electronics

MDPI

Article

Chinese Spam Detection Using a Hybrid BiGRU-CNN Network with Joint Textual and Phonetic Embedding

Jinliang Yao [1,2,*], Chenrui Wang [1], Chuang Hu [1] and Xiaoxi Huang [1]

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China; wcr@hdu.edu.cn (C.W.); huchuang@hotmail.com (C.H.); huangxx@hdu.edu.cn (X.H.)
[2] Zhejiang Key Laboratory of Brain-Machine Collaborative Intelligence, Hangzhou 310018, China
* Correspondence: yaojinl@hdu.edu.cn

Abstract: The proliferation of spam in China has a negative impact on internet users' experiences online. Existing methods for detecting spam are primarily based on machine learning. However, it has been discovered that these methods are susceptible to adversarial textual spam that has frequently been imperceptibly modified by spammers. Spammers continually modify their strategies to circumvent spam detection systems. Text with Chinese homophonic substitution may be easily understood by users according to its context. Currently, spammers widely use homophonic substitution to break down spam identification systems on the internet. To address these issues, we propose a Bidirectional Gated Recurrent Unit (BiGRU)–Text Convolutional Neural Network (TextCNN) hybrid model with joint embedding for detecting Chinese spam. Our model effectively uses phonetic information and combines the advantages of parameter sharing from TextCNN with long-term memory from BiGRU. The experimental results on real-world datasets show that our model resists homophone noise to some extent and outperforms mainstream deep learning models. We also demonstrate the generality of joint textual and phonetic embedding, which is applicable to other deep learning networks in Chinese spam detection tasks.

Keywords: Chinese spam detection; natural language processing; text classification; deep hybrid neural networks; joint embedding

Citation: Yao, J.; Wang, C.; Hu, C.; Huang, X. Chinese Spam Detection Using a Hybrid BiGRU-CNN Network with Joint Textual and Phonetic Embedding. *Electronics* **2022**, *11*, 2418. https://doi.org/10.3390/electronics11152418

Academic Editor: Esteban Tlelo-Cuautle

Received: 9 July 2022
Accepted: 2 August 2022
Published: 3 August 2022

1. Introduction

The evolution of social media in the internet era has been fueled by technological advancements. User-generated content can be found in various applications and websites, including social networks, bulletin boards, blogs, instant messaging tools, news commentaries, etc. [1]. The negative circumstances of the COVID-19 quarantine have caused a sharp rise in user-generated content on all platforms [2]. However, there are some criminal and malicious data, such as advertisements, scams, phishing, and other forms of spam, within the vast volume of user-generated text content produced daily. Polarizing opinions, reductions in users' online contact time, and a lower quality of the information supplied are some of the significant problems caused by the massive influx of various types of spam [3]. Unlike in email spam, spammers submit sex advertisements as comments on Web 2.0 apps with the comment feature. Spammers use specific terms to entice readers, and the posting material typically includes sexual connotations and contact information, such as a URL, QQ number, WeChat number, etc. In China, the vast majority of spam consists of sexually explicit advertising material, which is extremely dangerous to children, since these sex commercials readily influence them.

To prevent spam comments, several engineers are making efforts to combat spam delivery, and many companies and research groups offer a variety of spam detection systems. Filtering spam is one way to identify spam accounts based on their activity, profiles, and prior postings [4]. The follower and friend relationships between accounts are

used in social networks to identify spam accounts and their communities [5,6]. However, this method has significant disadvantages; spammers can still circumvent filter systems by altering their posting behavior and following real accounts. Another way of detecting spam is to classify posts based on their textual content by using a DNN-based text classification technique [7]. This delivers outcomes that are extremely competitive by learning the hierarchical representations through multiple processing layers [8,9]. Our objective is to identify particular content spam. Therefore, we are mainly concerned with a content-based method. Unfortunately, recent research has shown that DNN-based text classification techniques are intrinsically sensitive to adversarial examples that are intentionally created by inserting minor perturbations into innocuous input, which can cause the target DNNs to misbehave [10–13]. Spammers frequently change their messages by using variants to avoid detection by filtering systems, and the camouflaged spam content is usually well designed and verified. Spam detection in China's social media is more difficult due to the unique linguistic features of Chinese characters. Unlike English, Chinese uses more than 10,000 characters, which contain a large number of homophones. According to the work of Perfetti and Tan [14], on average, about 11 Chinese characters share the same pronunciation. For instance, spammers may camouflage some harmful words in their comments by using the relevant phonetic variants, such as substituting "微信 (WeChat)" with "威信" and substituting "裸贷 (naked loan)" with "倮贷", as shown in Figure 1.

Figure 1. Camouflaged spam texts. The original characters are in brackets, whilst the variants are in red.

Therefore, this paper proposes a novel solution for the detection of camouflaged Chinese spam, a hybrid BiGRU-CNN network with joint embedding (BiGRU-CNN-JE), which combines the Bidirectional Gated Recurrent Unit model [15] with the Text Convolutional Neural Network model [16]. BiGRU-CNN-JE is capable of learning knowledge of the variation of Chinese characters in terms of phonemes and integrating this knowledge with semantics and contextualized information. The first step is to eliminate the noise that can be easily distinguished from normal characters, and a clever operation is that some often-used noise symbols are divided into various groups. Second, the original Chinese character sequence is converted into a phonetic representation sequence. The two sequences are then turned into joint textual and phonetic embedding before being fed as inputs into the network. Through extensive experiments, we show that BiGRU-CNN-JE outperforms mainstream deep learning models, such as TextCNN, TextRNN [17], TextBiLSTM+Attention [18], TextRCNN [19], DPCNN [20], and Transformer [21]. In addition, we also verify the model robustness of BiGRU-CNN-JE when processing Chinese spam with the homophonic character noise. Finally, we show that the joint textual and phonetic embedding is generalizable and applicable to other deep learning networks.

2. Related Work

2.1. Traditional Detection Methods

The use of a blacklist is one of the most popular non-machine learning strategies. A keyword-blacklist-based spam filter detects whether the text contains keywords from a pre-built keyword library [22]. Although the keyword-blacklist-based method is effective and straightforward, a brief letter containing a few keywords may not constitute spam, and spammers can easily attack that term by adding noise. Spelling correction may serve

as an option to address spam attack problems—for instance, to reduce adversarial editorial attacks, to use a spelling-correction-based framework for preventing adversarial texts [23], or to use the context-aware spelling correction approach [24]. However, due to the sparsity and dynamic nature of training data, this method has limited effectiveness with Chinese texts and frequently fails.

2.2. Detection Methods Based on Machine Learning

The traditional machine learning method was used in spam detection after the progress of classification based on a bag of words. Specifically, Oak [25] employed a Support Vector Machine (SVM) to identify suicidal text content after extracting several features from suicidal text on the internet. Chan et al. [26] investigated a feature re-weighting approach with a novel re-scaling function to improve a linear classifier's resistance against good word assaulting spam. Almeida et al. [27] proffered a new, real, public, and non-encoded SMS spam collection. They demonstrated that SVM performed better than other assessed classifiers, such as Basic Naive Bayes, Flexible Bayes, Minimum Description Length, KNN, etc. However, traditional machine learning techniques typically necessitate extensive feature engineering. Deep exploratory data analysis on the dataset, followed by a simple dimension-reduction procedure, necessitates a significant amount of upfront labor, and this is difficult to reuse. Moreover, the model's fitting capacity is limited by the limited number of parameters in machine learning techniques.

2.3. Detection Methods Based on Deep Learning

Unlike traditional machine learning methods, deep learning methods can automatically learn latent representations of input data to perform classification. The existing deep learning algorithms for spam detection have demonstrated promising results in spam identification. Hassanpour et al. [28] used the word2vec algorithm to represent emails instead of the common keyword-based or other rule-based methods. They tested their method on an open dataset and discovered that deep learning classification methods outperformed traditional machine learning algorithms. Srinivasan et al. [29] suggested the use of natural language processing text representation and mapping to detect spam emails. Compared to typical machine learning classifiers, the deep learning architectures performed better in the experiments. Barushka and Hajek [30] presented an integrated distribution-based balancing approach and regularized deep neural networks for spam filtering. Tong et al. [31] proposed a capsule network model combining long–short attention for detecting Chinese spam, which used a multi-channel structure based on the long–short attention mechanism. According to the experimental results, the model outperformed the current mainstream methods, such as TextCNN, LSTM, and even BERT, in characterization and detection.

Although deep learning methods have achieved notable performance in many classification tasks, most models are not intended to detect camouflaged spam. To put it another way, processing camouflaged spam with noise that is not in the training dataset is tough. The task-oriented model should be able to identify the most critical spam text variations. In addition, deep learning techniques are widely employed in English spam detection, but their usage in detecting Chinese spam, particularly social media spam, is still in its early stages. Liu et al. [32] proposed a joint embedding of both textual and phonetic information of source sentences, which not only significantly improved the robustness of neural machine translation to homophone noise, but also improved the translation quality on some clean test sets. Due to the highly prevalent phenomenon of homophony in Chinese, phonetic information cannot be disregarded during the word-embedding training process. In our work, we investigate two types of joint textual and phonetic embedding that are expected to capture richer representations of Chinese text when used in Chinese spam detection tasks.

3. Methods

As depicted in Figure 2, there are two core factors in our proposed method: The first is data preprocessing in order to reduce the impact of symbol noise and to get two kinds of sequences; the BiGRU-CNN-JE model is then used to show the details of the model for detecting Chinese spam. We will go through them in depth in the rest of this section.

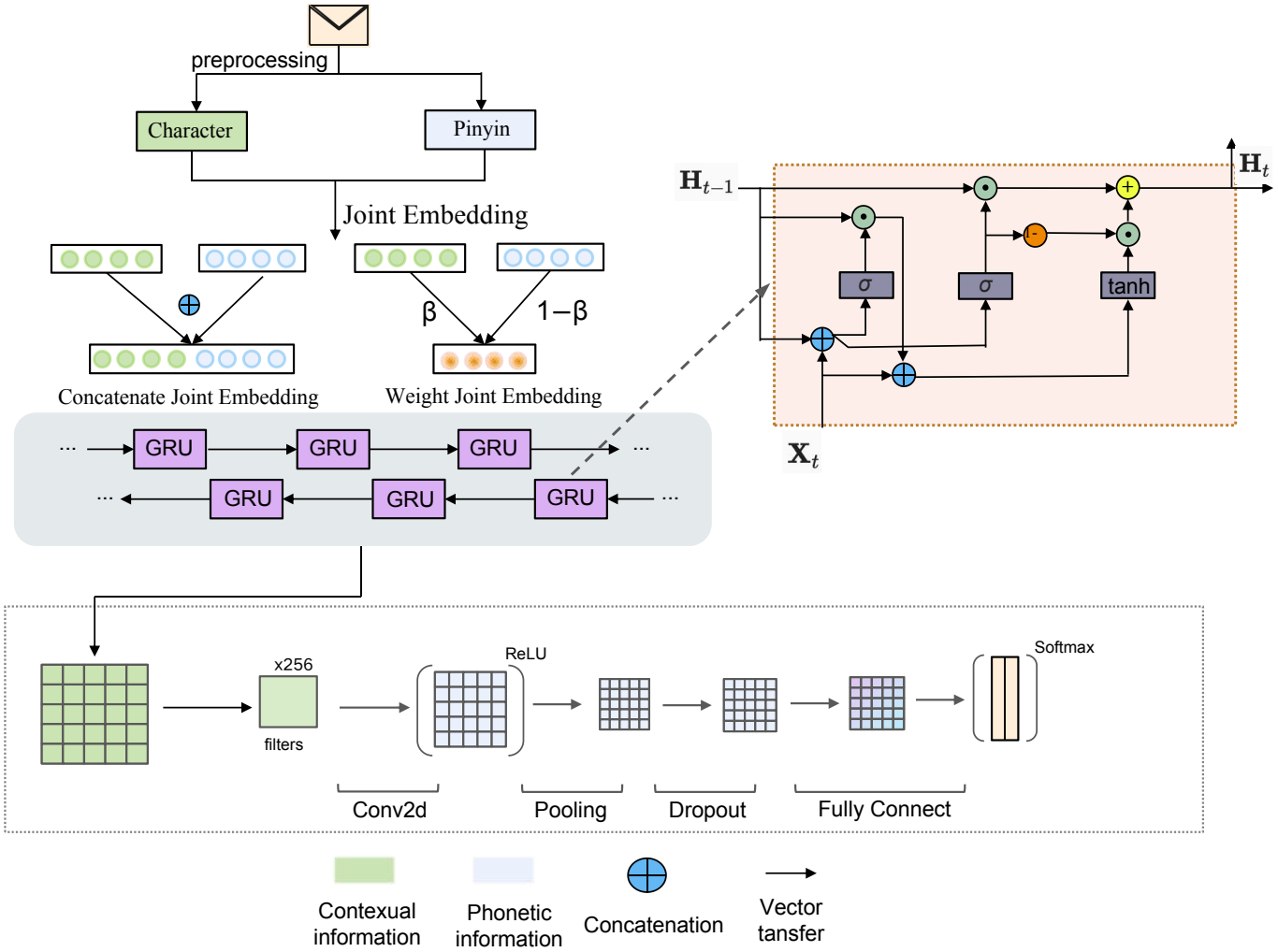

Figure 2. An illustration of the improved BiGRU-CNN with the joint embedding method.

3.1. Data Preprocessing

Data preprocessing of natural language processing (NLP) is a method for cleaning a text in order to make it ready to feed to models. Our data preprocessing process includes noise removal, normalization, and joint embedding preparation, as shown in Figure 3.

The first step is noise removal and normalization, which is a process of data cleaning. To improve the data quality, we apply the six common rule-based text-cleaning strategies listed below to the raw data:

- Normalizing English characters: converting all differently encoded alphabet symbols with similar shapes to standard lowercase English characters;
- Normalizing numeric characters: converting all coded symbols with numeric meaning in Unicode encoding to standard numeric character encoding;
- Transforming special-meaning symbols: converting some symbols with special meanings into corresponding special-meaning symbols. For instance, '†' and other symbols similar to '+' are converted into the Chinese character '加', which means "add";
- Converting traditional Chinese to simplified Chinese characters;

- Removing noise symbols to filter out all symbols of non-Chinese characters, non-English characters, and non-numeric characters;
- Representing consecutive numeric characters: constructing a new consecutive numeric character form in the text that has undergone the previous processing. In this step, a sequence of numeric characters is expressed as "<num_n>" according to the number of numeric characters, where n represents the number of consecutive numeric characters. For instance, "QQ2517645947" is expressed as "QQ<num_10>".

In NLP tasks, data cleaning is a crucial step. A dataset is frequently a collection of words that the computer does not understand without the cleaning process. Normalizing English characters, Chinese characters, numeric characters, and special meaning symbols helps to maintain a consistent flow during the NLP tasks. Noisy symbols that do not provide value or meaning to the NLP model are removed because the symbols in the sentence add up to the noise, which causes ambiguity when training the model. In addition, removing noise symbols from the training data will also reduce the size. Representing consecutive numeric characters eliminates the diversity of number values and the sparsity of training data, as the meaning of numbers in the data is limited. In general, cleaning the data to obtain high-quality data makes it possible for the model to more accurately fit the actual semantic features, strengthens the model's capacity for generalization, and increases the precision of information extraction.

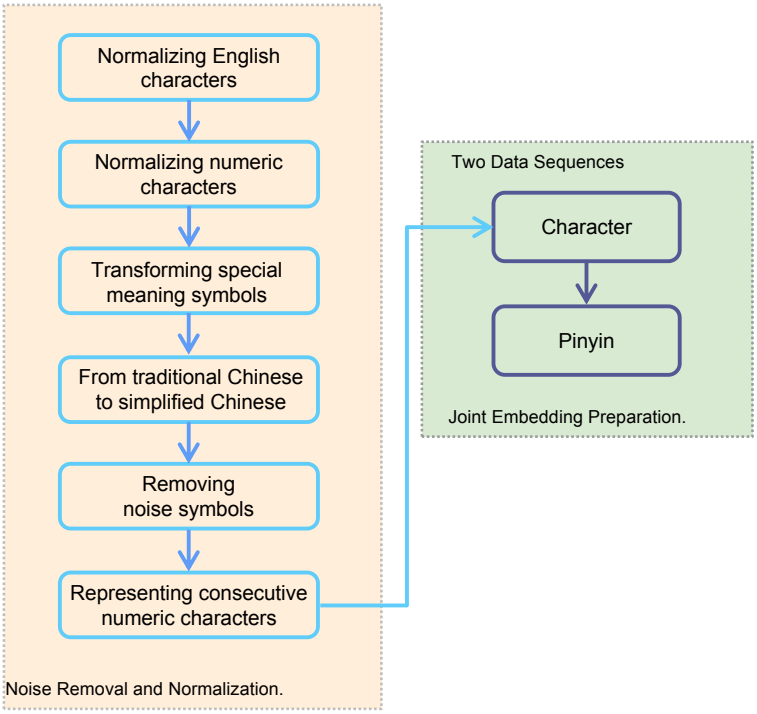

Figure 3. The data preprocessing procedure.

Next, since the phonetic information of Chinese spam in the real world is frequently modified, we employ the Pinyin encoding method to accurately represent the phonetic information for the joint embedding preparation. Pinyin is the romanization of Chinese characters based on their pronunciation. Pinyin can also be used as a written language to express meaning, and distinct characters' Pinyin codes may be the same. For simplicity reasons, the tonal information in Pinyin is discarded in our method. For instance, the Pinyin code of "威信 (prestige)" is "wei xin" and "微信 (WeChat)" is also "wei xin". Therefore, to achieve semantic embedding and phonetic embedding later, we must convert the original sequence into Pinyin sequences based on the Pinyin encoding method.

3.2. BiGRU-CNN-JE Model

The proposed BiGRU-CNN-JE neural network framework consists of the JE module, BiGRU module, and CNN module. To efficiently detect spam with Chinese variants, our model can leverage phonetic information, and it combines the advantages of TextCNN's parameter sharing with the benefits of BiGRU's long-term memory. We first create joint word embeddings, which we then feed into the BiGRU-CNN model for spam classification tasks.

3.2.1. Joint Embedding Module

The primary function of this module is to represent two sequences as a word-embedding matrix. We receive two types of sequences after preprocessing the data. First, we create two dictionaries based on the sequences and then convert the words in the text into dictionary indexes. Finally, we use the dictionary to initialize the word vector matrix. We choose to train directly in our model rather than using supplementary word vector techniques because our training dataset is large enough. To improve the integration of Pinyin information into our word embedding, in this paper, we experiment with two different joint strategies, i.e., concatenate joint embedding and weight joint embedding.

- Concatenate Joint Embedding (CJE):
  CJE denotes the direct concatenation of Chinese character embedding and Pinyin embedding, and the dimensions of the joint embedding are their sum. Formally, we denote by E_T the Chinese character embedding vector and by E_P the Pinyin embedding vector; the joint embedding vector E is obtained by

$$E = E_T \oplus E_P \tag{1}$$

  Obviously, this is the most straightforward to use and requires the fewest model parameters.
- Weight Joint Embedding (WJE):
  WJE refers to weighing the contextual and phonetic information and then combining the two embedding vectors. The joint embedding vector E is as follows:

$$E = \beta E_T + (1 - \beta) E_P \tag{2}$$

  where β is a trainable parameter. WJE is not a coarse-grained joint scheme, and it is capable of dynamically capturing more complex correlations across contextual and phonetic information.

3.2.2. BiGRU Module

The primary function of this module is to learn the contextual semantic features of the input word-embedding matrix. GRU is a gating-based recurrent neural network variation that is often used to handle sequence information. Using a gating mechanism, GRU can make a recurrent neural network recall past information, selectively forget unimportant details, and model the relationship of the long-term context, which helps detect Chinese spam.

BiGRU is a sequence-processing model that consists of two GRUs. One takes the input in a forward direction, and the other takes it in a backward direction, which can capture front and back information characteristics. Formally, given an input X_t at time t, the hidden states acquired by the forward GRU and the backward GRU are $\overleftarrow{h_t}$ and $\overrightarrow{h_t}$, respectively.

$$\overleftarrow{h_t} = \overleftarrow{GRU}(X_t, \overleftarrow{h_t}) \tag{3}$$

$$\overrightarrow{h_t} = \overrightarrow{GRU}(X_t, \overrightarrow{h_t}) \tag{4}$$

At a time t , BiGRU's hidden state is $h_t = [\overleftarrow{h_t}, \overrightarrow{h_t}]$, which is the concatenation of $\overleftarrow{h_t}$ and $\overrightarrow{h_t}$.

3.2.3. CNN Module

The primary function of this module is to extract the most crucial local features of the BiGRU's output. A d-dimensional word vector represents a Chinese character in a sentence in the standard TextCNN model [33]. Let $x_{1:n} = x_1, \ldots, x_n$ be a sequence of words. A convolution operation creates a new feature by applying a filter to a window of words (k-dimensional word vectors). Each window in the sequence receives the filter, which is applied as a dot-product between the embedding vectors in that window and the weight vector, which is frequently followed by a non-linear activation function φ.

Given a window of words $x_i, ..., x_{i+k}$, the concatenated vector of the i-th window is as follows:

$$X_i = x_i \oplus x_{i+1} \oplus ... \oplus x_{i+k} \tag{5}$$

where $\oplus$ is the concatenation operator. Formally, a new feature C_i generated by a filter of width k from a window of words as a result of the convolution operation is:

$$C_i = \varphi(wX_i + b) \tag{6}$$

where w is a filter weight and b is a bias term. Since multiple filters can be used in the model to extract multiple features, we utilize groups of filter sizes of various sizes to allow the model to extract data at various levels of granularity.

Generally, the dimension after a convolution operation needs to be reduced through pooling to ensure that subsequent classification layers can process the output shape of the model. The two most popular pooling operations are mean pooling and max pooling, which, respectively, generalize a feature's average presence and its most activated presence. The pooling operation in this module is the dynamic pooling method, the idea of which is to initialize a two-dimensional weight and to weight the max pooling and mean pooling, respectively, to form a dynamic pooling. Because the two-dimensional weight is determined during the training process, the final weight is biased towards the weight that is beneficial to the current task. Dynamic pooling incorporates more information when the two pooling methods are evenly matched, thereby increasing the model's accuracy. It also shows which pooling method is better for the current task based on the weights, preventing errors in the process of artificially setting the pooling mode.

Formally, given the mean pooling matrix P_{mean} and the max pooling P_{max}, the dynamic pooling P_{dym} operation is:

$$P_{dym} = \alpha P_{mean} + (1 - \alpha) P_{max} \tag{7}$$

where α is a trainable parameter.

Finally, the concatenation operation with dropout is followed by a dynamic pooling layer, followed by two fully connected layers in our model architecture.

4. Experiment

4.1. Dataset

Our model was evaluated on a chat spam dataset, which is a real-world dataset (in Chinese). We separated the chat spam dataset into training, validation, and test splits to prevent our model from overfitting and to accurately evaluate our model. The chat spam dataset was from Tian Ge Interactive Holdings Limited, which owns Miaobo, happy88, and other live broadcast platforms. These texts came from chat rooms, and spam content was mostly about pornography, commercial advertisements, and offensive content. The detailed information of the dataset is shown in Table 1. To better measure the model's performance, we did not use other datasets for pre-training during the experiment.

Table 1. Statistics of the Chinese chat spam dataset.

Dataset	Part	All	Spam	Normal
Chat	Training	208,240	104,242	103,998
	Dev	69,413	34,509	34,904
	Test	69,415	34,783	34,632

4.2. Experimental Configuration

We validated our proposed model in the Chinese spam detection task using the chat spam dataset. In addition, cross-entropy was used as the loss function in our neural network model. The closer the two probability distributions are, the lower the cross-entropy. The following is the cross-entropy formula:

$$Loss = -\sum_{i=1}^{n} y_i log y_i' \tag{8}$$

where n is the number of training samples used, and y_i and y_i' denote the actual and predicted values, respectively.

It is worth noting that we used the early stopping strategy to prevent the model from overfitting. Specifically, the training was terminated when the loss of the validation set did not decrease for a period of time. Additionally, to improve the information flow in the network, we performed Xavier initialization [34] of the network weights so that the variance of the output from each layer was as equal as possible.

To ensure the reproducibility of our experiment, we provided the main model hyperparameter configurations, as shown in Table 2. Additionally, we will release the datasets and code via GitHub (https://www.github.com/reatiny/chinesespam, accessed on 8 July 2022). All experiments were run on an NVIDIA TITAN RTX GPU to compare the classification processes.

Table 2. Model hyperparameter information.

Layers/Item	Hyperparameter	Value
Embedding	Embedding_size	300
	Sequence_length	32
BiGRU	Bidirectional	2
	Hidden_size	256
	Layer_number	2
Conv2d	Channel	1
	Filters_size	(2, 3, 4)
	Num_filters	256
Others	Learning_rate	1×10^{-3}
	Batch_size	128
	Drop_out	0.5

4.3. Performance Metrics

To evaluate the performance of different prediction models, the model evaluation metrics in the experiment were accuracy, precision, recall, and F1 score. We specified that TP and TN represent the sample numbers of true-positive and true-negative cases, and FP and FN represent the sample numbers of false-positive and false-negative cases. The calculation parameters are defined as follows:

- Accuracy is the proportion of correctly classified samples to total samples for a given datum;
- Precision is the percentage of samples predicted to be positive as true classes;
- Recall is the percentage of positive samples that are predicted to be positive;
- F1 score is the weighted average of precision and recall.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = \frac{2 * precision * recall}{Precision + Recall} \tag{12}$$

4.4. Experimental Results and Analyses

To evaluate the performance of the BiGRU-CNN-JE model, after data preprocessing, we chose several of the most commonly used neural networks as contrast models, including TextCNN, TextRNN, TextBiLSTM+Attention, TextRCNN, DPCNN, and Transformer. The specific experimental results are shown in Table 3.

Analysis of the experimental results shows that all of the commonly used deep learning models, such as TextCNN, achieved an F1 score below 0.92 in the spam detection task. The main reason is that the commonly used deep learning models only consider the textual information of the spam, but ignore information on the phonetic characteristics. In contrast, the BiGRU-CNN-JE model achieved an accuracy of over 0.94 for the chat spam dataset and outperformed the other models in terms of all evaluation metrics. Since the BiGRU-CNN-JE model uses more components and incorporates Pinyin embedding, it took 3 s on the GPU to classify 69,415 test instances, which was longer than the commonly used deep learning models. The results demonstrate that the BiGRU-CNN-JE model excels at combining phonetic information with semantic information and has more significant advantages in extracting comprehensive features of Chinese spam. However, the BiGRU-CNN-JE model necessitates more classification time and, thus, increases the time cost.

Table 3. Comparison of the Chinese chat spam detection performance of different models.

Model	Precision	Recall	F1 Score	Accuracy	Classification Time
TextCNN	0.9298	0.8996	0.9144	0.9156	1 s
TextRNN	0.9328	0.8981	0.9151	0.9165	1 s
TextBiLSTM+Att	0.9378	0.9009	0.9190	0.9204	1 s
TextRCNN	0.9362	0.9036	0.9196	0.9208	1 s
DPCNN	0.9280	0.9081	0.9179	0.9186	1 s
Transformer	0.9321	0.8534	0.8910	0.8954	2 s
BiGRU-CNN + CJE(ours)	**0.9513**	**0.9382**	**0.9447**	**0.9459**	3 s
BiGRU-CNN + WJE(ours)	**0.9566**	**0.9367**	**0.9465**	**0.9470**	3 s

In addition, we conducted a comparative test by adding varying degrees of homophonic noise to the data in the following tests to better highlight the effectiveness and robustness of our proposed BiGRU-CNN-JE model toward the homophonic noise. In this experiment, homophonic noise was added to the spam data at max rates of 0.1, 0.2, and 0.3. In a specific implementation, each Chinese character (A) in the data was substituted with other homophones (B) with a proportional probability. The substitution homophone B was obtained by the Python module nlpcda. Given a Pinyin code from any Chinese character in the test data, we randomly chose one Chinese character from the return of the nlpcda module. A sample is shown in Figure 4. The original sentence was used when the noise rate was 0, and the sentence would have more homophonic substitutions as the noise rate rose.

Noise Rate	Text(meaning: If you help promote Pandora Swarovski jewelry, you'll get a cut.)
0	兴 趣 帮 推 潘 多 拉 施 华 洛 世 奇 首 饰 你 提 成
0.1	兴 趣 帮 推 潘 多 拉 適 华 洛 世 奇 首 饰 你 提 成
0.2	型 趣 謗 推 潘 多 拉 施 华 洛 世 奇 首 視 你 提 成
0.3	臀 趣 傍 推 潘 多 拉 诗 华 洛 世 奇 首 饰 你 摘 呈

: homophone

Figure 4. A sample with different degrees of noise from homophones. The substituted characters were selected at random from our collection of Chinese homophonic vocabulary.

The performance of the commonly used deep learning models and the BiGRU-CNN-JE model is shown in Table 4 and Figure 5. It was observed that when homophonic noise was added, particularly between 0 and 0.1, the performance of the widely used deep learning models significantly declined. For the Transformer model, the F1 score metric of the model decreased the most in the interval of noise from 0.2 to 0.3. In comparison, the performance degradation of the proposed BiGRU-CNN-JE model was not obvious. The performance of our proposed model was almost consistent in the noise rate range from 0.1 to 0.3. In addition, at different rates of noise, the proposed model had the best performance with different measurement criteria. This provides evidence that the proposed BiGRU-CNN-JE model can resist homophonic noise to some extent. In contrast, the commonly used deep learning models are vulnerable.

Table 4. Comparison of the results with different homophonic noise rates with the commonly used deep learning models and the BiGRU-CNN-JE model.

Model	Noise Rate	Precision	Recall	F1 Score	Accuracy
TextCNN	0	0.9298	0.8996	0.9144	0.9156
	0.1	0.9178	0.8689	0.8927	0.8953
	0.2	0.9089	0.8697	0.8889	0.8910
	0.3	0.9078	0.8466	0.8761	0.8800
TextRNN	0	0.9328	0.8981	0.9151	0.9165
	0.1	0.9421	0.8451	0.8910	0.8963
	0.2	0.9247	0.8624	0.8925	0.8959
	0.3	0.9156	0.8582	0.8859	0.8893
TextBiLSTM+Att	0	0.9378	0.9009	0.9190	0.9204
	0.1	0.9084	0.8885	0.8983	0.8992
	0.2	0.9073	0.8851	0.8961	0.8971
	0.3	0.9164	0.8566	0.8855	0.8890
TextRCNN	0	0.9362	0.9036	0.9196	0.9208
	0.1	0.9223	0.8806	0.9010	0.9030
	0.2	0.9151	0.8723	0.8932	0.8955
	0.3	0.9208	0.8607	0.8897	0.8931
DPCNN	0	0.9280	0.9081	0.9179	0.9186
	0.1	0.9161	0.8828	0.8992	0.9008
	0.2	0.9179	0.8681	0.8923	0.8950
	0.3	0.9218	0.8468	0.8827	0.8872
Transformer	0	0.9321	0.8534	0.8910	0.8954
	0.1	0.9095	0.8357	0.8710	0.8760
	0.2	0.9121	0.8699	0.8696	0.8699
	0.3	0.9158	0.7921	0.8495	0.8593

Table 4. *Cont.*

Model	Noise Rate	Precision	Recall	F1 Score	Accuracy
BiGRU-CNN + CJE(ours)	0	**0.9513**	0.9382	**0.9447**	**0.9459**
	0.1	0.9396	0.9359	0.9378	0.9378
	0.2	0.9353	0.9369	0.9361	0.9359
	0.3	0.9406	0.9332	0.9369	0.9370
BiGRU-CNN + WJE(ours)	0	**0.9566**	0.9367	**0.9465**	**0.9470**
	0.1	0.9405	**0.9451**	**0.9428**	**0.9425**
	0.2	**0.9511**	0.9317	**0.9413**	**0.9417**
	0.3	0.9410	**0.9430**	**0.9420**	**0.9418**

We selected a predetermined threshold value for each of the metrics: The precision threshold value was 0.95, while the threshold values for recall, F1 score, and accuracy were all 0.94.

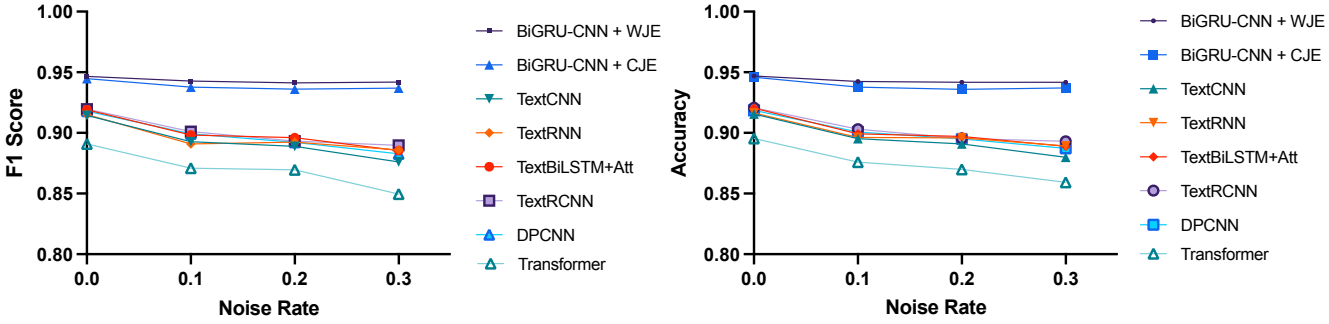

Figure 5. Performance changes in the models with different homophonic noise rates.

Finally, we investigated the generality of the two types of joint embedding with the intention of determining whether they could be applied to other commonly used deep learning networks. The experimental results of using joint embedding as an input for various deep learning models are shown in Figure 6. The analysis results show that the two types of joint embedding both contributed positively to the F1 scores of all the deep learning models. This demonstrates the importance of joint embedding in the corpus of Chinese spam text, as it meets the actual requirements and possesses a good value.

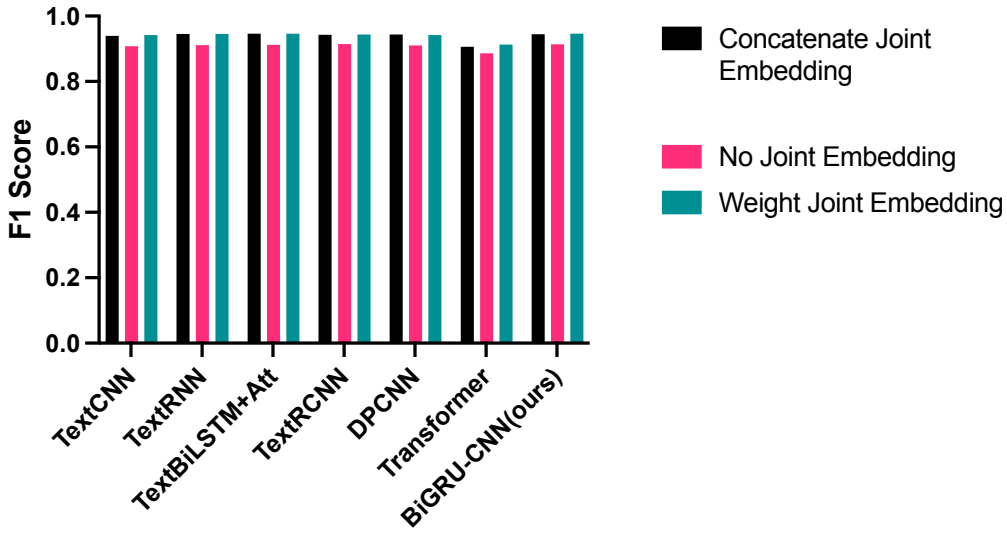

Figure 6. The influence of the joint textual and phonetic embedding on various deep learning models.

4.5. Ablation Experiment

To verify that our model's structural architecture is meaningful, we investigated the impact of a few components on the BiGRU-CNN-JE model's performance. Since the results of the two types of joint embedding are similar, we chose WJE in this experiment for the sake of clarity. Table 5 depicts the experimental outcomes. The investigation's results demonstrate that the joint embedding and BiGRU layer improve the model's performance.

Table 5. Ablation experiment targeting components of the BiGRU-CNN-JE model.

Model	Change	Precision	Recall	F1 Score	Accuracy
BiGRU-CNN-JE		0.9566	0.9367	0.9465	0.9470
Embedding	No Joint Embedding	0.9309	0.9037	0.9171	0.9181
BiGRU Layer	No BiGRU	0.9498	0.9342	0.9419	0.9422

5. Discussion and Limitations

We improved the model's performance on a Chinese chat spam dataset in terms of the precision, recall, accuracy, and F1 score metrics by incorporating joint embedding. The fact that phonetic information, particularly in Chinese, heavily reflects its semantics is the reason for why it can make the model perform better. In addition, the BiGRU-CNN-JE model is more robust in the presence of homophonic noise. The fundamental reason for robustness is that, in the presence of homophonic noise, the word embedding generated by the model will be closer to the word embedding of the original text, rendering the degradation of the model's performance metrics insignificant.

The classification task that we performed is valuable for long-term research because it helps spam censors on social media better filter out harmful information. Meanwhile, in the future, our proposed idea of joint textual and phonetic embedding may be used to detect spam in the case of non-Chinese languages. We believe that the language requires the following conditions: (1) the language text itself does not reflect phonetic information and (2) the language system has a corresponding phonetic representation for each text token in its vocabulary.

However, our research still has limitations. The following are unresolved issues in this study:

1. The proposed method primarily focuses on improving the ability to deal with Chinese homophonic noise. It is incapable of dealing with other types of textual noise. We believe that spammers will improve their technology to create adversarial spam to evade spam detection systems. The problem of managing the new noise-adding method is always a vital issue. Other common types of noise are the substitution of Chinese characters with similar shapes, the addition of good words, etc. We would like to find a new way to detect other noise in Chinese spam text in the future.
2. Since the BiGRU-CNN-JE model increases the total number of model parameters, it may slightly reduce efficiency and increase the overall time complexity. We argue that this would not be a barrier to real-world use in areas where network security needs to be guaranteed because spam is both annoying and harmful. To reduce the model's time complexity, we will consider addressing the following directions in the future. First, we will attempt to reduce model parameters through model compression via knowledge distillation [35,36] or improve the convolution operation via depthwise separable convolutions [37]. Next, we will try to use a simple autoencoder and deep neural network in a combined network structure [38] that can perform fast training and prediction. In addition, in an effort to reduce overall computation time, distributed computing [39] will be used in Chinese spam detection tasks.

6. Conclusions

In this paper, we propose an improved BiGRU-CNN model to eliminate the impact of homophonic noise that spammers add to Chinese spam. To limit the influence of noisy symbol sparsity, we apply a data preprocessing step. We collected a dataset from a few real-time broadcast platforms' chat rooms. The dataset consisted primarily of sex advertisement spam with various levels of noise. We evaluated the BiGRU-CNN-JE model and six deep learning models that are commonly used for Chinese spam detection: TextCNN, TextRNN, TextBiLSTM+Att, TextRCNN, DPCNN, and Transformer. The evaluation results demonstrated that our proposed method achieved great performance and outperformed these commonly used deep learning models. The BiGRU-CNN-JE model provided the highest accuracy (over 0.94) and F1 score (over 0.94). In comparison, the other models performed poorly in terms of the accuracy and F1 score metrics, with the Transformer model providing the lowest average accuracy of less than 0.90. We also verified the model robustness of the BiGRU-CNN-JE model when processing Chinese spam with homophonic noise. In addition, we demonstrated that the joint textual and phonetic embedding is general and can be used in other deep learning networks.

Author Contributions: Conceptualization, J.Y., C.W. and C.H.; methodology, J.Y. and X.H.; software, C.W.; validation, J.Y., C.H. and X.H.; formal analysis, J.Y.; investigation, C.W.; data curation, J.Y. and C.H; writing—original draft preparation, C.W.; writing—review and editing, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Key Research & Development Program of Zhejiang Province (No.2019C03127) and in part by Zhejiang Provincial Basic Public Welfare Research Project under Grant LGG20F020012.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bakshy, E.; Rosenn, I.; Marlow, C.; Adamic, L. The role of social networks in information diffusion. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–20 April 2012; pp. 519–528.
2. Luo, Y.; Xu, X. Comparative study of deep learning models for analyzing online restaurant reviews in the era of the COVID-19 pandemic. *Int. J. Hosp. Manag.* **2021**, *94*, 102849. [CrossRef] [PubMed]
3. Rao, S.; Verma, A.K.; Bhatia, T. A review on social spam detection: Challenges, open issues, and future directions. *Expert Syst. Appl.* **2021**, *186*, 115742. [CrossRef]
4. Nalarubiga, E.; Sindhuja, M. Efficient Classifier for Detecting Spam in Social Networks through Sentiment Analysis. *Asian J. Res. Soc. Sci. Humanit.* **2016**, *6*, 1066. [CrossRef]
5. Bindu, P.; Mishra, R.; Thilagam, P.S. Discovering spammer communities in twitter. *J. Intell. Inf. Syst.* **2018**, *51*, 503–527. [CrossRef]
6. Alom, Z.; Carminati, B.; Ferrari, E. Detecting spam accounts on Twitter. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, 28–31 August 2018 ; IEEE: Piscataway, NJ, USA , 2018; pp. 1191–1198.
7. Karan, M.; Šnajder, J. Cross-domain detection of abusive language online. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, 31 October 2018; pp. 132–137.
8. Yaseen, Q.; AbdulNabi, I. Spam email detection using deep learning techniques. *Procedia Comput. Sci.* **2021**, *184*, 853–858.
9. Deshmukh, R.; Bhalerao, V. Performance comparison for spam detection in social media using deep learning algorithms. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 193–201.
10. Papernot, N.; McDaniel, P.; Swami, A.; Harang, R. Crafting adversarial input sequences for recurrent neural networks. In Proceedings of the MILCOM 2016–2016 IEEE Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 49–54.
11. Gao, J.; Lanchantin, J.; Soffa, M.L.; Qi, Y. Black-box generation of adversarial text sequences to evade deep learning classifiers. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; IEEE: Piscataway, NJ, USA, 2018, pp. 50–56.

12. Behjati, M.; Moosavi-Dezfooli, S.M.; Baghshah, M.S.; Frossard, P. Universal adversarial attacks on text classifiers. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; IEEE: Piscataway, NJ, USA, 2019, pp. 7345–7349.
13. Jáñez-Martino, F.; Alaiz-Rodríguez, R.; González-Castro, V.; Fidalgo, E.; Alegre, E. A review of spam email detection: Analysis of spammer strategies and the dataset shift problem. *Artif. Intell. Rev.* **2022**, 1–29 . [CrossRef]
14. Perfetti, C.A.; Tan, L.H. The time course of graphic, phonological, and semantic activation in Chinese character identification. *J. Exp. Psychol. Learn. Mem. Cogn.* **1998**, *24*, 101. [CrossRef]
15. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
16. Chen, Y. Convolutional Neural Network for Sentence Classification. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2015.
17. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. *arXiv* **2016**, arXiv:1605.05101.
18. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 207–212.
19. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
20. Johnson, R.; Zhang, T. Deep pyramid convolutional neural networks for text categorization. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 562–570.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3058.
22. Qiu, Y.; Xu, Y.; Li, D.; Li, H. A keyword based strategy for spam topic discovery from the Internet. In Proceedings of the 2010 Fourth International Conference on Genetic and Evolutionary Computing, Shenzhen, China, 13–15 December 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 260–263.
23. Zhou, Y.; Jiang, J.Y.; Chang, K.W.; Wang, W. Learning to discriminate perturbations for blocking adversarial attacks in text classification. *arXiv* **2019**, arXiv:1909.03084.
24. Li, J.; Ji, S.; Du, T.; Li, B.; Wang, T. Textbugger: Generating adversarial text against real-world applications. *arXiv* **2018**, arXiv:1812.05271.
25. Oak, R. Poster: Adversarial Examples for Hate Speech Classifiers. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2621–2623.
26. Chan, P.P.; Yang, C.; Yeung, D.S.; Ng, W.W. Spam filtering for short messages in adversarial environment. *Neurocomputing* **2015**, *155*, 167–176. [CrossRef]
27. Almeida, T.A.; Hidalgo, J.M.G.; Yamakami, A. Contributions to the study of SMS spam filtering: New collection and results. In Proceedings of the 11th ACM Symposium on Document Engineering, Mountain View, CA, USA, 19–22 September 2011; pp. 259–262.
28. Hassanpour, R.; Dogdu, E.; Choupani, R.; Goker, O.; Nazli, N. Phishing e-mail detection by using deep learning algorithms. In Proceedings of the ACMSE 2018 Conference, Richmond, KY, USA, 29–31 March 2018; p. 1.
29. Srinivasan, S.; Ravi, V.; Alazab, M.; Ketha, S.; Al-Zoubi, A.; Kotti Padannayil, S. Spam emails detection based on distributed word embedding with deep learning. In *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 161–189.
30. Barushka, A.; Hajek, P. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Appl. Intell.* **2018**, *48*, 3538–3556. [CrossRef]
31. Tong, X.; Wang, J.; Zhang, C.; Wang, R.; Ge, Z.; Liu, W.; Zhao, Z. A content-based chinese spam detection method using a capsule network with long-short attention. *IEEE Sens. J.* **2021**, *21*, 25409–25420. [CrossRef]
32. Liu, H.; Ma, M.; Huang, L.; Xiong, H.; He, Z. Robust neural machine translation with joint textual and phonetic embedding. *arXiv* **2018**, arXiv:1810.06729.
33. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1746–1751.
34. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
35. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
36. Chia, Y.K.; Witteveen, S.; Andrews, M. Transformer to CNN: Label-scarce distillation for efficient text classification. *arXiv* **2019**, arXiv:1909.03508.
37. Kaiser, L.; Gomez, A.N.; Chollet, F. Depthwise separable convolutions for neural machine translation. *arXiv* **2017**, arXiv:1706.03059.

38. Dong, S.; Xia, Y.; Peng, T. Network abnormal traffic detection model based on semi-supervised deep reinforcement learning. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 4197–4212. [CrossRef]
39. Ding, T.; Yang, W.; Wei, F.; Ding, C.; Kang, P.; Bu, W. Chinese keyword extraction model with distributed computing. *Comput. Electr. Eng.* **2022**, *97*, 107639. [CrossRef]