Privacy Rating of Mobile Applications Based on Crowdsourcing and Machine Learning

Bin Pan, College of Management Science, Chengdu University of Technology, Chengdu, China
Hongxia Guo, School of Information Science and Engineering, Chengdu University, Chengdu, China
Xing You, Chengdu University of Technology, Chengdu, China
Li Xu, Chengdu University of Technology, Chengdu, China

ABSTRACT

With the advent of the 5G network era, the convenience of mobile smartphones has become increasingly prominent, the use of mobile applications has become wider, and the number of mobile applications has increased. However, the privacy of mobile applications and the security of users' private information are worrying. This article aims to study the ratings of data and machine learning on the privacy security of mobile applications and uses the experiments in this article to conduct data collection, data analysis, and summary research. This paper experimentally establishes a machine learning model to realize the prediction of privacy scores of Android applications. The establishment of this model is based on the intent of using sensitive permissions in the application and related metadata. It is to create a regression function that can implement the mapping of applications to score. Experimental data shows that the feature vector prediction model can uniquely be used to represent the actual usage and scheme of a system's specific permissions for the application.

KEYWORDS

Application Security, Data Collection and Analysis, Machine Learning, Mobile Application Security, Privacy Information Rating, Risk Data Research

1. INTRODUCTION

At present, there are few mobile application recommendation systems based on user trust behavior in existing work (Ramlatchan, 2018). The trust behavior of users using mobile applications can truly reflect their personal preferences, and their data is of great value for constructing user profiles and calculating recommendations. Secondly, the accuracy and personalization of existing recommendation systems still need to be improved. Existing mobile stores are basically scoring rules based on the number of downloads of applications and user evaluation, and such rules have the risk of low accuracy and malicious attacks. Last but not least, the issue of privacy protection in the mobile application recommendation system is still an open issue (Ghezzi et al., 2015). Since the recommendation result is based on the calculation of user data and even contains a lot of privacy data, if the user data cannot be properly protected, privacy leakage will occur.

The current Android system widely uses the permission model technology to manage and control the acquisition and access to information related to user privacy information (Li et al., 2019). However,

DOI: 10.4018/JGIM.20220701.oa5

the widespread application of mobile application platforms also has the problem of security and abuse of permission models. Many mobile applications often apply for unnecessary sensitive information permissions, so that users 'private information-related information often faces a huge risk of being leaked by malicious intentions. At the same time, many mobile applications will actively obtain and maliciously disclose the user's private information related information without the knowledge of the developer and the user. In recent years, a lot of research and work have begun to focus on the analysis of applications and the protection of the security and privacy of mobile application systems. Whether the permission of sensitive information behavior of the application is maliciously disclosed and whether it should be clearly allowed by the relevant departments, so a reasonable privacy information protection scheme using sensitive information permissions should be set up.

In response to the question of whether mobile phone privacy information may be exposed, M et al. Modified the software and used a virtual machine to implement dynamic mobile phone application flaw data analysis tool Taint-droid (Kumar et al., 2016; Xu et al., 2020). The advantage of this data analysis tool is that it can directly mark sensitive data as a taint source, and then track the data marked as taint at runtime, and analyze and judge based on the data marked as taint source and whether the data is leaked by the application Whether there may be leakage of mobile phone privacy information in the process of running the application (Wang et al., 2016). The study provides a continuous and automated application risk assessment framework, which automatically builds application models from mobile phone application metadata by automatically collecting mobile phone user responses to application permissions and use, and then uses a machine deep learning Methods to analyze and assess the risk of application (Wan et al., 2019). The research of Wei et al. Proposed that crowdsourcing data analysis technology can be used to study each user's expectation and acceptance of different privacy information combinations (Wei & Hou, 2016). By using this technology to detect the central expectations of each application and the amount of privacy information as well as the use rights and location, you can directly analyze the application core privacy information expectations and solutions for the use of information and behavior, That is used in third-party databases, or is required for the core functions of the application software (Ding et al., 2016). However, the results of the research indicate that the core expectations of users and the expectations and plans for the use of private information also directly affect each user's expectations and acceptance of the application's core private information and behavior.

The correlation analysis between the feature vector and the privacy score in the experiment shows that 74% of the features are positively correlated with the privacy score, while only 26% of the features are negatively correlated with the privacy score. Through experiments and data analysis, it is shown how to conduct research on the rating of privacy information features of mobile terminal applications based on software crowdsourcing and robot engineering learning technology (Lv & Qiao, 2020). First of all, the analysis and calculation results extract the privacy information feature vectors of the mobile terminal application and perform preprocessing. Then, we studied different machine learning and prediction model principles, created multiple privacy information risk scoring prediction models, and optimized them. At the same time, we compared with the existing research methods, and then used the Android application third-party library detection method to find the existing the detection method has certain limitations. This paper proposes and implements a third-party library detection method based on crowdsourcing data and module clustering, which provides a basis for the analysis of sensitive permission usage schemes, and compares with existing detection methods, reflecting the privacy information rating system of mobile applications Research results.

2. SENSITIVE AUTHORITY

2.1 Research Methods of Mobile Application Privacy Information

2.1.1. Performance Evaluation

The evaluation ability of the model requires an evaluation indicator. In this paper, the mean squared error (MSE) between the model 's predicted score and the actual application score is used as the main evaluation index of the model 's predictive ability. The smaller, the stronger the model's predictive ability and the better the model's performance (Yoeruek, 2017). In addition, k-fold cross-validation is used to automatically train the input data when constructing the model. The k value takes a positive integer between 1 and 10. Based on the experimental results, the k value that optimizes the prediction result will be taken. In the experiment, the input data is divided into k groups, one group is reserved for testing, and the other k-1 groups are used for training. This process is repeated k times, so that each group of data has a chance to be a test group, and then taken k times The average value of the mean square error of training is used as the final evaluation value. MSEk = i means that the k = i group of data is used as the mean square error value obtained by the training model of the test group. The calculation process of this value is shown in Equation (1), where Predictt represents the prediction score of the t-th vector of the application, Realt means the real score. MSE represents the mean square error value of the model, and the calculation process is shown in Equation (2):

$$MSE_{k=1} = \frac{1}{n}\sum_{t=1}^{n}(predict_t - real_t)^2 \tag{1}$$

$$MSE = \frac{1}{k}\sum_{i=1}^{k} MSE_{k=i} \tag{2}$$

2.1.2. Feature Combination

Feature combination refers to the recombination of basic features to construct higher-dimensional features. Combining features makes the problem linear in the low-dimensional feature space and characterizes non-linear features. If the samples cannot be linearly separated in the original feature space and the linear classifier is still used, it will lead to classification errors and algorithm errors (Peng et al., 2018). The combined features will undoubtedly increase the number of features and the complexity of the model (Guerrieri & Shimer, 2018). In many cases, the problem of overfitting the model may be exacerbated. But for large data sets, underfitting can be avoided. There are many ways to combine features, but a common method of combining features is to use the feature values of the original features to multiply each other, and the product forms a combined feature (Alshehri et al., 2017). The original features are called first-order features, and the doubled first-order features are called second-order features, where X is a feature vector. The characteristics of the second-order combination are as follows:

$$[X_1, X_2, \ldots, X_{15}] \rightarrow [X_1, X_2, \ldots, X_{15,} X_1^{\,2}, X_1X_2, \ldots X_{15}^{\,2}] \tag{3}$$

2.1.3. Interactive Recommendation Generation

This process is an interactive process including recommendation request users, SPs and PCs. In this process, users' final mobile application recommendations are generated (Mirarchi et al., 2016). Recommendation request users first send a request message to the SP to obtain recommendations. If the identity of the recommendation request user is verified by the PC, the SP will return a set of protected user relationship values to the recommendation request user (Gale et al., 2017; Shi et al., 2016). It is recommended to request the user to use the homomorphic encryption technology to process the user relationship value set, and then send the processed data to the SP. The SP uses the data sent by the recommendation request user, uses the homomorphic encrypted nature and the user data in the database, calculates the data necessary to generate the recommendation, and returns it to the recommendation request user (Kavitha et al., 2017; Zhang et al., 2016). Recommendation requests the user to calculate the final recommendation result in his own device. Specifically, the process has the following detailed steps:

1) The recommendation request user k sends the request for application recommendation to the recommendation cloud server SP using the anonymous identity through the client software as shown in formula (4), in which the user k signs his anonymous identity, which is the PC Signature of signature information (Luo et al., 2018). The SP receives the recommendation request information of the recommendation request user k and authenticates the user k to the privacy center PC. If the authentication fails, the SP rejects the request of user k. After the authentication is passed, proceed to the next step:

$$\{Sig_k(ID), Sig_{PC}(Sig_k(ID_k))\} \tag{4}$$

2) The recommended cloud server SP calculates the value of the cover relationship between the other user j and the recommended request user k in the system based on the formula (5) based on the identity ID of the recommended requester and the data of all system users stored in the database (Zhang et al., 2017), where and respectively indicate Other users and recommendation requesting user k indicate the usage trust behavior data of user k about application a (a ^{1} i indicates that the application is not to be recommended i) calculated within the time window t, and other similar symbols indicate similar meanings (Ma et al., 2016). Note that the user relationship value is actually protected within the time window t, the specificvalue is determined by the current time window:

$$\mathrm{Re}\,l_s(u_j, u_k) = E(s_1) * \sum_{a \neq i} \left(\sqrt{\frac{(T_a^k\left(t\right)_{UB} - T_a^j(t)_{UB})^2 + \left(T_a^k(t)_{CB} - T_a^j(t)_{CB}\right)^2 + \left(T_a^k(t)_{RB} - T(t)_{RB}{}^2\right)}{3}} \right) \tag{5}$$

3) The recommendation request user k uses its own homomorphic encryption public key to homomorphically encrypt the real relationship value to obtain formula (6), and then sends the relationship value set obtained by homomorphic encryption to the recommended cloud server SP (Lee et al., 2016):

$$HE\{HPK_k, \mathrm{Re}\,l(u_j, u_k)\} \tag{6}$$

4) It is recommended that the cloud server SP calculate the control parameters and send it to the recommendation requesting user k together with the output data and time stamp t in the algorithm (Hudson et al., 2016). The recommendation request user k receives the relevant data returned by the recommendation cloud server SP on the client, and decrypts the encrypted values obtained in the algorithm, namely X and Y, using their own homomorphic encryption private keys, and the decryption results are recorded as P and Q (Li & Sugimoto, 2018; Piao et al., 2016). For each application i to be recommended in the decryption set, the client calculates the recommendation value for the application i for the recommendation request user k obtained within the time window t according to formula (7):

$$R_i^k = \frac{Q}{E(s) * P} * N(i = 1,2,...,I) \tag{7}$$

2.2 Permissions and Usage Plan

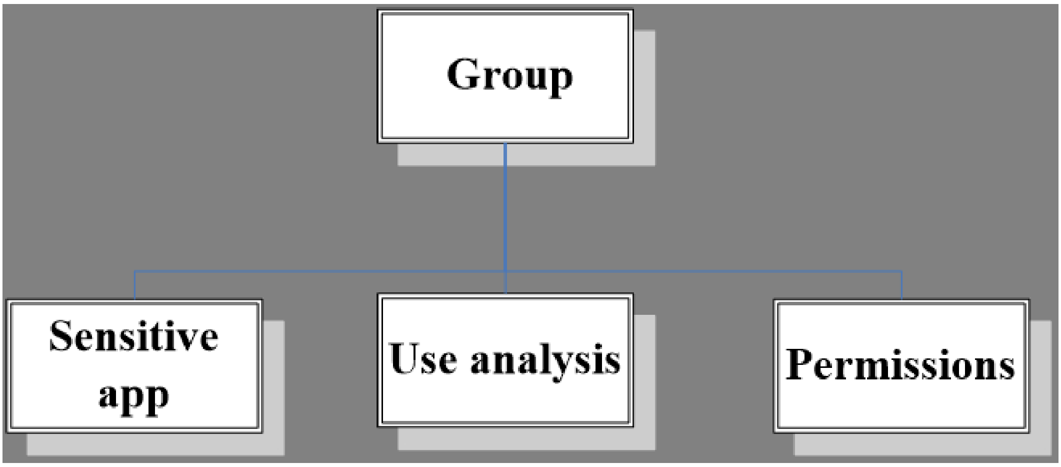
Static analysis technology is currently the most commonly used analysis technology in mobile application permission analysis. Through static analysis technology, it is easy to implement sensitive application permission analysis and analysis of usage permission schemes. In this paper, the decompilation process of the original mobile application Apk file is converted into an intermediate code by using the decompilation permission analysis tool Apktool (Son et al., 2017). On the one hand, we can directly obtain the application file with sensitive permissions of the mobile application. The sensitive application permissions actually required in the application file can be used in the mobile application code or the application code of the third-party database. The scheme of the file can directly obtain some privacy information features related to the application privacy information in the file. For example, the number of installation packages for each application component, the number of application components applying for sensitive permissions, the size of the installation package and other sensitive permission features (Dietze et al., 2016; Xie et al., 2016). In addition, it also indicates that we can analyze the mutual transfer time relationship between files and Apk in Smali's application intermediate code through the permission scheme to obtain the sensitive permissions that the application actually needs to use. Whether the sensitive use permission of application privacy information is reasonable is closely related to its sensitive use permission scheme, so this article analyzes its sensitive use permission scheme in different mobile applications for 11 sensitive application permission schemes that are frequently applied and used in mobile applications, and obtains < Sensitive applications, permissions, usage analysis> triples (Han & Keskin, 2016; Pentina et al., 2016; Saturday et al., 2017). The flow chart of the sensitive use permission scheme in different mobile applications is shown in Figure 1:

3. EXPERIMENTS ON OBTAINING PRIVATE INFORMATION IN MOBILE APPLICATIONS

3.1 Experimental Data Set

Crowdsourcing is used to investigate the actual data users expect from private information in mobile applications (Qi et al., 2018). Few users read end user license agreements or privacy rights because of the complexity of their privacy policies or the imbalance between user payment time and income, but crowdsourcing technology can successfully solve these problems. Providing a clear explanation reduces the complexity of understanding permissions, and pay attention to what behavior of the application is contrary to user expectations. First, ask participants to read some basic information, screenshots, and instructions about the applications provided by the Google Store. Next, ask one group of participants

Figure 1. Flow chart of the sensitive use permission scheme in different mobile applications



what they expect from using application permissions, and inform the other group of participants about their specific intent to use permissions (Hung et al., 2019). The experiment requires a comfort score for the app 's permission-related behavior, ranging from (uncomfortable) to +2 (very comfortable).

3.2 Experimental System

In this experiment, Android 6.0 and Android 5.0 were compared with the system behavior monitoring process for mobile application authorized applications. The main functions supported by Android are C ++ libraries, such as Opensqlite database kernel support, Opengles library 3d graphics browser support, Webkit library 3d browser database kernel support (Hamidi & Mousavi, 2018). At the same time, there are multiple Android application runtime libraries at this functional level, which provide Java core libraries, so Android developers can directly use their own Java language library to edit and run Android applications. This also includes a running Dalvik virtual machine, but since it was later changed to a javart running virtual machine environment, each Android application has its own running process and Dalvik's running virtual machine. The instance can be run in other Java virtual machines. Compared with the environment, Dalvik's virtual machine execution environment is specially customized for Android device developers on tablets and mobile phones, and the execution memory and CPU performance are greatly optimized.

3.3 Feature Vector Preprocessing

This article uses a feature vector of length 24 to represent the application. Since the feature data type has both digital and non-numeric features, the original feature data needs to be preprocessed, and different types of features are processed differently.

3.3.1. Normalization

Normalization is a method of simplifying calculations, which is to convert dimensional expressions to non-dimensional expressions, and standardized data is optionally limited to a certain range. The characteristic data levels of the data types used in this article are completely different. For example, the characteristic value of "download" is usually tens of thousands or even millions, and "component count" is in the single-digit level. The normalization process can reduce the impact of certain dimensions on the results. On the other hand, it can speed up the execution speed of the program. There are many ways to normalize data, but this article uses a linear transformation function to normalize numerical data.

3.3.2. Labeling

Non-numeric feature data has features such as application classification, permissions, and usage intentions. In this paper, the model must be trained after the non-numeric features of the training set are labeled. Before entering the model for privacy score prediction, all functions that require prediction score must be marked in advance. In this article, a hot code is used to mark non-numeric features. For each non-numeric function, assume a possible value of N. After a hot encoding, it is represented by a codeword of length N, where each bit is 0 or 1, and only one position in the codeword will have a value of 1, indicating the corresponding value.

3.4 Model Building

The purpose of this research is to establish a machine learning model for predicting the privacy score of Android applications. The model is based on the use of sensitive permissions in applications and related metadata. This is to create a regression function that can be applied to the evaluation graph. Establishing a prediction model is an iterative process. It is necessary to choose the model reasonably, conduct multiple comparative studies and select the best regression model as the prediction model based on the experimental data.

3.5 Experimental Operation

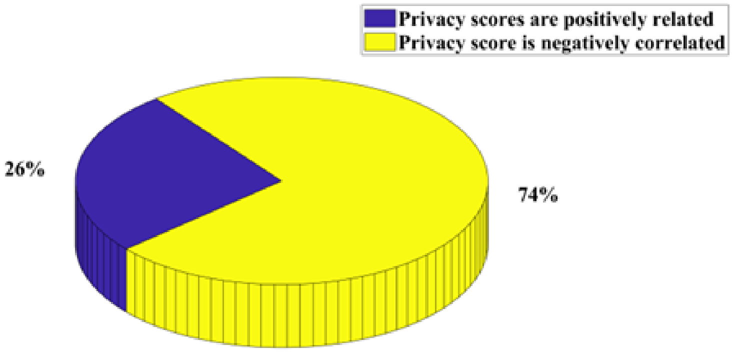
(1) Conduct random surveys of massive data and collect data by issuing questionnaires.
(2) Screening and analyzing data to ensure that the amount of effective data reaches the prescribed amount.
(3) Through the classification of big data, the general data distribution of the questionnaire survey is obtained.
(4) Analyze the proportion of sensitive permissions required by application software, and adopt the technical principle of feature model for data analysis.
(5) Organize all the valid data and use the formula to calculate the proportion of the number of privacy permissions required by the application software.
(6) According to the user's satisfaction, the normalized processing principle is used to process the data, and the relationship between permission use and user satisfaction is obtained.
(7) Finally, based on the two Android systems, the mobile application privacy rating analysis.

4. DISCUSSION ON THE CORRELATION BETWEEN PRIVACY AND FEATURE VECTOR

4.1 Correlation Analysis of Privacy Score

(1) Multiple secret privileges can be used in an application, and there may be multiple intents for the same secret privilege. Therefore, the number of <application, permission, intention> triplets associated with each application is different. In other words, the prediction model must output an unspecified number of scoring vectors (AppScore) for each input application to determine a reasonable scoring strategy. The final privacy score (FinalScore) of the application is determined according to the predicted application score vector, and the application privacy score level (PrivacyRate) is determined according to the final score. The method used in this article is to sort all the predicted values first, and then check whether they are negative. In the model built with the <authorization, intention> feature group as the core, the front.The accuracy of the prediction results of the five basic models (BR, LS, LN, LL, RD) is relatively close, and the MSE values all fluctuate around 0.4. The MSE values are all around 0.2, and the predictive ability of the combined model is significantly better than that of the basic model. The existence of a negative

Figure 2. Correlation analysis of features and privacy score



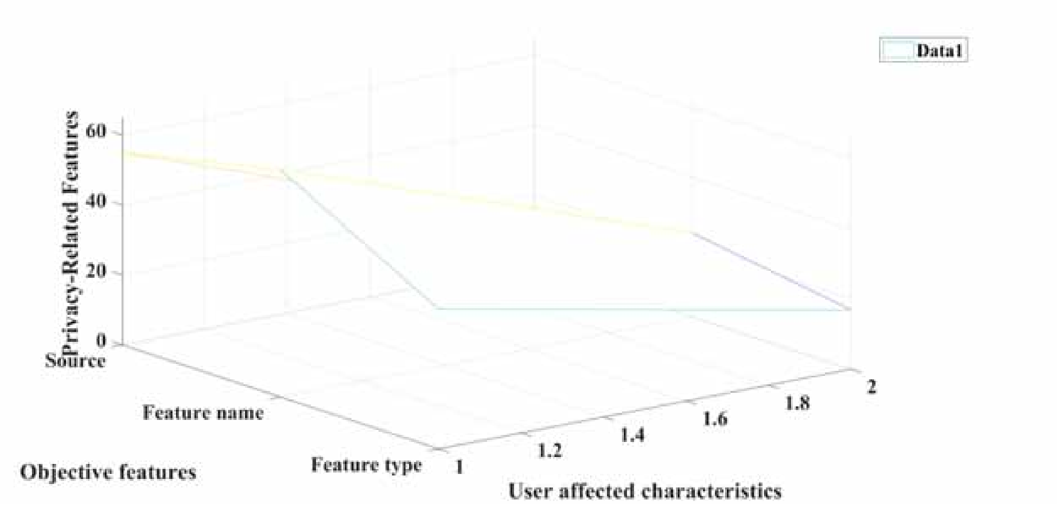
value indicates that the application has a user's unwelcome permission usage, and the sum of all negative values is used as the final privacy score. In the absence of negative values, it is more reasonable for applications to use sensitive permissions, and all positive values will be averaged into the final privacy score (FinalScore). Calculate the Pearson coefficient and score of each function to analyze the correlation between each function and the privacy score of the application. The range of correlation and Pearson coefficient is [-1,1], and the greater the absolute value, the correlation The stronger. Positive values indicate a positive correlation between features and levels, and negative values indicate a negative correlation between features and levels. This shows that 74% of features are positively correlated with privacy scores, while only 26% of features are negatively correlated with privacy scores. The correlation analysis results of function and privacy score are shown in Figure 2:

(2)The feature vector prediction model can be used to uniquely represent the actual usage and scheme of the system's specific permissions for the application. Before creating the feature vector prediction model, you need to create a feature vector set. This article describes how to extract multiple application feature data into a feature vector constructed by an application itself. Its feature data comes from two major parts, and some of the data containing features may come from "Metadata", that is, users can directly obtain most of the feature data related to an application itself from an application's software store, such as a Related data such as application downloads, number of evaluation users, star ranking, etc .; another part of the feature data may need to be extracted from the data source file of the application software installation package through static data analysis. In this paper, various characteristic data of an application are divided into three categories. According to experimental data, these three characteristics are distributed in proportions of 40%, 50%, and 10%. The distribution of the three feature ratios and data sources are shown in Table 2. The classification, name, and degree of user-affected features of most feature data are shown in Figure 3:

Table 1. Three characteristic proportion distributions and data sources

Feature type	Feature name	Source
User affected characteristics 40%	40%	17%
Objective features 50%	65%	24%
Privacy-Related Features 10%	55%	13%

Figure 3. Three characteristic proportion distributions and data sources



4.2 Correlation and Feature Vector Analysis

(1) The evaluation results of each model show that the progressive gradient regression tree has the best prediction effect. In order to further improve the performance of the model, please analyze the impact of various features on the prediction results of the model, and delete some features that do not significantly contribute to the prediction results. This experiment strives to reduce the dimension of features while maintaining the accuracy of the prediction model. First, calculate the importance of each function in the model. As shown in Figure 4, PermissionName (license name) and purpose (license use) are the two most important functions of the model, DName (whether to provide a developer name) and DEmail (development). The function importance value (with or without personal mailbox) is 0. This shows that DName and DEmail have little effect on the prediction function of the model, and basically do not affect the prediction results of the application. Analyzing the data in the training set, these two functions are Boolean variables that indicate whether the application has a developer name and email address. Another feature related to developers is DWeb, which indicates whether the developer's personal homepage is available. First of all, this function is more important than DName or DEmail. The statistical analysis of the training set data shows that the values of these three features have hardly changed. As shown in Table 2, the ratio of the three features is 1 and the ratio exceeds 94%. The Pearson coefficients of privacy scores are all less than 0.1, indicating that the correlation between these two features and privacy scores is weak. Therefore, these three features have been deleted from the feature

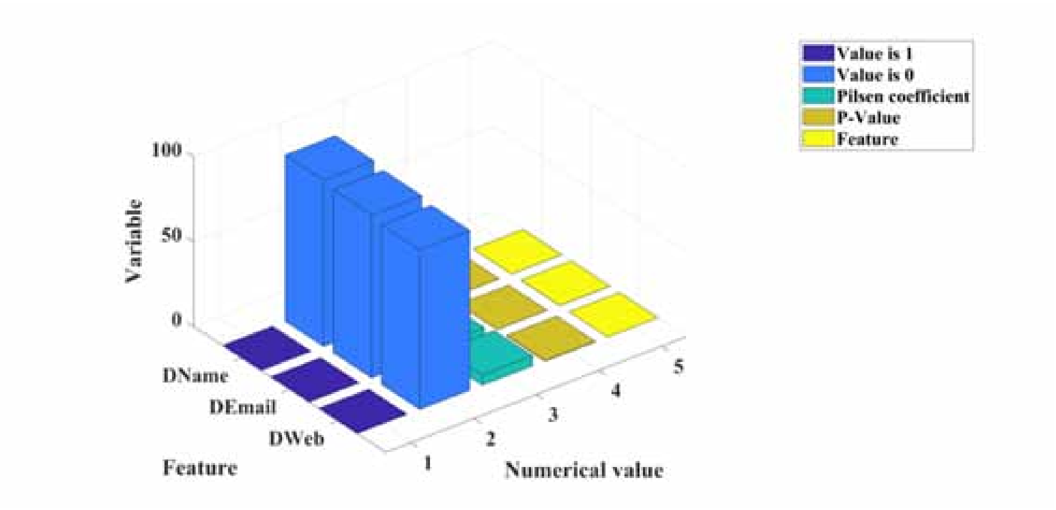
vector, and the remaining features are used to build a new model. So it can be seen that the MSE value is 0.209. Compared with the MSE value of the model before deleting these three features is 0.193, the inaccuracies of these three features in the prediction results are eliminated, and the impact is very small. Table 2 shows the distribution data of Boolean type features:

Table 2. Distribution of boolean type features

Feature	Proportion/%(Value is 1)	Proportion/%(Value is 0)	Pilsen coefficient	P-Value
DName	98	2	0.098	0.0334
DEmail	96	4	0.076	0.0956
DWeb	94	6	0.259	<0.0001

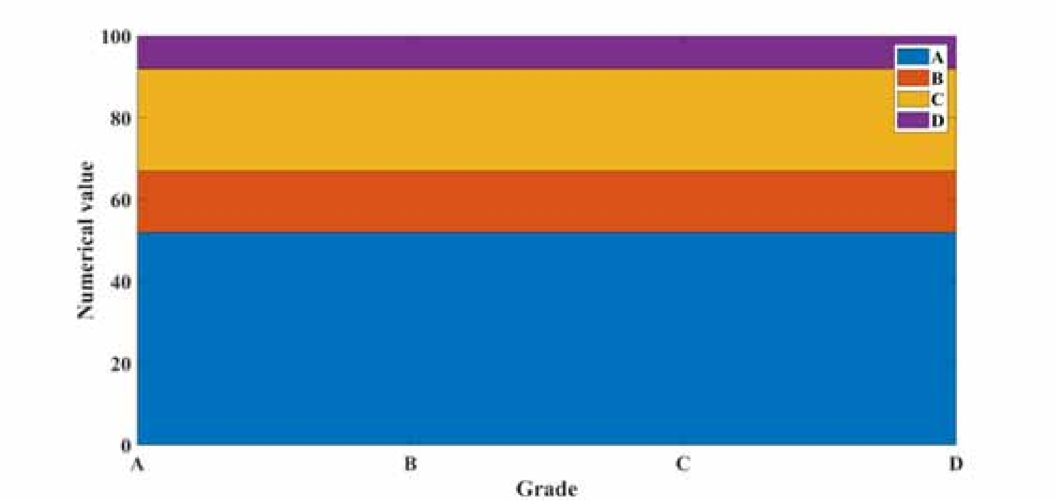
(2)Extend the application used as the predictor of the private information scoring application to the 10340 Android application in the Android App Store, decompile each Android application in turn, and then delete the confidentiality actually required by each application. Detailed analysis of high privileges, usage, and other requirements is based on feature vectors constructed from the feature subset data of the application. After a simple data preprocessing operation, this experiment only inputs models that can predict the score of privacy information, so that the application can be determined the privacy score of the program. The distribution of the prediction results is that the four application levels have different sensitive permission levels, and the a-level applications account for 52% of the sensitive permission levels. Most applications rated as b and a have a large percentage. The application is an application that does not directly use sensitive permissions, so such applications do not directly use sensitive permissions. In other words, they do not have sensitive permissions to directly access other users' personal information records or data. Therefore, the security level of this reputation level application is relatively high. Set the application score level to the highest sensitivity. Among them, applications with highly sensitive

Figure 4. Distribution of boolean type features



permissions of levels b and c are relatively close to 15% and 25%, respectively, and the proportion of applications with category d is 8% (Figure 5).

Figure 5. Proportion of four ratings of a, b, c and d



5. CONCLUSION

(1) This article proposes a method for evaluating the privacy of mobile applications based on the intention and expectations of the use license, and implements an evaluation tool. This article uses static analysis techniques to obtain the sensitive permissions actually used in each application to analyze its intent, and then combines them with other dimensional function applications to construct feature vectors and implement machine learning methods. Used to build regression models. In the experimental study of this paper, through the statistical analysis of the training set data, it can be seen that the values of these three features have little change, and the proportion of the three feature values is 1 exceeds 94%.

(2) This article implements an evaluation tool that uses data analysis techniques to obtain the sensitive permissions that each application actually uses, analyzes its intention to use it, and uses it in combination with other functional dimensions. Use machine learning techniques to construct vectors and build regression models. A prediction model of the privacy risk level of mobile applications based on the expected use of permissions is established. The model can predict the use of personal information that the user cares about and display it in stages, and this is intuitive and effective. To enable the analysis of mobile application license intentions, first obtain all sensitive permissions used by the application, and then use sensitive permissions and intent as the core function of the predictive model to analyze the intent of each sensitive permission.

(3) The experimental results show that the larger the absolute value, the stronger the correlation. A positive value indicates a positive correlation between the function and the evaluation, and a negative value indicates a negative correlation between the function and the evaluation. Correlation analysis between features and privacy scores shows that 74% of features are positively correlated with privacy scores, while only 26% of features are negatively correlated with privacy scores. At the end of the experiment, the mobile application has various mobile applications with sensitive permissions and sensitive permissions are easy to develop. The privacy level of

mobile applications helps users of mobile software to clearly understand whether their mobile applications disclose any privacy-permitted information, which is very beneficial to network security and is a very applicable practical guide.

ACKNOWLEDGMENT

This work was supported by youth fund project of Sichuan province education office(No. 10ZB088), youth key teachers training plan of Chengdu university of technology(No. 10912-JXGG2014013).

REFERENCES

Alshehri, A., Hewins, A., & Mcculley, M. (2017). Risks behind Device Information Permissions in Android OS. *Communications and Network, 9*(4), 219-234.

Dietze, R., Hofmann, M., & Ruenger, G. (2016). Water-Level scheduling for parallel tasks in compute-intensive application components. *The Journal of Supercomputing*, *72*(11), 4047–4068. doi:10.1007/s11227-016-1711-1

Ding, Z., Adachi, F., & Poor, H. V. (2016). The Application of MIMO to Non-Orthogonal Multiple Access. *IEEE Transactions on Wireless Communications, 15*(1), 537-552.

Gale, T. C. E., Lambe, P. J., & Roberts, M. J. (2017). Factors associated with junior doctors' decisions to apply for general practice training programmes in the UK: Secondary analysis of data from the UKMED project. *BMC Medicine*, *15*(1), 220. doi:10.1186/s12916-017-0982-6 PMID:29268742

Ghezzi, A., Cavallaro, A., Rangone, A., & Balocco, R. (2015). On business models, resources and exogenous (dis)continuous innovation: Evidences from the mobile applications industry. *International Journal of Technology Management*, *68*(1-2), 21–48. doi:10.1504/IJTM.2015.068777

Guerrieri, V., & Shimer, R. (2018). Markets with Multidimensional Private Information. *American Economic Journal. Microeconomics*, *10*(2), 250–274. doi:10.1257/mic.20160129

Hamidi, H., & Mousavi, R. (2018). Analysis and Evaluation of a Framework for Sampling Database in Recommenders. *Journal of Global Information Management*, *26*(1), 41–57. doi:10.4018/JGIM.2018010103

Han & Keskin. (2016). Using a Mobile Application (WhatsApp) to Reduce EFL Speaking Anxiety. *Gist Education & Learning Research Journal, 12*(12), 29-50.

Hudson, C. M., Mccurry, M. R., Lundgren, P., McHenry, C. R., & Shine, R. (2016). Constructing an Invasion Machine: The Rapid Evolution of a Dispersal-Enhancing Phenotype During the Cane Toad Invasion of Australia. *PLoS One*, *11*(9), e0156950. doi:10.1371/journal.pone.0156950 PMID:27658247

Hung, W.-H., Chang, I.-C., Chen, Y., & Ho, Y.-L. (2019). Chang, I-Cheng; Chen, Yan; Ho, Ying-Li. Aligning 4C Strategy with Social Network Applications for CRM Performance. *Journal of Global Information Management*, *27*(1), 93–110. doi:10.4018/JGIM.2019010105

Kavitha, J., Rani, P A J., & Sowmyayani, S. (2017). Wavelet-Based Feature Vector for Shot Boundary Detection. *International Journal of Image and Graphics*, *17*(01), 526–105. doi:10.1142/S0219467817500024

Kumar, Chan, & Coussens. (2016). Inflammation and Cancer. *Encyclopedia of Immunobiology, 420*(6917), 406-415.

Lee, J., Seko, A., & Shitara, K. (2016). Prediction model of band gap for inorganic compounds by combination of density functional theory calculations and machine learning techniques. *Physical Review B, Condensed Matter and Materials Physics, 93*(11), 115104.1-115104.12.

Li, C., & Sugimoto, S. (2018). Provenance description of metadata application profiles for long-term maintenance of metadata schemas. *The Journal of Documentation*, *74*(1), 36–61. doi:10.1108/JD-03-2017-0042

Li, H., Yang, M., & Evans, S. (2019). Classifying different types of modularity for technical system. *International Journal of Technology Management*, *81*(1-2), 1–23.

Luo, A., An, F., Zhang, X., Chen, L., Huang, Z., & Mattausch, H. J. (2018). Flexible feature-space-construction architecture and its VLSI implementation for multi-scale object detection. *Japanese Journal of Applied Physics*, *57*(4S), 04FF04. doi:10.7567/JJAP.57.04FF04

Lv, Z., & Qiao, L. (2020, April 20). Deep belief network and linear perceptron based cognitive computing for collaborative robots. *Applied Soft Computing*, *92*, 106300. doi:10.1016/j.asoc.2020.106300

Ma, X., Liu, Q., He, Z., Zhang, X., & Chen, W.-S. (2016). Visual Tracking via Exemplar Regression Model. *Knowledge-Based Systems*, *106*(C), 26–37. doi:10.1016/j.knosys.2016.05.028

Mirarchi, F. L., Ray, M., & Cooney, T. (2016). TRIAD IV: Nationwide Survey of Medical Students' Understanding of Living Wills and DNR Orders. *Journal of Patient Safety*, *12*(4), 190–196. doi:10.1097/PTS.0000000000000083 PMID:24583955

Peng, M., Zeng, G., Sun, Z., Huang, J., Wang, H., & Tian, G. (2018). Personalized app recommendation based on app permissions. *World Wide Web (Bussum)*, *21*(1), 89–104. doi:10.1007/s11280-017-0456-y

Pentina, I., Zhang, L., Bata, H., & Chen, Y. (2016). Exploring privacy paradox in information-sensitive mobile app adoption: A cross-cultural comparison. *Computers in Human Behavior*, *65*, 409–419. doi:10.1016/j.chb.2016.09.005

Piao, Y., Jung, J. H., & Yi, J. H. (2016). Server-based code obfuscation scheme for APK tamper detection. *Security and Communication Networks*, *9*(6), 457–467. doi:10.1002/sec.936

Qi, L., Dou, W., Wang, W., Li, G., Yu, H., & Wan, S. (2018). Dynamic Mobile Crowdsourcing Selection for Electricity Load Forecasting. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 46926–46937. doi:10.1109/ACCESS.2018.2866641

Ramlatchan, M. (2018). A survey of matrix completion methods for recommendation systems. *Big Data Mining and Analytics*, *1*(4), 308–323. doi:10.26599/BDMA.2018.9020008

Saturday, E. G., Li, Y. G., Ogiriki, E. A., & Newby, M. A. (2017). Creep-Life Usage Analysis and Tracking for Industrial Gas Turbines. *Journal of Propulsion and Power*, *33*(5), 1305–1314. doi:10.2514/1.B35912

Shi, B., Chen, N., & Wang, J. (2016). A credit rating model of microfinance based on fuzzy cluster analysis and fuzzy pattern recognition: Empirical evidence from Chinese 2,157 small private businesses. *Journal of Intelligent & Fuzzy Systems*, *31*(6), 3095–3102. doi:10.3233/JIFS-169195

Son, Y., Oh, S., & Lee, Y. S. (2017). Design and Implementation of the RSIL to LLVM IR Translator for Verification of the Intermediate Code on IoT Virtual Machine. *International Journal of Control & Automation*, *10*(10), 123–134. doi:10.14257/ijca.2017.10.10.11

Wan, S., Qi, L., Xu, X., Tong, C., & Gu, Z. (2019). Deep Learning Models for Real-time Human Activity Recognition with Smartphones. *Mobile Networks and Applications*, 1–13. doi:10.1007/s11036-019-01445-x

Wang, T., Duong, T. D., & Chen, C. C. (2016). Intention to disclose personal information via mobile applications: A privacy calculus perspective. *International Journal of Information Management*, *36*(4), 531–542. doi:10.1016/j.ijinfomgt.2016.03.003

Wei, L., & Hou, J. L. (2016). The guideline of prevention and treatment for hepatitis C: A 2015 Update. *Chinese Journal of Hepatology*, *23*(12), 906–923. PMID:26739465

Xie, S., Chen, L., Zuo, N., & Jiang, T. (2016). DiffusionKit: A light one-stop solution for diffusion MRI data analysis. *Journal of Neuroscience Methods*, *273*(273), 107–119. doi:10.1016/j.jneumeth.2016.08.011 PMID:27568099

Xu, X., Zhang, X., Khan, M., Dou, W., Xue, S., & Yu, S. (2020). A Balanced Virtual Machine Scheduling Method for Energy-Performance Trade-offs in Cyber-Physical Cloud Systems. *Future Generation Computer Systems*, *105*, 789–799. doi:10.1016/j.future.2017.08.057

Yoeruek, A. (2017). Using 2D simulation for the determination and verification of the extreme high water range of rating curves. *Wasserwirtschaft*, *107*(7-8), 35–37.

Zhang, T., Wang, J., Huang, J., Huang, Y., Chen, J., & Pan, Y. (2016). Adaptive marking threshold method for delay-sensitive TCP in data center network. *Journal of Network and Computer Applications*, *61*(C), 222–234. doi:10.1016/j.jnca.2015.10.012

Zhang, Y., He, Q., Xiang, Y., Zhang, L. Y., Liu, B., Chen, J., & Xie, Y. (2017). Low-cost and confidentiality-preserving data acquisition for internet of multimedia things. *IEEE Internet of Things Journal*, *5*(5), 3442–3451. doi:10.1109/JIOT.2017.2781737

Bin Pan was born in Guangan, Sichuan, P.R. China, in 1978. He received the Doctoral degree from University of Chinese Academy of Sciences, P.R. China. Now, he teaches in Chengdu University of Technology. His research interests include information system, big data analysis, and information service evaluation.

Hongxia Guo was born in Jingle, Shanxi, P.R. China, in 1976. She received the Master's degree from Southwest Petroleum University, P.R. China. Now, she works in Chengdu University, her research interests include communication technology, the internet of things and big data.

Xing You was born in Sichuan, P.R. China. He received the Master's degree from Chengdu University of Technology, P.R. China. Now, he teaches in Chengdu University of Technology. His research interests include management Science and information system.

Li Yu was born in Sichuan, P.R. China. He received the Master's degree from Chengdu University of Technology, P.R. China. Now, he teaches in Chengdu University of Technology. His research interests include management Science and applied statistics.