly from the security domain in terms of text length, format, and contextual relevance. Although the word vector features obtained using the pretraining model consider the contextual semantics, the lack of security prior knowledge and the presence of semantic gaps between the knowledge domains bring many limitations to improving the model performance further

[35]. Flattened deep learning methods perform poorly on fine-grained threat behavior classification tasks because of their inability to leverage the hierarchical relationship between tactics and techniques [36]. To solve the problem of the contextual relevance in the text and the relationship of word-to-word was stripped, the paper aims to design a knowledge extraction framework that contains Bert+BiLstm+GCN models. The framework has three parts: text representation, feature extraction, and text classifier to improve the text classification result with deep learning methods, which can help to enhance the effectiveness of the threat behavior recognition task by studying and optimizing the model architecture [37].

3.1 Model Architecture

This paper proposes an attack behavior extraction framework with a textual multi-label classification method, which fuses the hierarchical features from the tactics and techniques Heterogeneous Text by Graph Convolutional Network model for threat behavior recognition in cyber threat intelligence. The framework consists of three modules, Bert+BiLstm+GCN. It contains four parts: text representation, text feature extraction, text feature propagation, and output of the classification results, as shown in Fig. 1:

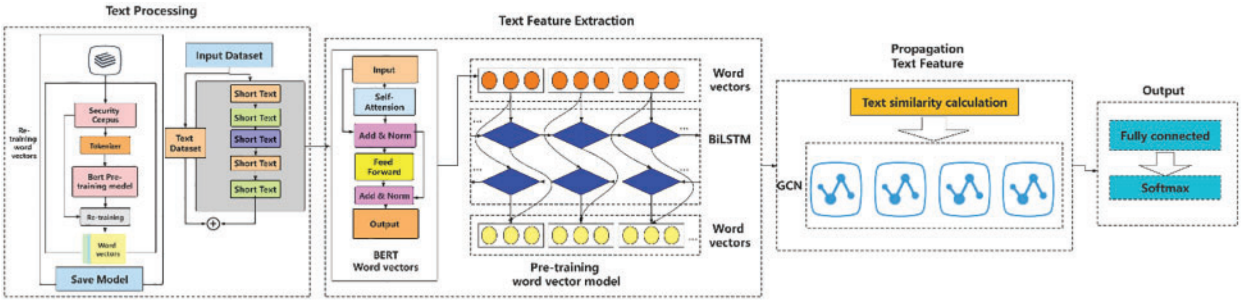

Figure 1: The framework of the threat behavior extraction model

(1) Text Representation: At first, we collect a large amount of cyber threat intelligence, including threat event reports, blogs, malware analysis reports, advanced threat actors analysis reports, and other textual data. We leverage the technical and tactical relationships summarized in the ATT&CK threat behavior framework to classify and annotate the threat behavioral text and form a training dataset reflecting the technical and tactical relationship between malware and advanced threat actors. The various types of threat intelligence from the training set needs to be divided into sentences and words, and long texts are processed into short texts of fixed length and sliced into words. Finally, these are fed into the Bert model for word vector transformation to form a text vector.

(2) Text Feature Extraction: Many Out-of-vocabulary (OOV) words were used to describe the critical features of threat behavior in the threat intelligence text. Given the challenges of word embedding techniques, the accuracy of OOV word representation affects the targeting task results. Therefore, this paper chooses the BERT model for word vector retraining and the BiLSTM model used as a feature extractor in this phase. The bidirectional extracted contextual features significantly improve the classification results.

(3) Text Feature Propagation: to address the problem of the poor effect of semantic association features, this method is based on a hypothesis-text features that can contribute to the improvement of the target task should they overlap in the semantic space, and the semantics

of the overlapping parts should be made closer when the target task is trained. Based on this assumption, the threat behavioral relationships should be used to construct a heterogeneous textual relational network, fuse and propagate the textual features by graph convolutional neural network model and explore the impact of the fusion of text features on the threat behavioral recognition task.

(4) Output the results, the text representations obtained by the GCN are input to the fully connected layer, cross-entropy is applied as the loss function for model training, the Softmax function generates the probability value of each category, and the types of entities are classified according to the maximum value of the probability. In this paper, the GCN model can propagate the features of tactical class text to technical text to solve the sparse problem of technical data and use the prior knowledge of the tactical text to improve the results of technical text classification. It is explained in detail in the following contents.

3.2 Re-Training Word Embedding Model

The pretraining model for word embedding tasks usually obtains the relative weights under a comprehensive corpus [38]. There are differences in the distribution of data features between the pretraining corpus and the target task corpus, which may lead to errors between the word vector representation of keywords and their correct semantics in the target context, introducing false noise to the model training and degrading the model performance. In order to address the problem, the method was used to retrain the security corpus with the Bert pretraining model and fine-tune the initial weights to enable the word vector to better match the context of the target task [39].

3.3 Textual Features Extraction Model

The BiLSTM model can understand the context better because it can ensure that each word gets more semantic information with full consideration of the context, providing a deep text feature representation. The quality of the initial word embedding features affects the performance of the target task, so this paper introduces the BiLSTM model to learn the context to obtain more semantic information and extract deeper text features.

3.4 Constructing Hierarchy Relationship of Heterogeneous Textual Network

In recent years, graph convolutional neural networks (GCN) can achieve end-to-end learning of node and structural features, showing advantages over other traditional models in the prosperous relationship of graph data [40]. This paper uses GCN models to explore the effectiveness of multi-label classification tasks in attack behavior extraction, which focuses on using hierarchy textual relational networks to identify attack behaviors in cyber threat intelligence and GCN models to fuse textual relational features. The purpose is to explore the effectiveness of the threat behavior identification task transformed into a multi-label text classification task.

3.4.1 Measuring Text Similarity

Firstly, we construct a textual relationship network based on the hierarchical relationship of text labels and the similarity of text semantics. The nodes in the text network are composed of short texts describing the attack behavior feature. The edges between nodes are built using the hierarchical relationship of text labels and the similarity of texts. The TF-IDF method is used to calculate text similarity. The detailed steps are as follows: (1) dividing the complete sentences into independent word sets according to the word splitting algorithm; (2) finding the ensemble of two-word sets; (3)

calculating the word frequencies of each word in sentences and vectorizing them; (4) calculating the text-similarity using the cosine formula. Generally speaking, the larger the word frequencies in the text, the more critical the word in these sentences, so the calculation formula is shown in Eqs. (1)–(3):

$$IDF(w) = \log\left(\frac{D}{Dw+1}\right) \tag{1}$$

$$TF(w) = \frac{n_w}{\sum_{i=1}^{1} n_i} \tag{2}$$

$$TF - IDF = tf(w) \times idf(w) \tag{3}$$

Secondly, the similarity between two texts is expressed by calculating the euclidean distance or cosine similarity between text vectors. Cosine similarity measures the similarity between vectors by calculating the cosine value. The more similar the vectors are, the smaller the angle between the vectors, and the closer the cosine value is to 1, the more similar the two vectors are. Suppose n1 and n2 are two n-dimensional vectors $n_1 = (x_1,x_2,\ldots\ldots x_n)$, $n_2 = (x_1,x_2,\ldots\ldots x_n)$, the cosine of the angle between them in Eq. (4), which can also be seen as the inner product of two vectors divided by the modal length of the vectors. After the keyword vector obtains, the vector accumulates horizontally to calculate the similarity between sentences to get the first-layer relationship of the text network. The calculation method shows in Eq. (5).

$$\cos(\theta) = \frac{\sum_{i=1}^{n}(X_i \times Y_i)}{\sqrt{\sum_{i=1}^{n}(X_i)^2} \times \sqrt{\sum_{i=1}^{n}(Y_i)^2}} \tag{4}$$

$$\cos(\theta) = \frac{n_1 \cdot n_2}{\|n_1\| \times \|n_2\|} \tag{5}$$

3.4.2 Constructing Heterogeneous Textual Relationships Network

The paper proposed a threat behavior extraction framework to construct a heterogeneous relationship network containing multi-layer relationships of word-word, word-document, and document-document [41]. The relationships are based on text labels of threat behavior, containing 12 tactics and 247 techniques. Then, we summarize 83 advanced threat actors and 233 malware, including these threat behaviors. We leverage these relationships to extend the text network. Text-relationship network construction steps are shown in Fig. 2.

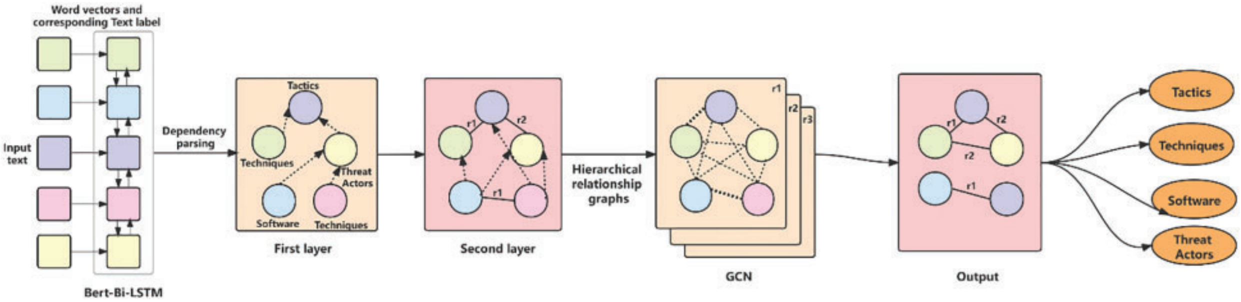

Figure 2: Schematic graph of heterogeneous textual relationships network

3.5 Text Feature Extraction and Propagation Model

This paper combines the Bert and BiLstm to train word vectors and represent text features, using the BiLSTM model to achieve deeper extraction of text features to make the text features consistent with the target task. Then, the graph convolutional neural network model (GCN) utilized the text relationship network to propagate the text features. GCN consists of the feature matrix of the nodes and the graph's adjacency matrix. The adjacency matrix mainly represents the edge relationships between the nodes. The hidden layer of GCN can use the propagation rules to aggregate the node information of the current layer and transmit the text features to the next layer. As the text features are propagated and aggregated layer by layer, the weighting influence of the labeling relationships makes the semantically close text overlap incrementally, thus achieving the best performance and better than the benchmark model in the experiment.

4 Experiment

4.1 Experimental Procedure

In this paper, all experiments were run on a server with 256 GB of RAM, a GeForce RTX 2080Ti, and an Intel(R) Xeon(R) Silver 4210R CPU @ 2.40 GHz on Ubuntu 18.04.5 with PyTorch CUDA version 11.3. The experimental procedures are shown in Fig. 3, which mainly includes four layers: input layer, text feature extraction layer, text feature propagation layer, and output layer.

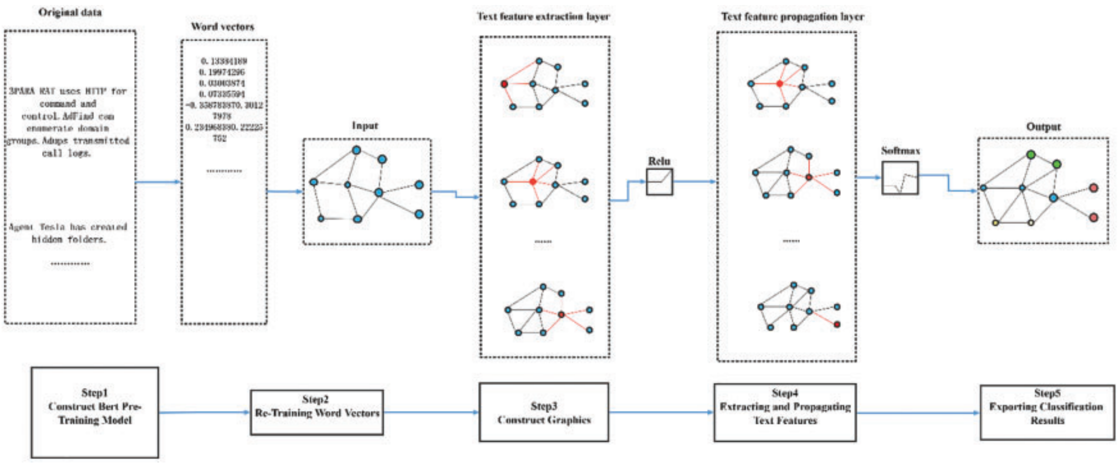
Original data
3PARA RAT uses HTTP for command and control. AdFind can enumerate domain groups. Adups transmitted call logs.
Agent Tesla has created hidden folders.
....
Word vectors
0.13384189
0.19974296
0.03003874
0.07335594
-0.338783870.3012
7978
0.234968390.22225
752
....
Input
Text feature extraction layer
......
Relu
Text feature propagation layer
......
Softmax
Output
Step1
Construct Bert Pre-Training Model
Step2
Re-Training Word Vectors
Step3
Construct Graphics
Step4
Extracting and Propagating Text Features
Step5
Exporting Classification Results

Figure 3: The overview graph of experimental procedure

4.1.1 Input Layer

The layer is mainly pre-processing of the dataset that the training text is processed into words by a word splitter and input into the retrained BERT model. The text vector-matrix Wi $\in R^{M*N}$ is obtained after the embedding layer, where M is the text length, and N is the dimensionality of the word vector. At present, the parameters of the BERT model are the dimensionality of input and hidden layer are both 768, have 12 layers of multi-head self-attention layers, and support sentences with a maximum length of 512 tokens.

4.1.2 Text Feature Extraction Layer

This layer mainly applies the BiLSTM model to BERT for feature extraction. For example, by setting the unit parameter in the BiLSTM model as 200, the 768-dimensional vector of words in BERT can be extracted as 200-dimensional vector so that each word can be better considered in context and the role of feature dimensionality reduction to improve the performance of the model.

4.1.3 Text Feature Propagation Layer

This layer mainly uses graphs to model textual relationships; the nodes in the graph are textual units containing words and short sentences. At the same time, the edges are linked based on semantic similarity between nodes. Specifically, the hierarchical relationship network of threat intelligence text is constructed, where the text labels are labeled with ATT&CK's attack techniques and tactics as the subject of the text. Since attack techniques and tactics have a typical hierarchical structure, i.e., a tactical classification contains multiple root techniques, and a root technique includes multiple sub-techniques. There are more techniques than tactics, which makes the technique identification much less effective and more complicated than the tactics in the case of data sets. Leveraging hierarchical relationships to improve the issues, the edges' weights between two nodes i and j are defined based on the word frequency-inverse document frequency (TF) and positive point mutual information (PPMI). As shown in Eq. (6):

$$A_{i,j} = \begin{cases} PPMI(i,j), & i,j \text{ are words, and } i \neq j \\ TF - IDF(i,j), & i \text{ represents a document, } j \text{ represents the word} \\ \quad 1, & i = j \\ \quad 0, & otherwise \end{cases} \tag{6}$$

GCN is designed for each layer based on the hierarchical relationships of techniques and tactics, which consists of the feature matrix and the adjacency matrix of the graph. The adjacency matrix represents the relationship between the reference nodes. The hidden layer can aggregate the node information of the current layer through propagation rules and transfer the features. The hidden layer can aggregate the node information of the current layer by propagation rules and move the features to the next layer by aggregating the text features of the upper layer to the lower layer. The features of different layers are similar or even partially overlapped, but not wholly overlapped so that the GCN model can enhance the ability. The hierarchical propagation rule for the ith node is shown in Eqs. (7)–(9).

$$h_i^l = \sigma\left(\sum_{j=1}^{N} \overline{A}_{ij} \cdot W^l \cdot h_i^{l-1} + b^l\right) \tag{7}$$

$$\overline{A} = D^{-\frac{1}{2}} \cdot A \cdot D^{-\frac{1}{2}} \tag{8}$$

$$D_{ii} = \sum_{j=1}^{N} A_{ij} \tag{9}$$

4.1.4 Exporting Results

The aggregated text feature vectors are connected to the fully connected layer. The Softmax function generates the probability values of each category. The threat behavior is classified according to the maximum value of the probability. After training the upper tactical layer, the model migrates the parameters to the lower layer. The technical training process is the same as the tactical training process. The calculation process is shown in Eq. (10):

$$P = \frac{e^{v_i}}{\sum_{j=1}^{n} e^{v_j}} \tag{10}$$

E is the exponential operation with e as the base, and p is the vector consisting of the probabilities that the sentence belongs to each label, i is the text divided into the number of the sentence. A softmax activates to complete the classification task. After training the upper tactical layer, the model migrates the parameters to the lower layer. The technical training process is the same as the tactical training process.

4.2 Experimental Evaluation Metrics

In terms of experimental metrics, Recall, Precision, and F1 values are used as metrics to evaluate the model performance. The calculation process shows in Eqs. (11)–(13):

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

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

$$F_1 = \frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{13}$$

4.3 Experimental Results Analysis

The paper adopts the advantages of semi-supervised text embedding and uses text label relationships for text representation learning. The method learns a low-dimensional text vector representation from a limited number of labeled and unlabeled texts and then uses the text representation features for multiple tasks. After several epochs of experiments and averaging, the experimental results for the tactics are shown in Tab. 1 and Fig. 4:

Table 1: The comparison of experimental results for attack tactics extraction

Model/Tactics	Recall	Precision	F1
Bert(before re-training)	0.7470	0.7578	0.7825
BiLstm	0.8918	0.9089	0.9003
BiLstm+Attention	0.8918	0.9145	0.9031
CRNN	0.9136	0.9212	0.9174
CRNN+Attention	0.9226	0.9261	0.9244
TextCNN	0.8693	0.9212	0.8945
TextCNN+Attetion	0.9376	0.9355	0.9366
TextGCN	0.8871	0.8953	0.8411
Bert+BiLstm+GCN	0.9498	0.9412	0.9486
HT-GCN	**0.9498**	**0.9412**	**0.9486**

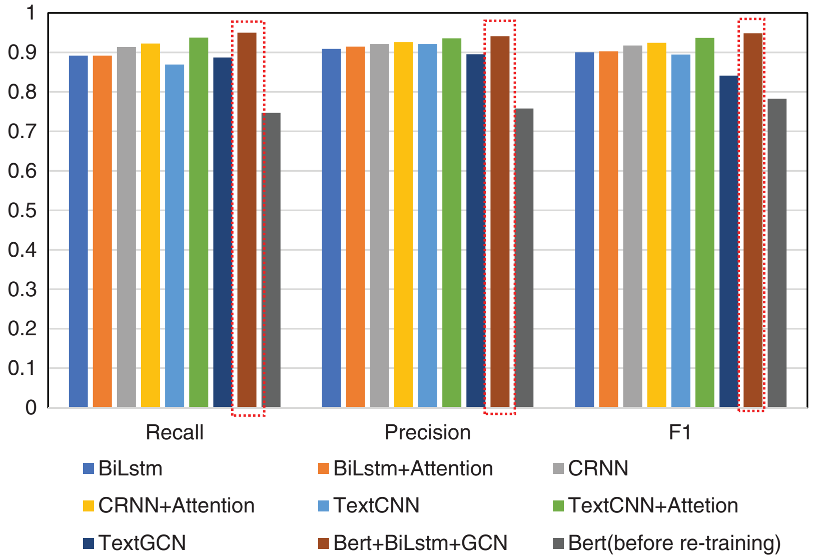

Figure 4: The comparison of experimental results for attack tactics extraction

The results show that the model proposed in this paper is optimal for the tactical extraction task in Recall, Precision, and F1. It was also found that the attention mechanism enables the model that focuses on the critical features, which is effective with RNN, CNN, and CRNN models. The model performance is also slightly upgraded by using the attention module. In the techniques extraction experiments, the hierarchical network is used to fuse the features of different levels of technique, which can combine the upper-level model's training parameters with improving the model's performance. The results show that the attack techniques extraction fused Bert+BiLstm+GCN models is much better than other benchmark models in Tab. 2 and Fig. 5.

Table 2: The comparison of experimental results for attack techniques extraction

Model/Techniques	Recall	Precision	F1
Bert(before re-training)	0.7353	0.7764	0.7382
BiLstm	0.8034	0.8097	0.8065
BiLstm+Attention	0.8179	0.8531	0.8353
CRNN	0.8335	0.8497	0.8415
CRNN+Attention	0.8324	0.8553	0.8437
TextCNN	0.8067	0.8771	0.7892
TextCNN+Attetion	0.8547	0.8879	0.8497
TextGCN	0.7982	0.7689	0.7512
Bert+BiLstm+GCN	0.9257	0.9278	0.9215
HT-GCN	**0.9257**	**0.9278**	**0.9215**

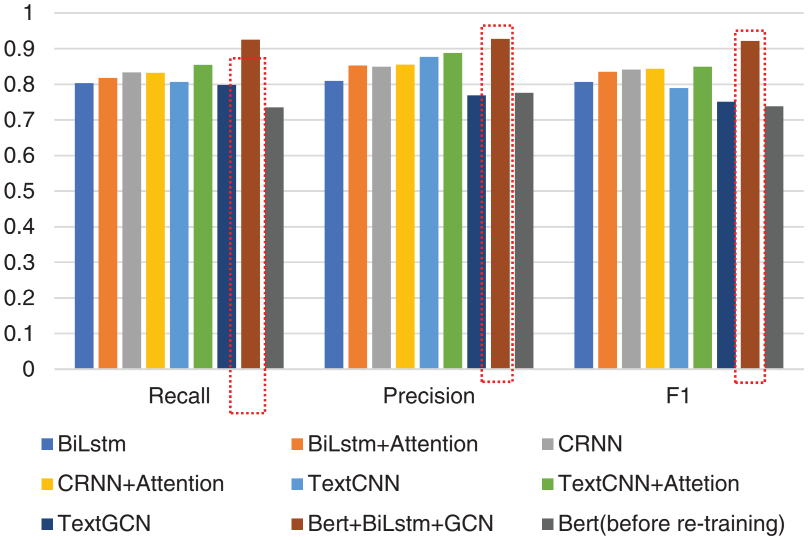

Figure 5: The comparison of experimental results for attack techniques extraction

We extended the experiment and dataset by the relationship between the threat behavior, malware, and advanced threat actors to verify the method's effectiveness in identifying malware and advanced threat actors through analyzing the threat behavior patterns used by malware and advanced threat actors. The task of threat behavior identification is transformed into malware and advanced threat actors identification, enabling mapping from threat intelligence to threat entities. The experiment results are shown in Tabs. 3 and 4 and Figs. 6 and 7. The method can enhance the identification results of malware and advanced threat actors. The F1 are 98.45% and 99.48%, which have achieved the best results for identifying malware and threat actors, which can lay the foundation for modeling threat behavior profiling for threat hunting.

Table 3: The experimental results for the identification task of software

Model/Software	Recall	Precision	F1
HT-GCN	0.9841	0.9855	0.9845

Table 4: The experimental results for the identification task of advanced threat actors

Model/Threat actors	Recall	Precision	F1
HT-GCN	0.9926	0.9994	0.9948

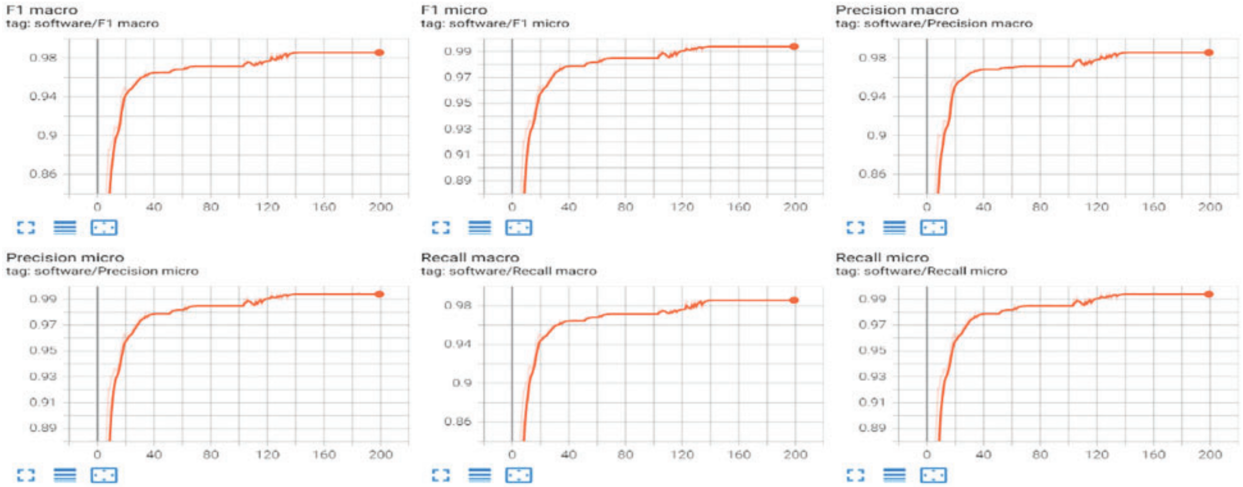

Figure 6: The experimental results graph for the identification task of software

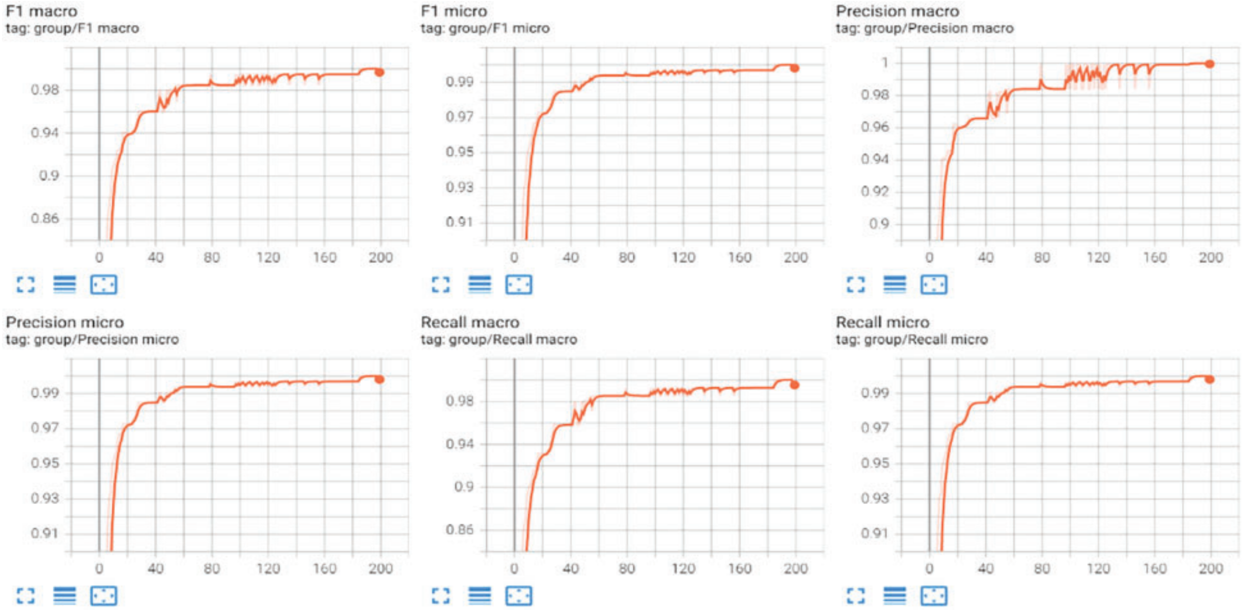

Figure 7: The experimental results graph for the identification task of advanced threat actors

5 Conclusion

Attack behavior analysis from threat intelligence is essential for APT defense. This paper proposes the knowledge extraction framework to extract the attack behavior (i.e., TTPs) and identify attack entities (i.e., malware and advanced threat actor) in cyber threat intelligence. The method provides reliability for narrowing the semantic gap between the threat intelligence text and attack behavior features, which can effectively help security analysts understand the threat behavioral patterns. In the future, we would construct more semantic feature graphs of threat behaviors and improve the model's performance for large-scale applications. We found that the attention mechanism improves the model results during our experiments. In the future work, we focus on applying the attention mechanism further to enhance the effects of threat behavior entity extraction and explore the few-shot learning method to improve the classification ability of a few categories and strengthen the generalization ability of the model minor sample problems, and added authentication to enhance the security of the usage process [42].

Funding Statement: This work was supported by China's National Key R&D Program, No. 2019QY1404, the National Natural Science Foundation of China, Grant No. U20A20161, U1836103, and the Basic Strengthening Program Project, No. 2019-JCJQ-ZD-113.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Ghafir, M. Hammoudeh, V. Přenosil, L. Han, R. Hegarty *et al.,* "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Generation Computer Systems*, vol. 89, no. 12, pp. 349–359, 2018.

[2] T. D. Wagner, K. Mahbub, E. Palomar and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," *Computers & Security*, vol. 87, no. 11, pp. 101589, 2019.

[3] P. N. Bahrami, A. Dehghanta, T. Dargahi, R. M. Parizi, K. R. Choo *et al.,* "Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures," *Journal of Information Processing Systems*, vol. 15, no. 4, pp. 865–889, 2019.

[4] A. Niakanlahiji, L. Safarnejad, R. Harper and B. T. Chu, "IoCMiner: Automatic extraction of indicators of compromise from twitter," in *Proc. of the 2019 IEEE Int. Conf. on Big Data(BD)*, Los Angeles, CA, USA, pp. 4747–4754, 2019.

[5] J. Tang, M. Qu and Q. Z. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining(KDD)*, Washington, DC, USA, pp. 1165–1174, 2015.

[6] X. R. Zhang, X. Sun, W. Sun, T. Xu and P. P. Wang, "Deformation expression of soft tissue based on BP neural network," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1041–1053, 2022.

[7] L. Luo, Z. Yang, P. Yang, Y. Zhang, L. Wang *et al.,* "An attention-based BiLSTM-CRF approach to a document-level chemical named entity recognition," *Bioinformatics*, vol. 34, no. 8, pp. 1381–1388, 2018.

[8] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang *et al.,* "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[9] R. Al-Shaer, J. M. Spring and E. Christou, "Learning the associations of Mitre ATT&CK adversarial techniques," in *Proc. of 2020 IEEE Conf. on Communications and Network Security (CNS)*, Avignon, France, IEEE, pp. 1–9, 2020.

[10] Y. Pan, T. Zhou, J. Zhu and Z. Zeng, "Construction of APT attack semantic rules based on ATT&CK," *Journal of Cyber Security*, vol. 6, no. 3, pp. 77–90, 2021.

[11] Y. Ahmed, A. T. Asyhari and M. A. Rahman, "A Cyber Kill Chain approach for detecting Advanced Persistent Threats," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2497–2513, 2021.

[12] W. Xiong, E. Legrand, O. Åberg and R. Lagerström, "Cyber security threat modeling based on the Mitre enterprise att&ck matrix," *Software and Systems Modeling*, vol. 21, no. 1, pp. 157–177, 2021.

[13] V. Legoy, M. Caselli, C. Seifert and A. Peter, "Automated retrieval of attack tactics and techniques for cyber threat reports," in *Proc. of 1st Cyber Threat Intelligence Symp.(CTI 2020)*, Zurich, Switzerland, 2020.

[14] G. Jayandhi, J. L. Jasmine and S. M. Joans, "Mammogram learning system for breast cancer diagnosis using deep learning SVM," *Computer Systems Science and Engineering*, vol. 40, no. 2, pp. 491–503, 2022.

[15] Y. P. Chang, X. L. Wang, M. H. Xue, Y. Z. Liu and F. Jiang, "Improving language translation using the hidden markov model," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3921–3931, 2021.

[16] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[17] Q. Yang, X. Wang, J. Zheng, W. Ge, M. Bai *et al.,* “LSTM android malicious behavior analysis based on feature weighting,” *KSII Transactions on Internet and Information Systems*, vol. 15, no. 6, pp. 2188–2203, 2021.

[18] X. R. Zhang, J. Zhou, W. Sun and S. K. Jha, “A lightweight CNN based on transfer learning for COVID-19 diagnosis,” *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1123–1137, 2022.

[19] Q. Zhang, X. Xiang, J. Qin, Y. Tan, Q. Liu *et al.,* “Short text entity disambiguation algorithm based on multi-word vector ensemble,” *Intelligent Automation & Soft Computing*, vol. 30, no. 1, pp. 227–241, 2021.

[20] J. Pennington, R. Socher and C. D. Manning, “Glove: Global vectors for word representation,” in *Proc. of Empirical Methods in Natural Language Processing Conf. (EMNLP 2014)*, Doha, Qatar, pp. 1532–1543, 2014.

[21] J. Devlin, M. W. Chang, K. Toutanova and K. Lee, “BERT: Pretraining of deep bidirectional transformers for language understanding,” in *Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, vol. 1, pp. 4171–4186, 2019.

[22] A. R. Abas, I. Elhenawy, M. Zidan and M. Othman, “BERT-CNN: A deep learning model for detecting emotions from text,” *Computers, Materials & Continua*, vol. 71, no. 2, pp. 2943–2961, 2022.

[23] M. Bounabi, K. Elmoutaouakil and K. Satori, “A new neutrosophic TF-IDF term weighting for text mining tasks: Text classification use case,” *International Journal of Web Information Systems*, vol. 17, no. 3, pp. 229–249, 2021.

[24] T. He, W. Huang, Y. Qiao and J. Yao, “Text-attentional convolutional neural network for scene text detection,” *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2529–2541, 2016.

[25] L. Yao, C. Mao and Y. Luo, “Graph convolutional networks for text classification,” in *Proc. of the 33rd AAAI Conf. on Artificial Intelligence(AAAI 2019)*, Hawaiian, HI, USA, 33, pp. 7370–7377, 2019.

[26] K. Zhang, H. Zhang, Q. Liu, H. Zhao, H. Zhu *et al.,* “Interactive attention transfer network for cross-domain sentiment classification,” in *Proc. of the 33rd AAAI Conf. on Artificial Intelligence(AAAI 2019)*, Hawaiian, HI, 33, pp. 5773–5780, 2019.

[27] X. L. Li, K. Yuan, X. F. Wang, Z. li, L. y. Xing *et al.,* “Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence,” in *Proc. of the 23th ACM Conf. on Computer and Communications Security(CCCS)*, Vienna, Austria, 24–28, pp. 755–766, 2016.

[28] G. Husari, E. AlShaer, M. Ahmed, B. Chu and X. Niu, “TTPDrill: Automatic and accurate extraction of threat actions from unstructured text of CTI Sources,” in *Proc. of the 33rd Annual Computer Security Applications Conf.(CSAC)*, Orlando, FL, USA, 132521, pp. 103–115, 2017.

[29] Y. Bengio, A. Courville and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[30] J. Zhao, Q. Yan, J. Li, M. Shao, Z. He *et al.,* “TIMiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data,” *Computer&Security*, vol. 95, no. 8, pp. 101867, 2020.

[31] J. Zhao, X. Liu, Q. Yan, B. Li, M. Shao *et al.,* “Automatically predicting cyber-attack preference with attributed heterogeneous attention networks and transductive learning,” *Computer&Security*, vol. 102, no. 3, pp. 102152, 2021.

[32] L. Deng, X. L. Wang, F. Jiang and R. Doss, “EEG-based emotion recognition via capsule network with channel-wise attention and LSTM models,” *Transactions on Pervasive Computing and Interaction*, vol. 3, no. 4, pp. 425–435, 2021.

[33] W. Sun, G. Z. Dai, X. R. Zhang, X. Z. He and X. Chen, “TBE-Net: A three-branch embedding network with the part-aware ability and feature complimentary learning for vehicle reidentification,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021. https://doi.org/10.1109/TITS.2021.3130403.

[34] X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun and S. K. Jha, “A robust 3-D medical watermarking based on wavelet transform for data protection,” *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.

[35] X. R. Zhang, X. Sun, X. M. Sun, W. Sun and S. K. Jha, “Robust reversible audio watermarking scheme for telemedicine and privacy protection,” *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.

[36] M. Schlichtkrull, T. Kipf, P. Bloem, R. V. Berg, I. Titov *et al.,* “Modeling relational data with Graph Convolutional Networks,” *Lecture Notes in Computer Science*, vol. 10843, pp. 593–607, 2018.

[37] W. Sun, L. Dai, X. R. Zhang, P. S. Chang, X. Z. He *et al.,* “Real-time small object detection algorithm in UAV-based traffic monitoring,” *Applied Intelligence*, vol. 52, pp. 1–16, 2021.

[38] Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang *et al.,* “BertGCN: Transductive text classification by combining GNN and BERT,” *Findings of the Association for Computational Linguistics (ACL-IJCNLP 2021)*, Online, pp. 1456–1462, 2021.

[39] X. GAO, J. Yu and S. W. Xu, “Text classification study based on graph convolutional neural networks,” in *Proc. of the 2021 Int. Conf. on Internet, Education and Information Technology (IEIT)*, Suzhou, China, pp. 102–105, 2021.

[40] J. Zhang, J. Liu and X. Lin, “Improve neural machine translation by building word vector with part of speech,” *Journal on Artificial Intelligence*, vol. 2, no. 2, pp. 79–88, 2020.

[41] S. Cao, X. Sun, L. Bo, Y. Wei and B. li, “BGNN4VD: Constructing bidirectional graph neural-network for vulnerability detection,” *Information and Software Technology*, vol. 136, no. 1, pp. 106576, 2021.

[42] Xl Wang, X. She, L. Bai, Q. Yang and F. Jiang, “A novel anonymous authentication scheme based on edge computing in VANETs,” *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3349–3361, 2021.