Research on Android Multi-classification Based on Text

Hua Zhang¹, Jiawei Qin¹, Yan Wang¹, Yuan Ma², Li Yao³* and Jun Lei³*

¹ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
² Yuzhong District of Chongqing Municipal Public security Burean Chongqing China
³ The National Computer Network Emergency Response Technical Team/Coordination Center, Beijing, China

*Email: yl,leijun@cert.org.cn

Abstract. In recent years, more and more malicious applications have appeared on mobile application platforms, and they are often disguised as social, communication, and game applications. If we classify applications by category when detecting malware, which can improve the accuracy of malware detection. Classification of applications' categories requires a large number of high-quality samples, but labels of applications' categories are very widely in different app stores, and samples of the same function type cannot be obtained quickly and efficiently. This thesis proposes a method for constructing a multi-classification model by using the text content of application description information, and guiding the classification of application by the category of application description. This method collects the description of an application in different app stores, predicts the category of the description through the classification model, and obtains the application category by voting. The model is based on CNN and RNN, and its F1-score is about 3% higher than the text classification model such as textCNN, LSTM. Its training prediction time and memory consumption are only 6% higher than that of textCNN and LSTM models. We named it CRNN, this thesis constructs a data set that can be used for application classification. The data set is classified using application description to obtain each application description and its category.

1. Introduction

With the increasing popularity of smartphones and the development of technological, smartphones have become the target of hacking attacks [1]. Once the smartphone is attacked by malicious applications, hackers can steal personal information from the phone, control the phone remotely, and charge financial costs to infected users, etc. [2]. These malicious behaviours disguise related malicious applications as benign applications and rely on application functions to attract users to install and perform subsequent behaviours, resulting in loss of users’ property.

There is a strong correlation between application functions and malicious behaviours. If the applications are divided into functions before the detection of malicious applications, the recognition accuracy of malicious applications can be improved [3]. Conventional Android function classifications are divided into static and dynamic detection categories, which are mostly based on features such as intent and application program interface (API). These features are used to represent application functions to build a two-class or multi-class model. However, this traditional Android application classification method spends a lot of time and cost on feature extraction, sandbox construction and other links [4]. Liu et al. [5] proposed to infer the requirements through the descriptions of Apps.
Therefore, many researchers have changed their research directions and started to study the value of application-related information provided in the application store for application functions classification.

In the process of investigating the application stores, we found that there are situations where the label names are not uniform among application stores, application labels are randomly filled, and there are applications with similar functions between different labels in China. According to the characteristics of application stores, this paper proposes to use application description information as the basis for dividing application categories, so as to complete the classification of application functions in application stores. This function classification based on application description has positive significance for application function classification and malicious application detection.

This paper describes this text content by application, and attempts to use natural language processing technology to divide the functions of the application. The main contributions of this paper are as follows:

1) In order to solve the problem of poor description quality in application stores, this paper proposes a categorized method based on keyword sets and JS divergence. This method uses the keyword matching method to obtain the description to be labelled category. This annotation method effectively expands the application description data set.

2) Aiming at the problems of high description similarity and uneven sample distribution among categories in the application description data set, this paper proposes a model that combines the characteristics of CNN and RNN to classify the data set better.

3) In the process of application description classification, a data set with a total of 560,000 description information in 17 categories was constructed to facilitate other researchers to conduct application function classification research based on application description.

2. Prerequisite

2.1. TextCNN

The textCNN model was proposed by Yoon Kim [6]. It applies the filtering operation in image processing to the text classification task, and uses the convolution kernels of different sizes to extract the key information of the sentence, as shown in Figure 1. After the word segmentation result is processed by word vector mapping, the data is converted into a vector matrix representation, and a feature vector is obtained through a convolution operation and then a pooling operation is performed to obtain a text feature representation vector.

![Figure 1. The model framework of textCNN.](image)
This construction method is very similar to the N-Gram model in the language model, which can capture the local similarity better. In the textCNN paper [6], the convolutional layer is designed as a feature map of multiple filters. Due to the high relevance of adjacent words in the sentence, a filter of the size of the convolution kernel can be used to extract key information from the text. By using the maximum-pooling strategy in the pooling layer, the most critical information captured by a filter in different related words can be mined.

Although CNN is easy to extract local key features, but there are some problems, for example: 1) It cannot obtain long sequences of information; 2) Complicated hyperparameter regulation of convolution kernel; 3) If pooling layers selected maximum value pooling strategy, a lot of valuable information will be lost, and the correlation between the part and the whole will be ignored.

In figure 2, the word vectors are input into the loop unit in sequence, and after the gate control process, they are combined into a higher-level sequence feature vector, and the final neural unit is combined with the classification function to achieve classification. When RNN processes context information of data, it is unable to reflect the importance difference of the output sequence information at each moment, which will result in the loss of valid feature information.

![Figure 2. Internal structure of LSTM unit.](image)

2.2. LSTM

Compared with textCNN, RNN can consistently express data context information, long-term memory network (LSTM) [7] as a recurrent neural network variant, is used to solve the dependency relationship before and after words, and ensure the extraction of sequence feature information. Its framework is shown in figure 2.

2.3. RCNN

Because both textCNN and RNN networks have their own advantages, there are previous workers [8] proposed the RCNN model, which retains the position invariance of CNN processing local information, the ability of RNN is to easily sequence information. The RCNN model framework is shown in figure 3.

![Figure 3. Framework of RCNN model.](image)

It can be seen from figure 3 that it expands the semantic information of word vectors through bidirectional LSTM, and then maps word vectors to low dimensions by using non-linear methods. By taking the maximum value of each dimension in the low-dimensional word vector in all time series, the abstract representation of the entire sentence is completed, and finally the Softmax function is used to obtain the classification result of the sentence. It is worth noting that the structure does not use convolution kernels to obtain abstract features, and its processing speed is also slow.

3. Motivation

3.1. The Hidden Form of Malicious Apps

China Internet Network Information Centre (CNNIC) in August 30, 2019 released in Beijing on 44 times, China Internet Development Statistics Report [9] pointed out that the tariff consumption...
category accounted for 74.2% of total number of malicious applications, malicious chargeback class account for 16.5%, and 6.1% of identity theft class. Tariff consumption is more common in "social" and "shopping" applications, malicious deduction applications are more common in "game" applications, and many pornographic deduction applications appear in "video" applications.

![Diagram of RCNN model](image)

**Figure 3.** The framework of the RCNN model.

Malicious apps usually rely on app descriptions to attract users to install and carry out subsequent violations. For example, an application called "Mobile Address Book" describes, "Regain old times, you need to use this address book, chat with old friends, make a phone call!". Through analysis, this application is an interception horse application. The strong correlation between this malicious behaviour and the application description can help the malicious application detector to quickly discriminate the application. The detector can crawl the application description and its files in the application store, firstly, classify the functions of the application through the description, and then conduct malicious detection in units of application functions. This can reduce the detection time of application function classification, and can also help the detector to improve the accuracy of malicious application recognition.

3.2. Review Mechanism of App Stores
App Store for the transfer application review mechanism is different, for example, Google Play requires a $25 registration fee when the developer registers an account, and lists a detailed application review mechanism. Malicious application spreaders have a higher cost of spreading malicious applications in such stores. "Huawei App Store" needs to use sesame credit or ID card for verification when applying for a developer account. This information is directly tied to the developer." Huawei App Store” will also be uploaded application security testing, which would greatly reduce the possibility of the malicious applications storage survival.
Through the investigation of domestic application stores, we found that there are many domestic application stores, and many stores lack a complete management mechanism, which has led to the spread of malicious applications. This phenomenon hinders the construction of domestic network application security systems. 360's *China Mobile Security Ecological Research Report* [10] released in 2018 pointed out that the current domestic application stores, different classes between applications large gap between the number of malicious application distribution set. Therefore, using application descriptions to divide the functions of applications can effectively help related researchers to detect and prevent malicious applications.

### 3.3. Research and Sample Acquisition of High-Frequency Applications

This paper through the research on domestic application stores, such as "Huawei app store", "application treasure", "Xiaomi application store" and so on, find "game" and "financial" category applications account for more, and malicious apps are prone to exist in these types of apps. At the same time, according to the actual application distribution in the stores, this paper obtains the description information of categories such as "Movie", "Music" and "Traffic". Since some descriptive information applications in different categories of inconsistent application stores, part of the description of the quality is not high, the paper proposes a method of labelling categories based on keyword distribution. This method can classify the application description information and quickly obtain the category labels.

This paper obtains the Chinese description information of the corresponding categories in a number of regular stores, such as "application treasure" and uses the application's label as the corresponding description label to obtain a large number of data sets, with a data volume of 310,000. And based on 310,000 description information, combined with the class labelling method, the data set is expanded.

### 4. Category and Data Set Expansion Based on Keyword Sets and JS Divergence

In the process of constructing the application description data set, we found that the application description in the domestic application store has problems such as poor content quality and inconsistent category labels. In order to facilitate the collection and expansion of data, this paper investigates the application category naming rules of "Google Play" and "App Bao", and designs a total of 17 application description categories, and crawls 310,000 descriptive information that conforms to the application category naming rules from "App Bao".

In this paper, keywords between different categories are extracted by TF-IDF and TextRank algorithm, and constitute a keyword set. Map 310,000 pieces of descriptive information from the "application treasure" to the keyword set in the form of word bags, and calculate the keyword distribution P for each category. This paper uses the JS divergence to calculate the distance between the description to be annotated and the keyword distribution of the existing category, and takes the category with the smallest JS divergence calculation distance as the currently described category to complete the category method based on the keyword set and JS divergence.

Considering the situation that the description information of some application stores is poor, it is impossible to directly classify and expand the application descriptions. In this paper, a data set expansion scheme based on category is designed. For garbled descriptions, no keyword descriptions, etc., manual labelling or direct abandonment is given. The specific process of the scheme is shown in figure 4.

This paper crawls 280,000 application descriptions in application stores such as "Application Hui" and "Kuan Network", and obtained 250,000 effective description information through the data set expansion scheme, showing good usability. And finally merged into 560,000 pieces of description information as a data set for description classification.

After obtaining 560,000 pieces of descriptive information, this paper finds that there is a huge sample imbalance among categories. During the analysis of the description information, we found that the partial results of the "transportation" and "tourism" categories have the same word segmentation results. For example, keywords such as "car" and "train" appear in both categories, because both...
categories are related to transportation, while the former focuses more on the use of transportation, and the latter focuses more on the selection of services such as "ticket transportation". Therefore, it can be seen that the application descriptions under different tags have different emphasis on the same word segmentation results.

Although the high-frequency categories "game" and "finance" are more in absolute numbers than other categories, there are some subtle differences within the categories. "Game" is a large category, which contains several small categories such as "Enlightenment and Puzzle" and "Gun Shooting". In the division of the descriptions of "games", some keywords description inevitably overlap with other categories. For example, "Enlightenment and Puzzle" may overlap with "education". "Finance" is divided into small categories such as "bank" and "loan", but compared with other categories, the keyword overlap is not high. The similarity and intra-category differences in the data set test the model's recognition of these subtle differences.

![Flowchart](image)

**Figure 4.** Data set expansion scheme based on category annotation.
5. Short Text Classifier

5.1. CRNN Model
This article gets inspiration from dealing with time-series image problems [11], and attempts to migrate the Fast-CRNN model to the text classification task. Fast-CRNN first extracts the key information in the image with CNN, and inputs the key information into the RNN to complete the construction of timing information.

In order to solve the problem of similarity and difference between descriptions in the application description data set, this paper uses keyword extraction technology TF-IDF and TextRank algorithm to fully analyze sample features such as "traffic" and "tourism". Try using CNN convolution operation to extract key information in each category described, using RNN create timing context. Through the attention module to effectively capture the differences of keywords in different categories.

In order to solve the problem of sample imbalance in the description data set, this paper designs a model that uses Focal Loss algorithm to dynamically update the weight of each sample. Experiments show that the model can well recognize the difficult tasks such as key information and transitional context in the text. The model can quickly identify a large number of easy-to-learn samples in the samples, which are mostly concentrated in the "finance" and "health" categories; at the same time, it can also provide confidence.

5.2. Design of Network Structure
This configuration ideas from references [11], which is a paper based on an image sequence recognition problem. It uses a convolutional neural network feature extraction method and uses a circulating neural network modelling and sequences transcribed integration. This paper aims at the advantages and disadvantages of convolutional neural networks and recurrent neural networks, and constructs the network structure shown in figure 5. CNN can extract local key information, and use RNN to capture contextual context, and then form global features to enhance the accuracy of application description classification. The training algorithm based on the CRNN model can be represented in algorithm 1.

### Algorithm 1: The algorithm of training CRNN model.

| Input: | The word segmentation result Z described by the application. |
|-------|-------------------------------------------------------------|
| Output: | The corresponding predicted category. |
| 1: | Word Embedding X = Word2Vec(Z) |
| 2: | for each filter do |
| 3: | feature = conv2(filter, W, X) + b |
| 4: | end for |
| 5: | Feature Map = Combine(feature) |
| 6: | for each LSTM do |
| 7: | z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) |
| 8: | r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) |
| 9: | \hat{h}_t = \tanh (W \cdot [r_t \cdot h_{t-1}, x_t]) |
| 10: | h = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t |
| 11: | end for |
| 12: | Feature Vector = Combine(h) |
| 13: | Prediction = Softmax(Feature Vector) |
6. Experimental Results and Analysis

6.1. Dataset
Since the multi-classification problem is a supervised learning task, this article crawls a total of 310,000 pieces of descriptive information in the domestic app store “Appbao”. It uses category labelling tools based on keyword distribution to further obtain 250,000 pieces of descriptive information. After summarizing, the data set has a total of 560,000, including 17 categories, and the data information can be seen in table 1.

| Category                        | Training set | Validation set | Test set |
|---------------------------------|--------------|----------------|----------|
| Finance                         | 75600        | 7560           | 15120    |
| Travel & Local                  | 7200         | 720            | 1440     |
| News & Magazines                | 9900         | 990            | 1980     |
| Entertainment                   | 3600         | 360            | 720      |
| Shopping                        | 18000        | 1800           | 3600     |
| Social & Communications         | 9900         | 990            | 1980     |
| Education                       | 82800        | 8280           | 16560    |
| Health & Fitness                | 16200        | 1620           | 3240     |
| Travel & Local                  | 10800        | 1080           | 2160     |
| Music                           | 5400         | 540            | 1080     |
| Video Player & Editors Apps     | 4500         | 450            | 900      |
| Photography                     | 8100         | 810            | 1620     |
| House & Home                    | 3600         | 360            | 720      |
| Maps & Navigation               | 1800         | 180            | 360      |
| Express                         | 3600         | 360            | 720      |
| Games                           | 162000       | 1620           | 32400    |
| Productivity                    | 10800        | 1080           | 2160     |

Table 1. Details of Data set.

Figure 5. The framework of the CRNN model.
6.2. Environment and Parameter
In order to verify the validity of the model reasonably, compare the experimental environment settings: the operating system is Ubuntu16.04, the memory is 128 GB, the processor is Inter Xeon CPU E5-2620 v4@2.10Ghz*32, using Python3 language built under the TensorFlow framework Model.

In the model training and execution prediction stages, the text sequence length is uniformly set to 200; convolutional level dependent parameter reference document [6], set convolution kernel size 4; set the number of convolution kernels to 256; The number of LSTM layer neurons is set to 128; the training batch period is 20, and the result is output once every 100 rounds of iteration. The optimizer uses AdamOptimizer and the learning rate updater uses CyclicLR.

6.3. Analysis of Results
In order to ensure that the evaluation effect of test data is similar to the result of predicting unknown data in real scenarios, this paper directly uses the sample division method of the above data set in the training process. The models selected for the comparative experiment in this paper are four models: textCNN, RCNN, LSTM and CRNN. The training and validation sets using the model is trained, the test set test model results. Figure 6 is the F1 value results of each model, and Figure 7 is the average F1 value results of each model.

![Figure 6. F1-score of different models.](image)

The average F1 value of the textCNN model is 0.873, which performs poorly in the "traffic" and "tourism" categories. The reason is that the keyword similarity between the "transportation" sample and the "tourism" sample is relatively high. TextCNN extracts the representation of keywords according to the convolution operation, and uses the maximum pooling operation to select key information. This maximum pooling operation loses a lot of valuable information, resulting in the textCNN model not being able to identify the "traffic" and "tourism" categories.

The average F1 value of the LSTM model is 0.881, and its performance in the "traffic" and "tourism" categories has improved by 10% and 0.2% compared to textCNN, respectively. Analysis of these two types of samples reveals that although the similarity of the keywords is high, the narrative methods are different, that is, the context is different. Therefore, the LSTM model can be used to identify context differences well, and the recognition rate of "traffic" has been effectively improved.

RCNN average model F1 is 0.8688. The causes of poor effect in that the LSTM will be captured on model before and after the text information is directly spliced to the word vectors. This will disturb the information of the word vector itself, causing the model to always be in a fluctuating state, so it has not achieved good performance on the dataset.
The average F1 value of the CRNN model is 0.911, and its effect is significantly improved compared to the previous model. The reason is that the model is valid information extraction using convolution operation, and cyclic artificial neural network context, effectively captures differences between the same samples. The CRNN model uses the Focal Loss loss function and Attention module, which makes the accuracy of the model further increase.

![Figure 7. Average of F1-score of different models.](image)

By comparing the time-consuming training between models. As shown in Figure 8, it is found that except for the long time-consuming training of the RCNN model, the other three models are relatively close. The CRNN model improves the average F1 value index by 3.8% and 3% compared to textCNN and LSTM, respectively, and increases the training time-consuming index by 44% and 6% compared to textCNN and LSTM, respectively.

![Figure 8. Training time consumption of different models(second).](image)

7. Conclusion
In this paper, the text information of the application description is used to classify the application from a functional perspective. When the quality of the application description is not high, the design uses category labeling tools and data set expansion schemes based on keyword sets and JS divergences, effectively obtaining a total of 560,000 application description information. For the existence of inter-class application descriptions have similarities, there are differences within the category, sample imbalance problem, this paper designs a neural network model suitable for processing current tasks that combines the characteristics of CNN and RNN. This model solves the problem that it is difficult to distinguish the descriptions of similar functional categories. The average F1 value of this model is 0.911, which is about 3% higher than that of CNN, LSTM and other models. The time consumption is close to that of LSTM and other models, only an increase of 6%. When dealing with application description classification tasks, this paper obtained 560,000 training data for a total of 17 categories, which can provide training corpus for subsequent researchers and provide some guidance for application function classification.

References
[1] Wu L, Du, X Wu J 2014 MobiFish: A LightAndroid Multi-label Classification Research Based on Text Contentsweight Anti-Phishing Scheme for Mobile Phones 23rd International Conference on Computer Communication and Networks (ICCCN) (IEEE) pp 1-8
[2] Moon H, Jeon Y, Kim J 2014 Reducing the Impact of Repackaged App on Android 22nd International Conference on Information & Communication Technology Convergence (ICTC) IEEE pp 800-801.
[3] Li X 2019 Android malicious application detection model based on application classification (GuangZhou, China: GuangZhou University)
[4] Yuan C, Wei S, Wang Y, et al 2017 Android Applications Categorization Using Bayesian Classification 9th International Conference on Cyberenabled Distributed Computing & Knowledge Discovery (CyberC) IEEE pp 173-176.
[5] Liu Y, Liu L, Liu H 2019 Information Recommendation Based on Domain Knowledge in App Descriptions for Improving the Quality of Requirements IEEE Access pp 9501-9514
[6] Kim Y 2014 Convolutional Neural Networks for Sentence Classification 11th Conference on Empirical Methods in Natural Language Processing (EMNLP) pp 1746-1751
[7] Lipton Z, Berkowitz J, Elkan C 2015 A Critical Review of Recurrent Neural Networks for Sequence Learning, arXiv preprint arXiv:1506.00019.
[8] Lai S, Xu L, Liu K, et al 2015 Recurrent Convolutional Neural Networks for Text Classification 29th AAAI Conference on Artificial Intelligence pp 2267-2273
[9] China Internet Network Information Center 2019 The 44th Statistical Report on Internet Development in China [OL]. http://www.cac.gov.cn/2019-08/30/c1124938750.htm, 2019-08-30.
[10] 360 Security Center. 2018 China Mobile Security Ecological Research Report [R]. (Beijing: Qihoo Technology Co., Ltd.) 2018.
[11] Shi B, Bai X, Yao C 2017 An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition IEEE Transactions on Pattern Analysis & Machine Intelligence pp 2298-2304