IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 28, 2021, accepted July 14, 2021, date of publication July 26, 2021, date of current version July 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3099489

Security Enhanced Sentence Similarity Computing Model Based on Convolutional Neural Network

QIFENG SUN[1], XINGZHE HUANG[1], GODFREY KIBALYA[2], NEERAJ KUMAR[3,4,5], (Senior Member, IEEE), SANTHOSH KUMAR S. V. N.[6], PEIYING ZHANG[1,7], AND DONGLIANG XIE[7], (Member, IEEE)

[1]College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China
[2]Department of Network Engineering, Technical University of Catalonia (UPC), 08034 Barcelona, Spain
[3]Department of Computer Science and Engineering, Thapar University, Patiala 147004, India
[4]Department of Computer Science and Information Engineering, Asia University, Taichung 41354, Taiwan
[5]School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, India
[6]School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India
[7]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding authors: Peiying Zhang (zhangpeiying@upc.edu.cn), Godfrey Kibalya (godfrey.mirondo.kibalya@upc.edu), and Qifeng Sun (sunqf_upc@163.com)

This work was supported in part by the Major Scientific and Technological Projects of China National Petroleum Corporation (CNPC) under Grant ZD2019-183-006, in part by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2020MF006, in part by the Fundamental Research Funds for the Central Universities of China University of Petroleum (East China) under Grant 20CX05017A, and in part by the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) under Grant SKLNST-2021-1-17.

ABSTRACT Deep learning model shows great advantages in various fields. However, researchers pay attention to how to improve the accuracy of the model, while ignoring the security considerations. The problem of controlling the judgment result of deep learning model by attack examples and then affecting the system decision-making is gradually exposed. In order to improve the security of sentence similarity analysis model, we propose a convolution neural network model based on attention mechanism. First of all, the mutual information between sentences is correlated by attention weighting. Then, it is input into improved convolutional neural network. In addition, we add attack examples to the input, which is generated by the firefly algorithm. In the attack example, we replace the words in the sentence to some extent, which results in the adversarial data with great semantic change but slight sentence structure change. To a certain extent, the addition of attack example increases the ability of model to identify adversarial data and improves the robustness of the model. Experimental results show that the accuracy, recall rate and F1 value of the model are due to other baseline models.

INDEX TERMS Security enhancement mechanism, attack examples, convolutional neural network, attention mechanism, sentence similarity.

I. INTRODUCTION

Deep learning model shows great advantages in various fields, including computer vision, classification system, prediction model and so on [1]–[6]. In the past work, researchers have devoted themselves to improving the accuracy of the model. With the application of high-precision model in various fields, the security problem of the model is more and more prominent. While powerful, the neural network methods exhibit a rather strong barrier of entry, for various security reasons. The attacker can influence the decision-making result of the model by changing the input samples slightly, and then affect the further decision-making of the system. As a basic task of natural language processing, sentence similarity analysis is widely used in machine translation [7]–[9], document duplicate checking [10]–[12] and simultaneous interpretation systems [13]–[15]. In addition, it is also widely used in social information processing and intention detection. In machine translation, the system evaluates the semantic similarity according to the user's input, and then matches the corresponding translation. In the previous work, researchers mostly focused on how to improve the accuracy of model classification or evaluation results. In the task of sentence pair classification, the model can roughly

The associate editor coordinating the review of this manuscript and approving it for publication was Manuel Rosa-Zurera.

classify the text according to its basic semantics. In the model which needs higher accuracy, sentence similarity is evaluated by giving similarity score. However, these studies are based on the model to improve the accuracy of sentence classification, lacking of consideration of model security. The attacker influences the classification result of the model by injecting counter samples, which will also reduce the accuracy of the model. In the computer vision model [16]–[19], the calculation results of the model are affected by the injection of adversarial examples. This security problem also exists in the field of natural language processing. In recent years, researchers have proposed some text sentiment classification models based on adversarial [20]–[23] to enhance the robustness of the model. For improving the security of the model, this paper proposes a method to identify such counterwork cases, which is to leverage on hybrid attention mechanism and counterwork neural network model. We propose a model combining attention mechanism and adversarial convolution neural network for sentence similarity analysis. The main contributions are summarized as follows:

1. In the task of sentence similarity analysis, the intrusion of adversarial sample always imposes some key security obstacles and then attacks the model. Based on this very situation, we propose a multi feature model to extract feature information from sentences. The multi feature model considers both semantic information and location information of sentence words. When dealing with confrontation samples, the multi feature model can identify subtle changes in sentences by calculating the similarity of words. In addition, it can also be used to extract the location information for the counter samples. In this paper, the proposed multi feature model can fully extract sentence features and also complete the recognition of adversarial examples.

2. In order to improve the security of the model, we integrate the training of adversarial examples. The generation of adversarial examples is completed by population iteration of firefly algorithm. Aiming at the improvement on the interaction between adversarial examples and deep learning model, the text generation process is combined with deep learning model through conditional judgment mechanism. The resistance samples generated by the deep learning model will be tested until the specified conditions are met. This mechanism enables the deep learning model to capture the characteristics of adversarial examples.

3. In addition, we add countermeasure dropout layer to convolutional neural network. The generated adversarial texts, as adversarial attacks samples, will be gathered with original texts to complete adversarial training for semi-supervised learning. Because the adversarial example is a kind of wrong sample to some extent, this kind of sample destroys the original inherent law of the text. Therefore, the accuracy of model training will be affected by the addition of adversarial examples. In order to solve this problem, we add a loss layer to the convolutional neural network. This kind of resistance loss is used to calculate the loss value of the model for the counter sample. The experimental results show that the model has better performance in the analysis of counter samples.

4. In this paper, we propose an interactive method to generate adversarial examples. This method takes the evaluation results of the model into account in the iterative process of genetic algorithm. In addition, in order to ensure the change rate of adversarial examples, we set the change rate parameters to find the balance between the adversarial examples and the deep learning model. The interactive adversarial examples generation method can ensure the change rate of samples and the performance of the model, which is of great significance for training high security model in practical application.

The next chapter is related works. The third chapter is the proposed model, including multi feature attention model, the generation of adversarial examples based on genetic algorithm and the counter convolution neural network model. In the fourth chapter, we verify the performance of the model through experiments, including the accuracy of sentence similarity analysis and the security of dealing with adversarial examples. In the fifth chapter, we summarize our work.

II. RELATED WORKS

Sentence similarity analysis [24]–[26], as a basic task of natural language processing [27]–[31], is widely used in the field of natural language processing. At present, many applications including machine translation [32]–[34], intelligent voice customer service [35]–[38], intelligent chat [39], [40] are based on sentence similarity analysis. In the past, the main goal is to improve the accuracy of sentence similarity analysis model. Researchers proposed a sentence similarity analysis model based on multiple features [41], [42], including the semantic, grammatical and sentence length features of sentence pairs. With the rapid development of deep learning model, neural networks are the driving force behind state-of-the-art algorithms for machine translation, syntactic parsing, and many other applications. While powerful, the security of the neural network model has been ignored. Therefore, many researchers began to improve the accuracy of the model while incorporating security considerations. At present, considering the accuracy and security of sentence similarity model, the model can be divided into the following three categories: sentence similarity calculation model based on multi features, sentence similarity calculation model based on deep learning and sentence deep learning model based on anti-network and heuristic algorithm. The model based on multi feature and deep learning studies the accuracy of the model. Based on the model of antagonism neural network and heuristic algorithm, the security problem is studied. Fraud attacks the deep neural network model with imperceptible adversarial examples, which requires the model to have the

ability to deal with the adversarial examples. There are two kinds of defense methods, one is to detect the adversarial examples directly, the other is to enhance the robustness of the deep neural network. In the following content, we introduce the extraction of sentence features and the security of the model.

A. THE COMPUTATION METHODS OF SENTENCE SIMILARITY BASED ON MULTI-FEATURES

Recent literature on sentence similarity has shown abundantly proposed methods [43]. Most research studies exploited semantic similarity between two words for measuring how similar two input sentences are. Some of them utilized bag-of-words techniques to calculate sentence similarity. These bag-of-words techniques based on the assumption that the more similar two sentences are the more same words they share. In literature [44], the authors proposed a method that based on semantic dependency relationship analysis to compute sentence similarity, the method took advantage of semantic level and dependency syntactic level to measure the sentence similarity, and experiment result of this method is satisfied. Islam and Inkpen [45] presented a method for measuring the semantic similarity of texts using a corpus-base measure of semantic word similarity and a normalized and modified version of the Longest Common Subsequence (LCS) string matching algorithm. Li *et al.* [46] proposed a method that combines word-to-word semantic similarity and word order similarity for measuring sentence similarity. It keeps all function words and weights significance of each word by its information content derived from Brown Corpus. Aliguliyev [47] put forward a method for sentence similarity computation by integrating multi-features. This approach computes the sentence similarity by endowing the syntax feature, semantic feature and word feature of the sentences with different weights. Dan *et al.* [48] introduced a method for assessing the semantic similarity between sentences, which relies on the assumption that the meaning of a sentence is captured by its syntactic constituents and the dependencies between them. Nguyen *et al.* [49] proposed a novel method for measuring semantic similarity between two sentences. The method mainly takes advantage of syntactic and semantic features to assess the similarity. Reference [50] proposed a more comprehensive model of sentence feature extraction, which collects the semantic information, syntactic information and word order information of sentences at the same time. When extracting syntactic information, the model only extracts the subject, predicate and object of the sentence to construct the sentence matrix, which leads to incomplete feature extraction of the sentence. However, if all the information in the sentence is considered comprehensively, the main features of the sentence cannot be highlighted. In later sections, we address this shortcoming by using attention mechanisms. In literature [51], the authors proposed a sentence similarity analysis model based on multi head attention mechanism. The model makes a combination between deep learning model and attention mechanism. In subsequent chapters, the performance of this model is proved to be outstanding.

B. THE COMPUTATION METHODS OF SENTENCE SIMILARITY BASED ON DEEP LEARNING

Neural network provides a powerful learning mechanism. It is very attractive in dealing with natural language processing problems. It also improves the research progress, especially the integration of attention mechanism makes the analysis accuracy further improved. Most of the neural models used in natural language processing researches can be divided into two types, convolution neural network and short-term memory neural network. Convolution neural network is able to fully extract the features of high-dimensional word vectors through multi-layer convolution and pooling. Short-term and long-term memory neural network has a relatively simple neuron, which simplifies the training parameters and retains the dependency between words in sentences. The deep learning model is mainly about the vector transformation and feature extraction of features in sentences. At present, word2vec is the main tool to transform the semantic units of sentences into the vectors. Through the training of this model, the vector table is obtained, which is mapped to the form of high-dimensional matrix after the segmentation of sentences. In reference [52], the authors propose a sentence similarity calculation method based on convolution neural network. In this model, the feature matrix of a sentence is extracted, and the syntactic structure of a sentence is considered in addition to the word vector. The semantic information of a sentence is mostly contained in its subject predicate and object. Through the part of speech tagging of sentence segmentation and sentence components, words without practical meaning are filtered and input to neural network to calculate the similarity of sentence pairs. The literature [53] proposed a sentence similarity analysis model based on convolution neural network. The algorithm uses convolution layer and pooling layer to extract and reduce the dimension of sentence feature matrix. Based on the characteristics of convolutional neural network, the model uses 300 dimensional high-dimensional word vectors. High dimensional word vectors make it contain more detailed features of sentence alignment. Therefore, the model has achieved good performance. What's more, Long Short-Term Memory (LSTM) has been shown to perform more outstanding than RNNs on tasks involving long time lags. The authors of [54] propose to use twin LSTM network to measure the similarity of semantic text. LSTM has been proved easily learns to instantiate a counter, and by combining two counters it can even learn a simple context-sensitive language. The traditional model ignores the correlation of mutual information such as synonyms and co-occurrence words between sentence pairs. Attention mechanism realizes the semantic association of sentence pairs before the input deep learning model by weighting the semantic units after segmentation according to the features.

C. THE COMPUTATION METHODS OF SENTENCE SIMILARITY BASED ON ADVERSARIAL NETWORK AND HEURISTIC ALGORITHM

Although deep learning model improves the accuracy of sentence similarity analysis model, the security of neural network model is also revealed. The attack on the model can be divided into black box attack and white box attack. The black box attackers know nothing about the internal structure, training parameters, and defense methods of the attacked model, and can only interact with the model through the output. White box attack has mastered the network structure and output of deep neural network. Some aggressive samples induce the model to make wrong judgment by making some subtle changes to the input data. In order to improve the security of deep learning model, the researchers have proposed the adversarial examples and adversarial network model. In order to reduce the fitting degree of the model, the countermeasure samples do not change the structure of the network but only change the content of the input data. Adversarial network is to add adversary neural network to the traditional network structure to learn the characteristics of adversary samples so as to realize the differentiation of adversary text. In reference [55], researchers proposed an algorithm for text generation based on genetic algorithm. Through the crossover, mutation and genetic operation of genetic algorithm, the original text content is replaced to get the counter sample. This method can reduce the impact on the text semantics by replacing the synonyms in the text. However, there are many grammatical and temporal errors in the replaced adversarial examples. In addition, this kind of word change cannot use gradient information to generate disturbance efficiently. The key to counter samples is how to reduce the impact on semantics as much as possible to cheat deep learning model. It is difficult for the model based on heuristic algorithm to influence the judgment of the model on the gradient. In reference [56], a text classification model based on confrontation training is proposed, which can reduce the fitting degree and improve the security performance by changing the data of the input depth learning model. However, this method not only improves the security, but also reduces the accuracy of model analysis. Ali *et al.* [57] proposed a text sentiment analysis model based on anti-dropout, which includes two kinds of short-term and long-term memory neural networks. One is the LSTM model based on random dropout, the other is based on the model against dropout. The authors of [58] proposed an improved attention mechanism model (MAN) which makes full use of the bidirectional long short-term memory (Bi-LSTM) network, and which learns the sentiment polarities of aspect terms in sentences by proposing the mutual attention mechanism. And, the MAN has shown the promising performance in terms of aspect-level sentiment classification. What's more, Long Short-Term Memory (LSTM) has been shown to perform more outstanding than RNNs on tasks involving long time lags. Compared with the network of adversary training, adversarial dropout model not only improves the security of deep learning model, but also ensures the accuracy of model calculation.

III. THE PROPOSED MODEL

In this paper, we propose an adversarial convolution neural network model based on attention mechanism, which can deal with the security of adversarial modulation by adding adversarial mechanism to convolution neural network. First of all, the attention mechanism is used to extract mutual information in sentences. Genetic algorithm is used to generate adversarial examples. The original training samples and the adversarial examples are used in the training process. In the convolution neural network, we add the adversarial dropout layer. The concept of anti-dropout is first proposed in the field of image processing. Different from the countermeasure training, it does not change the input of data, but prevents over fitting by operating the hidden layer. In order to improve the security of the model, the more sparse network can be obtained by antagonizing the neurons in the hidden layer. In the following chapters, the detailed description of the generation of adversarial examples and the structure of adversarial convolution neural network can be found.

A. GENERATION OF COUNTERMEASURE SAMPLES BASED ON GENETIC ALGORITHM

1) GENERATION OF INITIAL POPULATION

The initial population is the starting point of genetic algorithm for genetic, crossover and mutation. To some extent, the quality of the initial total group determines the number of iterations of the algorithm. First, select the initial samples to attack. Select n pairs of sentences from the corpus, and the original sentence pairs can be evaluated by the model. The basic semantic unit of a sentence is obtained after word segmentation. Use word2vec to get the vector representation of words. Because the adversarial text we want to generate needs to ensure that the changes of sentences are as small as possible, so we only replace the synonyms of sentences. Calculate the similarity between the segmented words and the words in the dictionary to obtain the synonym list. The initial iterative population of genetic algorithm is obtained by synonymous substitution of words in sentences. In Algorithm 1, we describe the process of constructing initial population.

2) STOP CONDITION OF GENETIC ALGORITHM

Adversarial examples are attack data used to train neural networks. From the point of view of model security, our aim is to make the model classify the adversarial examples correctly. Therefore, when using genetic algorithm to iterate the initial population, the results of each iteration are input into the attacked model to calculate the accuracy of the model against the sample similarity calculation. In addition, in order to ensure the change rate of the adversarial examples, the initial text population has been iterated enough

Algorithm 1 *Population Generation Algorithm*

Input: *Selected original sentence sequence text*[] *Similarity score of sentence sequence sta_score*[]
Output: *Initial adversarial text Change rate of sentences ratio*[]
text[] *sta_score*[] *wordvec*[] *synonym*[]
Segmentation, part of speech tagging and mapping of the original sentence
for *i in text*[] **do**
  get j from wordvec[]
  if $cos(i, j) > threshod1$ **then**
    synonym.append(j)
  end if
end for
for *m in text*[] **do**
  Replace text with synonyms
end for
ratio=revisedWord/totalWord
ratio.append(ratio)
return [adversarial text, ratio[]]

times. Therefore, the threshold value is set for the change rate of countermeasure samples to control the quality of the generated countermeasure samples. In algorithm 2, we describe the termination conditions of genetic algorithm in detail. Among them, *threshod*1 represents the threshold of accuracy rate, *threshod*2 represents the threshold of revised rate.

Algorithm 2 *Iterative Termination Algorithm*

Input: adversarial samples generated in the last iteration
while *accuracy* $>$ *threshod*2 *and revisedrate* $>=$ *threshod*3 **do**
  Input the sample to the adversarial convolution neural network
  Obtain the similarity calculation result of model output
  Calculation accuracy
  Calculate the revised rate of text
end while

3) GENETIC PROCESS

Fitness is used to measure the individual's adaptability to the environment after crossing, inheritance and variation. Individuals with high fitness are more likely to be retained in the next iteration, while individuals with low fitness are eliminated in the next iteration. We measure the quality of the generated adversarial text from two aspects: the similarity score given by the model and the change rate of the text. The calculation formula of fitness is as follows.

$$Fit(x_i) = \frac{\alpha}{|sta\_score - Jud\_score|} + \frac{(1-\alpha) * revWords}{totalWords}, \quad (1)$$

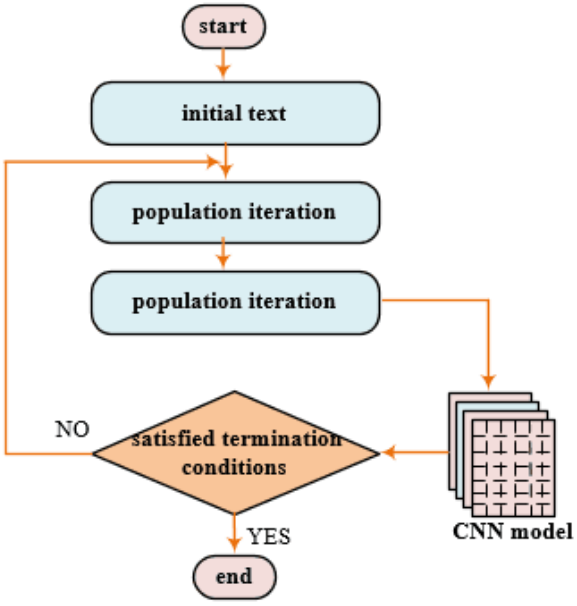

FIGURE 1. Principle of attention mechanism.

where, x_i represents the individual of the adversarial text, and α is the coefficient between 0 and 1. revisedWords is the number of words replaced in individual x_i. *totalWords* is the total number of words in an individual. *sta_score* score is the similarity score given in the data set. *Jud_score* is the similarity score given by the model.

After calculating the fitness of individuals in the population, the population is selected. In the algorithm, we use fitness as the only index of population selection. The individuals whose fitness is lower than the threshold are eliminated by setting the fitness threshold. After the winning individuals are preserved, the change rate of individuals and the part of speech of substitutes are recorded, and the part of speech of synonyms is inherited to the next population as an excellent gene. The cross process between individuals in a population is completed by setting random numbers. Some individuals in the population complete the cross process by learning each other's position and part of speech. The process of population variation is completed by replacing different parts of speech and adjusting the change rate up and down. Through the cross, heredity and variation among individuals in the population, the generation of adversarial text is completed. The process of text generation based on genetic algorithm illustrated in FIGURE 1.

B. AN ADVERSARIAL CONVOLUTION NEURAL NETWORK MODEL BASED ON ATTENTION MECHANISM

1) SENTENCE PAIR MUTUAL INFORMATION EXTRACTION

The traditional sentence pair similarity analysis model lacks the consideration of mutual information between sentence pairs. In the analysis of sentence pair similarity, the common words, semantic relations and position relations among the common words will have a great impact on sentence seman-

tics. The proposed model weights the mutual information of sentences before the pair is input into the anti-convolution neural network. In this way, more attention can be paid to the key information when the neural network extracts sentence features. In the extraction of sentence mutual information, we mainly consider word2vec word vector embedding and co-occurrence word position information embedding.

2) WORD VECTOR EMBEDDING

After the word segmentation mapping of sentences, the eigenvector representation is obtained. $sentence_A = \{w_1, w_2, w_3, \ldots, w_n\}$, $sentence_B = \{w'_1, w'_2, w'_3, \ldots, w'_m\}$, where n and m represent the sentence length of $sentence_A$ and $sentence_B$, respectively. Calculate the cosine distance of the basic semantic units between sentence pairs respectively, and the summation calculation method is shown in formula 2. Summation is to sum the cosine similarity of words in different positions of two sentences. Among them, the cosine similarity calculation formula is as formula 3.

$$w2v\_embedding = COS(w_i, w'_j), \tag{2}$$

$$cos(x_i, y_i) = \frac{\sum_i^n (x_i \times y_i)}{\sqrt{\sum_i^n x_i^2} \times \sqrt{\sum_i^n y_i^2}} \tag{3}$$

where, w_i and w'_j are the words in the sentence pair. According to formula 2, the word vector matrix of sentence pairs is calculated. The calculation method is shown in formula 4.

$$w2v\_matrix = \begin{bmatrix} COS(w_1, w'_1), & \ldots, & COS(w_1, w'_m) \\ COS(w_2, w'_1), & \ldots, & COS(w_2, w'_m) \\ COS(w_3, w'_1), & \ldots, & COS(w_3, w'_m) \\ \ldots, & \ldots, & \ldots \\ COS(w_n, w'_1), & \ldots, & COS(w_n, w'_m) \end{bmatrix}, \tag{4}$$

After the weight matrix between sentence pairs is obtained, the weight vector of each semantic unit in $sentence_A$ relative to $sentence_B$ is calculated by summing the row elements of the matrix. Sum the matrix column elements and calculate the weight vector of each semantic unit in $sentence_B$ relative to $sentence_A$.

3) POSITION INFORMATION EMBEDDING

Word vector embedding of word2vec takes into account semantic similarity, context, and structural information of the text. In addition, the number of words in a sentence and the corresponding position also have an impact on semantic changes. Position embedding generates a position embedding weight matrix based on the edit distance of words in the text. First, the co-occurrence words in the text are retrieved globally and a co-occurrence word set is generated $set_{comWord} = \{w_1^c, w_2^c, \ldots, w_k^c\}$, where k represents the number of co-occurrence words in the sentence, $w_i^c \in set(A) \cap set(B)$. Then, the position information of co-occurrence words w_k^c in $sentence_A$ is retrieved in the text pair, which is recorded as $loc_A(w_k^c)$, and the position information in $sentence_B$ is recorded as $loc_B(w_k^c)$. Finally, the position information is used as the index to obtain the words corresponding to the position of the text pair, and the edit distance between the words and the co-occurrence words is calculated to generate the position embedding weight matrix based on the edit distance as shown in fomula 5.

$$pos\_embedding = \begin{cases} \dfrac{2 * Ed(w_k^c, w'_{loc_A(w_k^c)})}{\min\{l(A), l(B)\}}, & loc_B(w_k^c) \le l(A) \\ \dfrac{2 * Ed(w_k^c, w'_{loc_B(w_k^c)})}{\min\{l(A), l(B)\}}, & loc_A(w_k^c) \le l(B) \end{cases} \tag{5}$$

where, the Ed represents the edit distance, which is defined as the minimum number of insertion, deletion and replacement of two adjacent basic units needed to convert one string to another. In other words, the edit distance refers to the minimum number of edit operations required between two strings to change from one to the other.

4) MULTI FEATURE ATTENTION MATRIX

After calculating the weight vector based on word2vec embedding and the position embedding vector based on edit distance, the row vector and column vector are added respectively, and the probability is normalized by softmax function. The input of the neural network is obtained by weighting the matrix of the text with the normalized vector. See formula 6 and formula 7 for calculation.

$$Att\_Matrix_{sentence_A} = softmax(row\_vec + pos\_row) * [w_1, w_2, \ldots, w_n], \tag{6}$$

$$Att\_Matrix_{sentence_B} = softmax(col\_vec + pos\_col) * [w'_1, w'_2, \ldots, w'_n]^T, \tag{7}$$

where, $Att\_Matrix_{sentence_A}$ and $Att\_Matrix_{sentence_B}$ respectively represent the multi feature attention sentence matrix after weighting the mutual information in the sentence. In Algorithm 3, we describe the construction process of the attention matrix in detail. In algorithm 3, we describe the construction process of the attention matrix in detail.

C. ADVERSARIAL CONVOLUTIONAL NEURAL NETWORK

After obtaining the multi feature attention matrix of a sentence, we need to further process the real number matrix which contains a lot of information in the sentence. In our proposed model, the anti-training and the anti-dropout layer are added to the traditional convolutional neural network, which improves the robustness of the model and prevent over fitting. Note that the proposed adversarial neural network has the same structure as the traditional neural network in this paper. The difference is that they have different training samples. In the adversarial neural network, we input the adversarial examples generated by genetic algorithm. In the following chapters, we first introduce the structure of convolutional neural network, countermeasure training and countermeasure dropout layer.

Traditional convolution neural network includes input layer, convolution layer, pooling layer and output layer. In our

Algorithm 3 Multi Feature Matrix Generation Algorithm

Input: $sentence_A = \{w_1, w_2, w_3, \ldots, w_m\}$, $sentence_B = \{w'_1, w'_2, w'_3, \ldots, w'_m\}$,$set_{comWord} = \{w^c_1, w^c_2, \ldots, w^c_k\}$
Output: Att_matrix_A, Att_matrix_B
for *i in* $sentence_A$ **do**
  for *j in* $sentence_B$ **do**
    $w2v\_matrix.append(cos(w_1, w'_1))$
  end for
end for
for *m in* $set_{comWord}$ **do**
  Calculate the pos_embedding matrix according to formula (4)
end for
Sum the row_elements of w2v_matrix and pos_embedding to get row_vec and pos_row
Sum the elements of w2v_matrix and pos_embedding column to get col_vec and pos_col
Calculate Att_matrix_A *and* Att_matrix_B *according to formula* (6) *and* (7)
return $[Att\_matrix_A, Att\_matrix_B]$

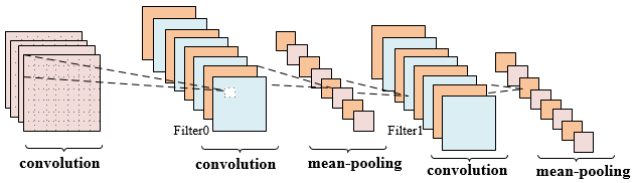

FIGURE 2. The structure of convolutional neural network.

model, the model results construct according to the matrix dimension features we input are shown in FIGURE 2. The model consists of two layers of convolution, each of which consists of eight convolution kernels. Because the special information in the feature matrix is scattered, we choose the average pooling method to extract sentence features as scattered as possible. The step size of the convolution operation is set to 2. In order to extract the edge information of matrix better, the form of 0 filling is used in the convolution process.

1) ADVERSARIAL TRAINING

Adversarial training is a regularization method, which can effectively reduce the fitting degree of the model and prevent over fitting phenomenon. In our model, the text generated by genetic algorithm is used to realize the training of confrontation. Confrontation training makes the model not only consider the grammatical structure of words, but also the semantic information of words, which can better distinguish sentence features and improve the quality of classification model. As the same as the traditional training method, the adversarial training uses the back propagation algorithm to update the adversarial training according to the classification loss. The calculation formula of adversarial training is as follows.

$$\gamma_{ap} = -\beta g / ||g||_2, \tag{8}$$

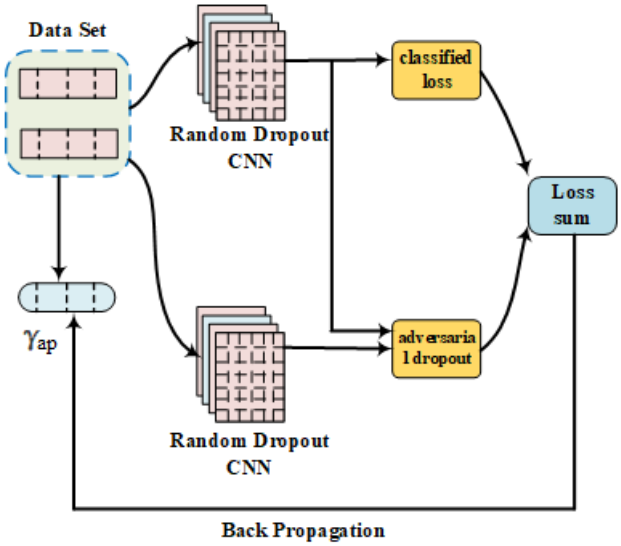

FIGURE 3. Adversarial training process.

where, g is the gradient of back propagation. γ_{ap} is the calculated disturbance term. The negative gradient direction is the direction in which the loss of the model decreases the fastest. In order to make the disturbance change the output of the model to the greatest extent, the positive gradient direction is the direction in which the disturbance is the largest. After the disturbance is added, the gradient rises and the disturbance γ_{ap} is updated. In order to show the training process of confrontation training samples more specifically, we further describe it in FIGURE 3.

2) ADVERSARIAL DROPOUT

Different from adversarial training, adversarial dropout does not change the input of CNN. It is realized by selectively discarding or masking hidden neurons according to the characteristics of adversarial training. Adversarial dropout is a regularization method, which can get more sparse network and improve the robustness of the model. Its structure is shown in FIGURE 4. In adversarial loss, neural network calculates the hiding degree of nodes according to the loss mechanism, KL divergence is calculated by text feature matrix and label, and the state of hidden layer is updated by back propagation algorithm according to KL divergence. We use Jacobi formula to calculate the hiding degree of nodes.

$$i = S_{CNN} \cdot \bigtriangledown_{S_{CNN}} KL[p(y|\overrightarrow{v}, \varepsilon_\delta, \theta)||p(y|\overrightarrow{v}, \varepsilon, \theta)], \tag{9}$$

S_{CNN} is the state of neurons in the convolution layer. ε_δ is the initial value of node hiding degree, ε is the value of random masking, and θ is the parameter of the model.

Then, according to the result of Formula 9, it is decided whether to hide the node of the convolution layer. K is used to represent the k-th convolution layer in the model, i is used to represent the i-th convolution unit of each convolution layer,

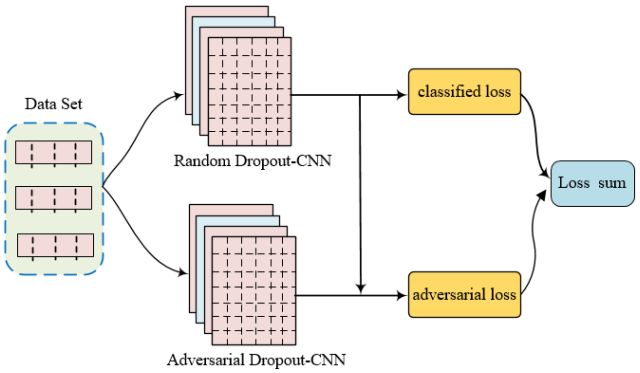

FIGURE 4. Adversarial dropout process.

and ε is used to represent whether each unit is hidden or not.

$$\varepsilon^{k,i} = \begin{cases} 0, & i_{k,i} < 0 \\ 1, & i_{k,i} > 0 \end{cases} \tag{10}$$

The calculation formula of node state of convolution layer is as follows.

$$S_{CNN} = S'_{CNN} \cdot \varepsilon, \tag{11}$$

3) OVERALL STRUCTURE OF THE MODEL

In this paper, we propose a mechanism to improve the security and robustness of the model by adding anti-training on the basis of the traditional convolutional neural network. The structure of the model, the idea of anti-training and the anti-dropout have been described in the above chapters. This section mainly introduces the combination of the model against training and against dropout. The idea of confrontation training comes from the field of image processing. It attacks the deep learning model by adding small disturbances to the image samples. In the same way as in the field of image processing, we use genetic algorithm to generate countermeasure samples, which have subtle disturbance. Compared with the original word vector, if the word vector is not homogenized, the subtle disturbance is likely to be covered. Therefore, we normalize the word vectors against the sample map. As shown in the formula below.

$$\bar{Sen}_{vec} = \frac{Sen_{vec} - E(vec)}{\sqrt{var(vec)}}, \tag{12}$$

$$E(vec) = \sum_{i=1}^{k} f_i * v_i, \tag{13}$$

$$var(vec) = \sum_{i=1}^{k} f_i (v_i - E(vec))^2, \tag{14}$$

where, Sen_{vec} represents the original sentence vector mapped out against the sample. $\bar{Sen}_{vec}$ represents the sentence vector after the probability normalization of the value of the counter sample. v_i represents the i-th word, and f_i represents the corresponding frequency of the word.

In the part of confrontation training, the antagonism samples generated by genetic algorithm are mapped to obtain the antagonism vector. Then the word vector is weighted by multi feature attention. Finally, the adversary vector and the original word vector are input into the adversary dropout and roll into the neural network respectively, and the original feature matrix and the adversary feature matrix are obtained as the input of softmax layer respectively. According to the similarity value and tag value given by the model, the classification loss is calculated, and the calculation method is shown in formula 15.

$$class_{loss} = \sum_{i=1}^{N} y_i \times log\hat{y}_i + (1 - y_i) \times log(1 - \hat{y}_i), \tag{15}$$

where, y_i represents the similarity value given in the database. $\hat{y}_i$ represents the forecast category given by the model. N represents the total number of sentence pairs entered into the model. Then, the sentence feature matrix and label of the confrontation training are used as the input of the softmax classifier to calculate the loss value of the confrontation training. The calculation of the loss function is shown in formula 16.

$$adversarial_{loss} = -\frac{1}{N} \sum_{i=1}^{N} logp(y_n | s_n + r'_{ap}; \epsilon'_{ad}), \tag{16}$$

where, r'_{ap} represents anti disturbance, N represents the total number of samples, s_n represents the word vector set of input text, and y_n is the standard similarity of input sentences.

In the adversarial dropout part of convolutional neural network, the parameters of adversarial dropout are calculated according to the random shadowing, and then the hidden layer state of the anti-convolutional neural network is updated according to the anti-shadowing. The last hidden layer state is the feature of sentence pair. The text feature and real tag are input into the softmax classifier to calculate the loss of adversarial dropout. Its loss function is shown in formula 17.

$$loss_{dropout} = -\frac{1}{N} \sum_{i=1}^{N} logp(y_n | \epsilon'_{ad}; s_n + r'_{ap}, \theta), \tag{17}$$

where, r'_{ap} represents anti disturbance, N represents the total number of samples, s_n represents the set of input sentence pairs, and y_n represents the similarity value corresponding to the input text. θ represents the parameters of the model. ϵ'_{ad} stands for confrontational masking.

In the above content, the classification loss, the countermeasure training loss and the countermeasure dropout loss of the model have been introduced respectively. In the process of training, the deep learning model can only descend along one optimization direction. Therefore, we have carried on the weighted summation processing to these losses. The calculation method is shown in formula 18.

$$total_{loss} = \alpha class_{loss} + \beta adversarial_{loss} + \gamma loss_{dropout}, \tag{18}$$

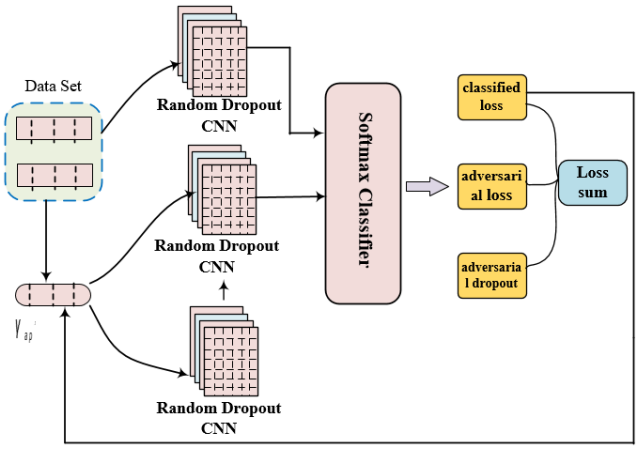

FIGURE 5. Adversarial CNN.

where, α, β, γ is the weight coefficient of classification loss, countermeasure training loss and countermeasure dropout loss. $\alpha + \beta + \gamma = 1$.

The loss function of the model contains three optimization objectives. In FIGURE 5, we further describe the process of the model.

IV. MODEL TRAINING

In this section, we introduce the training process of anti-convolution neural network based on multi feature attention mechanism. Because the model needs to fight against the text, the training process includes two parts. First, the original text extracted from the corpus is trained by genetic algorithm to obtain the adversarial text. Then, the original text and the adversarial text are input into the convolutional neural network to train the deep learning model.

A. PARAMETER SETTING

In the genetic algorithm, the setting of parameters is very important to the number of iterations and the quality of the population. In our model, the initial population number is set to 100, the maximum population number is 200, the probability of cross between individuals in the population is 0.4, and the probability of variation in the population is 0.01. Iteration when the adversarial text meets the stop condition, the maximum number of iterations is not set. In the neural network model, the amount of data input to the neural network each time is set to 64. The initial parameters are generated by normal random initialization. The model parameters are saved every 100 times training.

V. EXPERIMENT AND RESULT ANALYSIS

In this section, in order to evaluate the performance of our multi feature attention based anti-convolution neural network model. We have carried out the following experiments. Firstly, the paper analyzes the antagonistic text generated by genetic algorithm, and shows the change rate of the text and the replacement words of the antagonistic text. Then, we input the original text and the training countermeasure text into the model to train the model and evaluate the model from the accuracy, F1 value and recall rate. The security performance of the model is improved by training the adversary training text and the original text at the same time. After that, we use specific examples to analyze the performance of the model more accurately and propose the improvement direction in the future work. The structure of the experiment section is as follows: Experiment 1 describes the generation algorithm of adversarial samples. In Experiment 2, the semantic evaluation accuracy of the proposed model and related models is verified. In Experiment 3 and Experiment 4, we tested the security of the model by using the sentence pairs in TABLE 2 and the adversarial samples generated from MSRP dataset.

TABLE 1. The selected dataset.

	Year	Gener	total pairs
belief.txt	2015	forums	375
headlines.txt	2015	forums	541
images.txt	2015	captions	487
MSRP train	-	web news	5081
MSRP test	-	web news	1726

A. DATASET

In the experiment, we selected STS (Semantic Textual Similarity) dataset and MSRP (Microsoft Research Paraphrase Corpus) dataset respectively. The corpus of STS includes picture title, news title and question answering corpus. In TABLE 1, we give the number of sentence pairs in the dataset and the source of the data. Among them, ''belief.txt'' represents that the content in the document comes from web article. ''headlines.txt'' represents that the content in the document comes from the news title. ''images.txt'' represents that the content in the document comes from the image title.

B. EXPERIMENTAL SETTING

We use Python 3.7 to implement the algorithm mentioned in this paper, and use tensorflow deep learning framework to build adversarial convolution neural network. The word vector dimension used is mapped to 50 dimensions. We do all experiments on the personal computer with 4 gigabytes memory and Intel i5 quad core CPU. In the experiment, $threshod1 = 0.02$ of Algorithm 1 is used to filter out the words with large semantic difference in the synonym set. $Threshod2$ in Algorithm 2 is set to 0.6. For $threshod3$, we adopt a more flexible way to control the replacement rate of words in sentences. Due to the different sentence length, we set the replacement rate to $1/length\_Sentence$, where $length\_Sentence$ stands for the length of the sentence. Based on the above parameters, we carried out experiments on the hardware equipment.

C. EXPERIMENTAL 1: ADVERSARIAL TEXT GENERATION

In this paper, we propose a method based on confrontation training to enhance the robustness of the model.

TABLE 2. Adversarial sentence pairs.

No.	Sentence Pairs
1	"A person is throwing a cat on to the ceiling. A person throws a cat on the ceiling."
1*	"A people is throwing a **Caterpillar** on to the ceiling. A **people** throws a cat on the ceiling. "
2	"The man hit the other man with a stick. The man spanked the other man with a stick."
2*	"The man **hit** the other man with a stick. The man **paddle** the other man with a stick."
3	"NATO traces path out of Afghanistan. NATO Leaders Commit to Afghanistan Transition."
3*	"NATO **draw** path out of Afghanistan. NATO Leaders Commit to **Islamic State of** Afghanistan Transition."
4	"Indian finance minister quits to run for president. Julian Assange Plans to Run for Australian Senate."
4*	"Indian finance minister **throw** to run for president. Julian Assange **architectural** plan to Run for Australian Senate."
5	"Stocks rise early after Greek deal, then flatten. Stocks edge higher after Greek debt deal."
5*	"Stocks rise early after Greek **handle**, then flatten. **Ancestry** edge taller after Greek debt deal."
6	"Two green and white trains sitting on the tracks. Two green and white trains on tracks."
6*	"two green and white trains sitting **is** the tracks. Two green and white **groom** on tracks."
7	"An Apple computer sitting on the floor. A Macintosh computer sitting on the floor."
7*	"an apple computer sitting on the **coldcock**. a macintosh computer **posture** on the floor."
8	"the gop is fighting for freedom of choice. so now it is a fight for freedom of choice."
8*	"the gop is fighting for **free** of choice. so now it is a fight for **liberty** of choice."

Confrontation training does not change the structure of deep learning model, it affects the learning experience of the model through adding the form of adversarial examples. In this paper, we use genetic algorithm to generate adversarial examples through genetic, mutation and cross operation of population texts. Compared with the original sample, the sample has the replacement of synonyms in sentences, and describes the degree of variation of the sample by calculating the change rate. It is worth noting that although the synonyms in the sentence pair are replaced, the standard similarity score of the sentence pair does not change. In TABLE 2, we show the original sentence pair and the adversarial sentence pair. Substitute words in sentences are marked in bold. Among them, i represents the original sentence pair, i^* represents the adversarial sentence pairs generated by genetic algorithm. where, $i = 1, 2, \ldots, 8$. From TABLE 2, it can be noted that the main sentence components such as the subject or object in the sentence pair are replaced with synonyms compared with the original text. This kind of subtle change is hard to find in a large amount of text data. However, some sentences are grammatically wrong because synonym set substitution ignores some imitating considerations such as sentence tense and sentence structure. At present, it is not feasible to manually filter counterattack text, so these errors are difficult to find in a large number of text data. This is a huge security risk for deep learning models.

D. EXPERIMENTAL 2: COMPARISON WITH RELATER MODELS

In Experiment 2, we selected five related baseline models to compare with our proposed model. The verification of the model mainly includes two aspects. First of all, we compare the sentence feature extraction method based on multi feature attention with other models to verify the effect of feature extraction. Then, in order to test the security and robustness of the proposed model, we input the adversarial text generated by genetic algorithm into the baseline model to verify the processing ability of the model. In the experiment, we use accuracy, recall rate and F1 value to test the ability of the model to process sentence pairs. Use credibility to measure the security of the model. The formula for the measure is as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{19}$$

$$Precision = \frac{TP}{TP + FP}, \tag{20}$$

$$Recall = \frac{TP}{TP + FN}, \tag{21}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{22}$$

$$ConfidenceRate = \frac{Correct}{SampleSize}, \tag{23}$$

Among them, TP represents the model to give a positive score to the positive samples in the data set. TN representative model gives a negative score to the negative samples in the data set. The FP representative model gives a negative evaluation of the correct results in the dataset. The FN representative model gives a negative evaluation to the positive samples in the data set. When dealing with adversarial examples, $ConfidenceRate$ represents the confidence rate of the model, and $Correct$ represents the number of countermeasures samples correctly handled by the model. $SampleSize$ represents the total number of adversarial examples processed by the model.

In our model, we use the multi feature attention mechanism to extract the features of sentence pairs. At the same time, we consider the syntax features and location information of sentences, and embed the feature information into the matrix. In order to improve the security and robustness of the model, countermeasures training and dropout are used. According to the characteristics of the model, we choose four baseline models for comparison. The baseline models use MRSP data set to test the accuracy and F_1.

- The sentence similarity analysis model based on convolutional neural network proposed by Zhang *et al*. [50] takes into account the syntactic and semantic information of the sentence at the same time. After extracting the long sentence, the model can pay attention to the main components of the sentence.
- The mutual information attention mechanism proposed by the authors of [58] also associates the text before the data is input into the deep learning model. Unlike

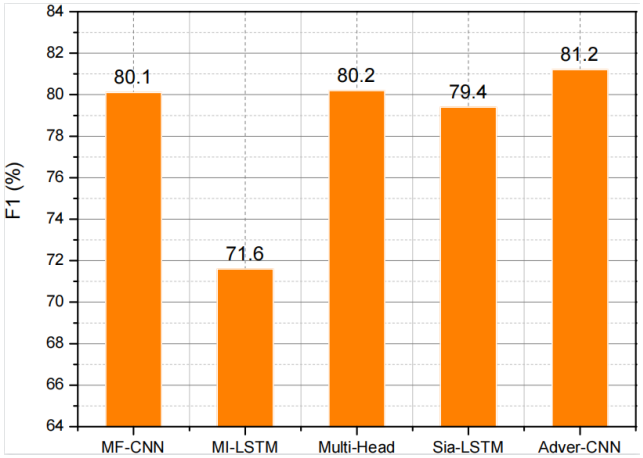

FIGURE 6. Comparison of F1.

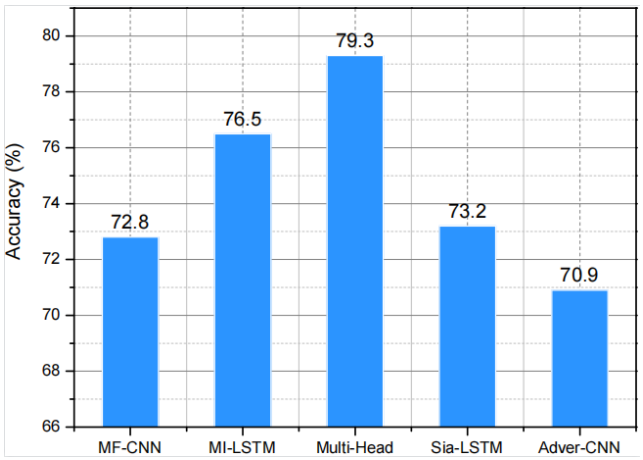

FIGURE 7. Comparison of accuracy.

our work, they use the bidirectional LSTM network to process the text vector. Compared with convolution neural network, short-term memory neural network can better deal with the dependency and context information in long sentences.

- The authors of [54] propose to use twin LSTM network to measure the similarity of semantic text. In the model, there is no prior knowledge or external corpus, only simple word embedding. At the same time, attention mechanism is not used to weight sentence matrix.
- The literature [51] proposed a sentence similarity analysis model based on multi head attention mechanism. Multi head attention is widely used in machine translation. It weights the input content by using deep learning model. Our attention mechanism is to weight the data before it is input into the deep learning model. Based on the deep learning model to realize the sentence weighting, the model can learn the weighted weight according to the sentence characteristics, which is more adaptive to the data set.

In FIGURE 6 and FIGURE 7, we compare the F1 value and accuracy of the baseline model respectively with that of the convolution neural network model. It has the modest accuracy compared with the baseline model. The accuracy of the model decreases with the addition of counter samples. The convolution neural network based on multi feature fusion has poor effect in accuracy comparison, and its accuracy is significantly lower than the other four comparison models. The convolutional neural network based on multiple features does not pay attention to the fusion mechanism, and only calculates sentence similarity by analyzing sentence grammatical components. This model leads to the lack of fine-grained calculation of sentences, which ignores the components other than subject predicate and object. In Figure 6, the F1 value of the model is better than that of the comparison model. In the feature extraction of sentences, we consider the features of sentences, including the mutual information between sentence pairs, the position and semantic relationship of synonyms. This makes the model perform better in comprehensive performance evaluation.

Attention mechanism has been added to other models. It is worth noting that attention model based on multi head is more accurate than mi LSTM and Sia-LSTM. Multi head attention mechanism is an algorithm based on deep learning model. It can flexibly train parameters in the model to learn the most suitable weighting method for data text. In addition, in our model, in addition to the full weighting of sentence multi features, we also add confrontation training and confrontation dropout. The addition of antagonistic samples reduces the fitting degree of the model, and makes the model give more reasonable similarity scores to the sentence pairs input to the neural network. As the harmonic mean of precision and recall, F1 reflects the performance of the model comprehensively. Our model F1 is 81.2%.

E. EXPERIMENTAL 3: SIMILARITY CALCULATION OF ADVERSARY SENTENCE

In this section, in order to analyze the performance of model sentence similarity analysis more carefully and compare the security of the model. We input the original sentence pair in TABLE 2 and the sentence pair generated by genetic algorithm into the anti-convolution neural network, and compare the analysis results. In TABLE 2, we replace synonyms. However, the substitution of synonyms is filtered by genetic algorithm. First, select an object to be selected from the list of synonyms. Then, the algorithm in Algorithm 1 is used to evaluate the replaced text. The synonym will be selected only when the replaced text meets the replacement conditions. Otherwise, another synonym will be selected from the list of synonyms to be selected until the substitution conditions are met. From TABLE 3, we can see the score of sentence similarity calculation given by the model. Among them, *sta_score* represents the similarity score of the sentence pair given by the data set, *cal_score* represents the score given after the model calculation, *adv_sen_ID* and *raw_sen_ID* represent the serial number of the adversary sentence pair and the original sentence pair respectively, which corresponds to the serial number in TABLE 3. From TABLE 3, we notice

TABLE 3. Comparison between standard score and calculated score.

raw_sen_id	*sta_score*	cal_score	adv_sen_id	cal_score
1	5.0	4.3	1*	3.9
2	4.2	4.5	2*	3.4
3	3.2	2.7	3*	3.5
4	0.2	1.0	4*	1.1
5	3.6	2.8	5*	2.5
6	4.1	4.0	6*	3.8
7	5.0	4.6	7*	4.1
8	2.4	2.9	8*	2.7

TABLE 4. Accuracy of different counter ratio.

accuracy	*raw_sen*	counter_sen	counter_ratio
66.4927%	1725	0	0%
66.5329%	1725	20	2.29%
66.6106%	1725	60	3.36%
66.6301%	1725	100	5.47%
66.3806%	1725	140	7.50%
66.3239%	1725	180	9.44%
66.4483%	1725	220	11.31%

that the deviation between the calculated score of the model and the standard score of the original sentence pair is between 0.1 and 0.9, which indicates that the multi-feature attention model proposed by us fully extracts the features in the sentence, making the calculation accuracy of sentence similarity close to the standard score. In addition, for the antagonistic text generated by genetic algorithm, the score deviation given by our model is between 0.2 and 1.1 compared with the original text score, and the accuracy of calculation can basically guarantee the security of the model. The model shows a good recognition ability for aggressive adversary text.

F. EXPERIMENTAL 4: SIMILARITY CALCULATION OF ADVERSARY SENTENCE

For testing the ability of model recognition counter text as a whole, we use the test set of MSRP data set to test the whole training model. Different counting ratios are applied to each group of experiments to test the accuracy of the model. There are 1725 sentence pairs in the original data set. The confrontation text generation algorithm is used to generate counter samples, and the generated counter samples are input into the test set to evaluate the performance of the model. The experimental results are shown in TABLE 4. In TABLE 4, *raw_sen* stands for the number of original sentences. *counter_sen* represents the number of counter texts. *counter_ratio* stands for the ratio of counter text to total text. Before adding counter text, the accuracy of the model to the test set is about 66%. With the addition of confrontation text, the accuracy of the model not only does not decline, but also slightly increases, which shows that our model can classify the counter text correctly. This phenomenon is closely related to our counter text generation algorithm. We associate the iterative process of counter text with the pre-trained neural network model to improve the recognition ability and security of the model.

VI. CONCLUSION AND FUTURE WORK

Sentence similarity analysis is a basic task of natural language processing. Most of the traditional sentence similarity analysis models integrate multiple features in a single way, and do not transform features into word vector matrix. In addition, the security of deep learning model is also concerned. In the previous work, only the improvement of analysis accuracy is pursued, and the security of sentence similarity analysis model is completely ignored.

In this paper, we propose an adversarial convolution neural network model based on multi feature attention mechanism. Compared with other work, the method of sentence feature extraction and the security of deep learning model have been improved to some extent. First of all, the multi feature attention model, which is used to weight sentence features, gives a comprehensive consideration to sentence components, including synonym position information, semantic relationship, etc. Secondly, in order to improve the security of the model, we introduce anti-training and anti-convolution neural network into the sentence similarity analysis model. The addition of countermeasure mechanism not only improves the security of the model, but also improves the calculation accuracy of the model.

In the future work, we will improve the structure of the adversarial convolution neural network to further improve the security of the model.

REFERENCES

[1] S. Zhou and B. Tan, "Electrocardiogram soft computing using hybrid deep learning CNN-ELM," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105778.
[2] S. He, Z. Li, Y. Tang, Z. Liao, F. Li, and S.-J. Lim, "Parameters compressing in deep learning," *Comput., Mater. Continua*, vol. 62, no. 1, pp. 321–336, 2020.
[3] G. S. Aujla, A. Jindal, R. Chaudhary, N. Kumar, S. Vashist, N. Sharma, and M. S. Obaidat, "DLRS: Deep learning-based recommender system for smart healthcare ecosystem," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[4] D. Goularas and S. Kamis, "Evaluation of deep learning techniques in sentiment analysis from Twitter data," in *Proc. Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. (Deep-ML)*, Aug. 2019, pp. 12–17.
[5] S. Roy, W. Menapace, S. Oei, B. Luijten, E. Fini, C. Saltori, I. Huijben, N. Chennakeshava, F. Mento, A. Sentelli, and E. Peschiera, "Deep learning for classification and localization of COVID-19 markers in point-of-care lung ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 8, pp. 2676–2687, Aug. 2020.
[6] S. Hassan, A. Irfan, A. Mirza, and I. Siddiqi, "Cursive handwritten text recognition using bi-directional LSTMs: A case study on Urdu handwriting," in *Proc. Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. (Deep-ML)*, Aug. 2019, pp. 67–72.

[7] M. Singh, R. Kumar, and I. Chana, "Neural-based machine translation system outperforming statistical phrase-based machine translation for low-resource languages," in *Proc. 12th Int. Conf. Contemp. Comput. (IC)*, Aug. 2019, pp. 1–7.
[8] H. Bing, "Statistical machine translation algorithm based on improved neural network," in *Proc. Int. Conf. Robots Intell. Syst. (ICRIS)*, Oct. 2017, pp. 294–297.
[9] S. Nejati, "Testing cyber-physical systems via evolutionary algorithms and machine learning," in *Proc. IEEE/ACM 12th Int. Workshop Search-Based Softw. Test. (SBST)*, May 2019.
[10] J. Fan, J. Jiao, W. Wu, and T. Zhao, "A model-checking oriented modeling method for safety critical system," in *Proc. 1st Int. Conf. Rel. Syst. Eng. (ICRSE)*, Oct. 2015, pp. 1–6.
[11] C. Kening, Z. Boming, W. Wenchuan, and S. Hongbin, "An intelligent checking system for power system operation tickets," in *Proc. 4th Int. Conf. Electr. Utility Deregulation Restructuring Power Technol. (DRPT)*, Jul. 2011, pp. 757–762.
[12] L. J. Wu, "The development of the intelligent checking system for nuclear power plant 3D model," in *Proc. Int. Conf. Power Syst. Technol. (POWERCON)*, Nov. 2018, pp. 4668–4673.
[13] H. Ruan, Y. Li, Q. Wang, and Y. Liu, "A research on sentence similarity for question answering system based on multi-feature fusion," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Oct. 2016, pp. 507–510.
[14] M.-M. Shao and D.-M. Qian, "The application of levenshtein algorithm in the examination of the question bank similarity," in *Proc. Int. Conf. Robots Intell. Syst. (ICRIS)*, Aug. 2016, pp. 422–424.
[15] C. An, J. Huang, S. Chang, and Z. Huang, "Question similarity modeling with bidirectional long short-term memory neural network," in *Proc. IEEE 1st Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2016, pp. 318–322.
[16] D. Zhang, S. Jie, H. Gang, and L. Gao, "Sharp and real image super-resolution using generative adversarial network," in *Proc. Int. Conf. Neural Inf. Process.*, 2017, pp. 217–226.
[17] C. Ledig, L. Theis, F. Huszár, J. Caballero, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
[18] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017.
[19] R. Gupta, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and B. Sadoun, "HaBiTs: Blockchain-based telesurgery framework for healthcare 4.0," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Aug. 2019, pp. 193–201.
[20] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 312–326.
[21] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *Proc. IEEE Secur. Privacy Workshops*, May 2018, pp. 50–56.
[22] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 4957–4968, Sep. 2019.
[23] D. He, M. Ma, S. Zeadally, N. Kumar, and K. Liang, "Certificateless public key authenticated encryption with keyword search for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3618–3627, Aug. 2018.
[24] M. Long, F. Peng, and Y. Zhu, "Identifying natural images and computer generated graphics based on binary similarity measures of PRNU," *Multimedia Tools Appl.*, vol. 78, no. 1, pp. 489–506, Jan. 2017.
[25] H. Li, W. Li, H. Wang, and J. Wang, "An optimization of virtual machine selection and placement by using memory content similarity for server consolidation in cloud," *Future Gener. Comput. Syst.*, vol. 84, pp. 98–107, Jul. 2018.
[26] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.
[27] K. Gu, L. Wang, and B. Yin, "Social community detection and message propagation scheme based on personal willingness in social network," *Soft Comput.*, vol. 23, no. 15, pp. 6267–6285, 2018.
[28] L.-B. Zhang, F. Peng, L. Qin, and M. Long, "Face spoofing detection based on color texture Markov feature and support vector machine recursive feature elimination," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 56–69, Feb. 2018.
[29] R. Sun, L. Shi, C. Yin, and J. Wang, "An improved method in deep packet inspection based on regular expression," *J. Supercomput.*, vol.75, no. 6, pp. 3317–3333, 2019.
[30] F. Peng, D.-I. Zhou, M. Long, and X.-M. Sun, "Discrimination of natural images and computer generated graphics based on multi-fractal and regression analysis," *Int. J. Electron. Commun.*, vol. 71, no. 1, pp. 72–81, 2017.
[31] W. Hao, L. Xiang, Y. Li, P. Yang, and X. Shen, "Reversible natural language watermarking using synonym substitution and arithmetic coding," *Comput. Mater. Contin.*, vol. 55, pp. 541–559, Jan. 2018.
[32] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473
[33] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *Proc. Annu. Conf. Artif. Intell.*, 2002, pp. 18–32.
[34] C. Callison-Burch, P. Koehn, C. Monz, and J. Schroeder, "Findings of the 2009 workshop on statistical machine translation," in *Proc. 4th Workshop Stat. Mach. Transl. (StatMT)*, 2009, pp. 12–58.
[35] Y. Liu, Y. Xiao, Q. Zhang, and Z. Liang, "Fixed telephone intelligent voice service system on telecom platform," *Adv. Mater. Res.*, vols. 765–767, pp. 2710–2714, Sep. 2013.
[36] K. Zhang, J. He, J. Yang, X. Zhang, and Z. Wang, "The application of intelligent voice analysis technology in the customer call center of telecom operators," *Multi Media Tools Appl.*, 2015.
[37] L. R. Rabiner, "Applications of speech recognition in the area of telecommunications," in *Proc. Workshop Autom. Speech Recognit. Understanding*, 1997, pp. 501–510.
[38] A. J. Dugan and D. E. Mcdysan, "ATM virtual private networks," *Commun. ACM*, vol. 38, no. 2, pp. 101–109, 2000.
[39] J. Zakos and L. Capper, "Clive an artificially intelligent chat robot for conversational language practice," in *Proc. Hellenic Conf. Artif. Intell.*, 2008, pp. 437–442.
[40] S. Zhang, J. Gu, and Y. Yan, "Research on personalized recommendation system based on intelligent chat robot," IEEE Beijing Sect., Chongqing, China, Tech. Rep. 02, 2011.
[41] S. Huang and Y. Zhang, "Sentence similarity calculation method based on multiple-features," *J. Beijing Inf. Sci. Technol. Univ.*, vol. 32, no. 5, pp. 45–49, 2017.
[42] S. Sangeetha and M. Arock, "Recognising sentence similarity using similitude and dissimilarity features," *Int. J. Adv. Intell. Paradigms*, vol. 4, no. 2, p. 120, 2012.
[43] E. Shareghi and S. Bergler, "Feature combination for sentence similarity," in *Proc. Can. Conf. Artif. Intell.*, 2013, pp. 150–161.
[44] A. John, P. S. Premjith, and M. Wilscy, "Extractive multi-document summarization using population-based multicriteria optimization," *Expert Syst. Appl.*, vol. 86, pp. 385–397, Nov. 2017.
[45] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Trans. Knowl. Discovery Data*, vol. 2, no. 2, pp. 1–25, 2008.
[46] Y. Li, D. McLean, Z. Bandar, J. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.
[47] R. M. Aliguliyev, "A new sentence similarity measure and sentence based extractive technique for automatic text summarization," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7764–7772, May 2009.
[48] S. Dan, R. Banjade, and V. Rus, "A sentence similarity method based on chunking and information content," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*, 2014, pp. 442–453.
[49] H. T. Nguyen, P. H. Duong, and T. Q. Le, *A Multifaceted Approach to Sentence Similarity*. Cham, Switzerland: Springer, 2015.
[50] P. Zhang, X. Huang, and M. Li, "Disease prediction and early intervention system based on symptom similarity analysis," *IEEE Access*, vol. 7, pp. 176484–176494, 2019.
[51] W. Bao, W. Bao, J. Du, Y. Yang, and X. Zhao, "Attentive Siamese LSTM network for semantic textual similarity measure," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Nov. 2018, pp. 312–317.
[52] H. Yao, H. Liu, and P. Zhang, "A novel sentence similarity model with word embedding based on convolutional neural network," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 23, p. e4415, Dec. 2018.
[53] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Inf. Sci.*, vol. 373, pp. 388–403, Dec. 2016.
[54] M. Y. Wang, C. L. Li, J. D. Sun, W. R. Xu, S. Gao, Y. H. Zhang, P. Wang, and J. L. Li, "Learning sentences similarity by multi-head attention," in *Proc. Int. Conf. Netw. Infrastruct. Digit. Content (IC-NIDC)*, Aug. 2018, pp. 16–19.

[55] M. Alzantot, Y. Sharma, A. Elgohary, B. J. Ho, and K. W. Chang, "Generating natural language adversarial examples," 2018, *arXiv:1804.07998*. [Online]. Available: https://arxiv.org/abs/1804.07998
[56] S. A. Israel, J. H. Goldstein, J. S. Klein, J. Talamonti, F. Tanner, S. Zabel, P. A. Sallee, and L. McCoy, "Generative adversarial networks for classification," in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2017, pp. 1–4.
[57] F. Ali, S. El-Sappagh, and D. Kwak, "Fuzzy ontology and LSTM-based text mining: A transportation network monitoring system for assisting travel," *Sensors*, vol. 19, no. 2, p. 234, Jan. 2019.
[58] N. Jiang, F. Tian, J. Li, X. Yuan, and J. Zheng, "MAN: Mutual attention neural networks model for aspect-level sentiment classification in SIoT," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2901–2913, Apr. 2020.

QIFENG SUN received the M.S. and Ph.D. degrees in computer science and technology from the China University of Petroleum (East China), in 2004 and 2011, respectively. He is currently a Lecturer with the College of Computer Science and Technology, China University of Petroleum (East China). His research interests include natural language understanding and artificial intelligence for networking.

XINGZHE HUANG is currently pursuing the degree with the College of Computer Science and Technology, China University of Petroleum (East China). His research interests include artificial intelligence and natural language processing.

GODFREY KIBALYA received the B.Sc. degree in telecommunications engineering from Makerere University Uganda, in 2010, and the M.Sc. degree in telecommunications engineering from the University of Trento, Italy. He is currently pursuing the Ph.D. degree with the Department of Network Engineering, Technical University of Catalonia (UPC), Spain. His research interests include network function virtualization, and application of artificial intelligence in network management.

NEERAJ KUMAR (Senior Member, IEEE) received the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir, India, in 2009. He was a Postdoctoral Research Fellow with Coventry University, Coventry, U.K. He is currently a Full Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, Punjab, India. He is also a Visiting Research Fellow with Coventry University, Newcastle University, U.K. He has guided many research scholars leading to Ph.D. and M.E./M.Tech. His research is supported by funding from UGC, DST, CSIR, and TCS. He has more than 6200 citations to his credit with current H-index of 42. He has edited more than ten journals special issues of repute and published four books from CRC, Springer, IET U.K., and BPB publications. He has published more than 300 technical research articles in top-cited journals, such as IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Industrial Electronics, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Consumer Electronics, IEEE Transactions on Industrial Informatics, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Smart Grid, IEEE Network, IEEE Transactions on Communications, IEEE Wireless Communications, IEEE Internet of Things Journal, IEEE Systems Journal, *Computer Networks*, *Information Sciences*, *FGCS*, *JNCA*, *JPDC*, and *ComCom*. He has received the Best Paper Award from IEEE Systems Journal and ICC 2018, Kansas city, in 2018. He has been a Workshop Chair at IEEE Globecom 2018 and IEEE ICC 2019 and a TPC Chair and a member for various international conferences. He is an Associate Technical Editor of *IEEE Communication Magazine* and *IEEE Network Magazine*. He is an Associate Editor of *IJCS* (Wiley), *JNCA* (Elsevier), *Computer Communications* (Elsevier), and *Security and Communication* (Wiley). He has been a Guest Editor of various international journals of repute, such as IEEE Access, *IEEE Communication Magazine*, *IEEE Network Magazine*, *Computer Networks* (Elsevier), *Future Generation Computer Systems* (Elsevier), *Journal of Medical Systems* (Springer), *Computer and Electrical Engineering* (Elsevier), *Mobile Information Systems*, *International Journal of Ad Hoc and Ubiquitous Computing*, *Telecommunication Systems* (Springer), and *Journal of Supercomputing* (Springer).

SANTHOSH KUMAR S. V. N. received the B.E. degree in computer science and engineering and the M.E. degree in software engineering from Anna University, Chennai, India, in 2011 and 2013, respectively. He is currently an Assistant Professor with the VIT, Vellore Campus, India. He works in the areas of security and data dissemination in wireless sensor networks. He does research in information systems (business informatics), computer communications (networks), and computer security and reliability. He has published 20 articles in reputed international journals and conferences. His research interests include wireless sensor networks, the Internet of Things, and mobile computing. He is a Peer Reviewer of *Peer-to-Peer Networking and Applications* (Springer), IEEE Transactions on Computational Social Systems, and *Wireless Personal Communications*.

PEIYING ZHANG received the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, in 2019. He is currently an Associate Professor with the College of Computer Science and Technology, China University of Petroleum (East China). Since 2016, he has been publishing multiple IEEE/ACM transactions/journal/magazine papers, such as IEEE Transactions on Industrial Informatics, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Vehicular Technology, IEEE Transactions on Network Science and Engineering, IEEE Transactions on Network and Service Management, IEEE Transactions on Emerging Topics in Computing, IEEE Network, IEEE Access, IEEE Internet of Things Journal, *ACM TALLIP*, *Computer Communications*, and *IEEE Communications Magazine*. His research interests include semantic computing, future internet architecture, network virtualization, and artificial intelligence for networking. He served as the Technical Program Committee of ISCIT 2016, ISCIT 2017, ISCIT 2018, ISCIT 2019, Globecom 2019, COMNETSAT 2020, SoftIoT 2021, IWCMC-Satellite 2019, and IWCMC-Satellite 2020.

DONGLIANG XIE (Member, IEEE) received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2002. He was a Visiting Researcher with the Department of Electrical and Computer Engineering, State University of New York at Stony Brook. He is currently a Full Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China. His research interests include resource-constrained wireless communication and information-centric networks, including architecture of ubiquitous and heterogeneous networks, complex network analysis, as well as content retrieval and service management.