Android Apps Security Assessment using Sentiment Analysis Techniques: Comparative Study

https://doi.org/10.3991/ijim.v15i24.27359

Abeer Aljumah, Amjad Altuwijri, Thekra Alsuhaibani(✉),
Afef Selmi, Nada Alruhaily
Qassim University, Buraydah, Saudi Arabia
iofaausat@gmail.com

Abstract—Considering that application security is an important aspect, especially nowadays with the increase in technology and the number of fraudsters. It should be noted that determining the security of an application is a difficult task, especially since most fraudsters have become skilled and professional at manipulating people and stealing their sensitive data. Therefore, we pay attention to trying to spot insecurity apps, by analyzing user feedback on the Google Play platform and using sentiment analysis to determine the apps level of security. As it is known, user reviews reflect their experiments and experiences in addition to their feelings and satisfaction with the application or not. But unfortunately, not all of these reviews are real, and as is known, the fake reviews do not reflect the sincerity of feelings, so we have been keen in our work to filter the reviews to be the result is accurate and correct. This study is useful for both users wanting to install android apps and for developers interested in app optimization.

Keywords—sentiment analysis, android, google play, user review, mining threat, security

1 Introduction

Mobile applications are becoming increasingly common in people's daily lives as technology progresses at such a quick pace. This causes an increase in the number of applications in the application stores daily, which makes it difficult for users to choose the appropriate application for them in several aspects, the most important of which is the safety of that application. An estimated one in every 36 mobile devices has risky apps installed, according to some estimates [1]. Determining application security is difficult, especially for those who do not have the technical knowledge, so we decided to determine the level of security for the application by analyzing user reviews using sentiment analysis, which studies have proven successful in many areas.

Using a computer, sentiment analysis or opinion mining analyzes the way people feel and think about a wide range of topics, including products, services, issues, events, and themes. As a result, in order to track how the public feels about a specific entity,

sentiment analysis can be utilized. This knowledge can then be put to use. This type of knowledge can be used to understand, explain, and forecast social processes. When it comes to business, sentiment analysis is crucial for strategizing and gaining insight into customer opinion on a company's goods and services. In today's customer-focused company culture, knowing your customer is critical. Sentiment analysis incorporates elements of psychology, sociology, natural language processing, and machine learning. Data volumes and processing power have recently increased dramatically, allowing for more sophisticated forms of analytics. As a result, sentiment analysis using machine learning has grown increasingly popular [2].

The rest of this paper is organized as follows. Section 2 discusses related works about spam detection, aspect-based sentiment analysis and evaluation of mobile apps security. Section 3 focus on the proposes solution. Section 4 discusses methodology used in developing the proposed system. At the end section 5 explains the conclusions.

2 Related work

This section is divided into three fields related to this paper. In each field, we lay out some related studies and existing systems.

2.1 Spam detection

We found that several techniques and methods are suggested to help detect fake reviews with greater accuracy. One of the most effective ways is the process of extracting features from the text, that can be categorized into two main groups: Features related to review content, which focus on the text of the review by analyzing its features such as Bag of word (BOW), word embedding (WE), and term frequency (TF) etc [3][4]. While the second method focuses on the features of reviewers that include characteristics of the user who is posting reviews such as IP-address, the number of posts of the reviewer, etc. Using these two methods to spot spam reviews yields a better and more accurate result [5][6]. On the other hand, given the approaches used, most researchers have worked with supervised classification models, while few researchers have worked with unsupervised models. Unsupervised models can be hard due to the lack of a reliable labeled data-set of reviews. [6]

Supervised learning method of spam detection. Supervised learning is a classification method intended to train the machine through labeled data to predict the output. While this technology can be used to detect and assess assaults in numerous areas of cybersecurity, it is also known as Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), and Random Forests (RF). Also, it can be used to discover spam reviews which require a labeled dataset to detect spam of unseen data. [7]. Table 1 compere number of studies about supervised learning on spam detection.

Table 1. Studies of supervised learning on spam detection

No	Dataset	Size of Data	Feature Used	Algorithm Used	Language and Tool	Best Method	Metrics	Result
[8]	Dataset for Movie reviews V2.0 and V1.0	2000 reviews of V2.0, 1400 reviews of V1.0	POS	SVM, NB, and DT-J48, K-NN, KStar algorithms.	Weka tool	SVM	Accuracy Precision	81.35% to V2.0 76% to V1.0 81.1% to V2.0 74.9% to V1.0
[9]	Yelp dataset.	64195 reviews	TF LDA, WE (Word2Vec)	SVM, LR and Multi-layer Perceptron classifiers	Python 3.6., Liu Yang, Scikit-learn	LR with LDA, and MLP with LDA	Accuracy	81.3%
[10]	Yelp dataset (Restaurant, hotel).	Restaurant 67019 reviews, hotel 5858 Reviews	CF, PF, UTF.	NB, J48, RF, JRip and AdaBoost	Weka tool, AMT	AdaBoost	Precision Recall ROC	83%. 73.4%. 74%.
[11]	Yelp dataset (Hotel, Restaurant and Doctor).	No-mention	Bigram, POS, TF_IDF, MMI, PCA	DT classifier	No-mention	DT classifier	Recall precision F-measure	77.63%. 76.21%. 76.91%
[6]	hotel reviews	1,600 reviews	WF, sentiment polarity, length of review.	SVM, NB, EM	AMT to deceptive reviews, Python language with Scikit-learn and numpy packages.	NB	Accuracy	86.32%
[12]	Stony brook university	61541 reviews	Reviewers Feature, rating of reviews, WF.	LR, NB, SVM, XGBoost	XGBoost	XGBoost	Prediction, Recall, F1_score	0.99%. 0.99%. 0.99%.

Unsupervised learning methods of spam detection. Supervised learning requires a labeled dataset to train a model. Unsupervised learning has been suggested to overcome this problem. In fact, most research that uses supervised learning technique is based on pseudo fake reviews rather than real fake reviews [13]. Unsupervised learning approaches can be used for more ideas related to spam review discoveries as illustrated in Table 2.

Table 2. Studies of unsupervised learning on spam detection

No	Idea	Tool Used
[14]	They have improved the use of Review skeptic, to be also used in spam detection from non-hotel reviews by adding reviewer behavior and time criteria to it, Known as Review Alarm.	Review skeptic is automated tool that uses text-related criteria for defining hotel spam reviews.
[15]	They proposed a framework combining LSTM and Spam reviews can be identified using an LSTM-autoencoder. Instead of assigning a class label to a review, the goal is to teach a model to learn the patterns of real reviews from the specifics of the review's textual content. Therefore, if the new review contains representational words different from those learned in the training model that could be considered an anomaly from what it has learned, the system considers them spam reviews.	The model uses One Hot embedding to learn real review patterns and then calculate the reconstruct ion loss and cluster them by EM into spam or real review.
[16]	Predictive capabilities of review, reviewer, and product vector representations will be exploited by applying Doc2 and Node2 algorithms to a raw spam review dataSet. To distinguish between fraudulent and genuine reviews, give each one a separate vector representation. In order to create a classifier for spam review identification, the results from both steps are aggregated and fed into the logistic regression algorithm.	The Node2vec algorithm uses review metadata to create an underlying reviewer-product network that can improve the vector representation of each reviewer and product. At the same time, Doc2vec is used for generating document embedding from their textual content.
[13]	They applied the collective classification algorithm MHCC over the users-reviews-IP addresses' heterogeneous network to detect spam reviews. As there are likely to be many phony reviews buried in the unlabeled collection, the classifier could be confused, hence MHCC views unlabelled/unfiltered reviews as negative data. As a result, they tried to transform MHCC into a model of Collective PU learning. (CPU).	Only during initialization does the CPU model treat unlabeled data as negative. Classification results are evaluated and a reliable positive and negative state is generated based on the trained classifiers after the original classifier is known.
[17]	The system aims to reach high detection accuracy by using only a small number of positive labeled data sets and many unlabeled data. As well as, they used behavior density to improve the detection accuracy by doing a secondary check for spam reviews.	They used PU learning combined with behavior density to prevent users from spreading fake reviews in the App store.
[18]	By using relational data (review-user-product graph) and metadata (behavioral and textual data) and building a relationship between them, it aims to consider the challenge of spam detection as network classification task to uncover spammers, as well as products targeted by spam. In addition, the system can work as semi-supervised fashion by accepting a small set of labeled data, this version called SpEagle+. As well as SpLite version was launched that aims to reduce computational load, by relying on review features rather than the user and product features.	Metadata (such as labels, timestamps, and review content) supports network classification, as spam detection guides by extracting features from reviews, which are subsequently converted into an anti-spam score for inclusion in class priors.

2.2 Sentiment analysis

As a text-analysis technique, sentiment analysis aims to discover people's emotional polarity in the document as a whole (such as a positive or negative opinion). In addition to paragraph, sentence, or clause. It is now a common social media analysis tool carried out by companies, marketers, and political analysts [19]. Sentiment analysis has many types, such as Fine-grained, Emotion detection, Aspect-based, and Multilingual sentiment analysis. This section explains the research papers that are related to Aspect-based Sentiment analysis as shown in Table 3.

Table 3. Studies of aspect-based sentiment analysis

No	Dataset Source	Dataset Description	Sentiment Analysis Method	Aspect Extraction Method	Metrics	Result
[20]	Google Play	1200 review in "Beautiful widgets" and "Where is my Perry" apps	SAS® Sentiment Analysis Studio 12.1 is	SAS® Enterprise MinerTM 7.1	Precision	92% for rule-based models and 81% for a statistical model
[21]	Reviews from multiple social media platforms and websites	there are 2000 reviews for restaurant domain and 4000 reviews for hotel domain	-NBM -SVM -ME -RFT -FLR the best performance was achieved using NBM	hybrid aspect identification method	accuracy	88.08% on the restaurant's dataset and 90.53% on the hotel's dataset.
[22]	Google Play and Apple store	Over 12,000 reviews were written for 60 different government mobile apps in the United Arab Emirates.	utilizing techniques based on lexical and rule-based considerations	GARSA	accuracy	96.57%
[23]	ICLR	3,343 papers	*K*-means clustering	-FFNN-uni -MNB - RF - SVM-rbf -SVM-lin -BiLSTM-CNN -FFNN-sci The best performance was achieved using FFNN-uni	F1-score	71%
[24]	Not mention	100 reviews have been hand-selected and categorized for use in the study.	rule-based algorithm	PMI	F1-score	for AUTOMATED EXTRACTION OF SENTIMENT LEXICON: Positive: 0.619 Negative: 0.626 for EXTRACTION ASPECT TERM: Terms that occur a lot: 0.457 0.516 is a low-frequency word.

2.3 Evaluation of mobile apps security

Open-source nature of Android makes it the most popular smartphone operating system. Static analysis, dynamic analysis, and hybrid analysis are all methods used to check for Android security flaws.

Static analysis can't catch exploits being used in the wild. During runtime, data flows can be inspected to get around this limitation [3].

Static and dynamic analysis are combined in hybrid analysis. Using this technique, dynamic analysis data can be included into a static analysis program [25]. Table 4 [25] presents a comparison of static, dynamic, and hybrid analytic methods.

Table 4. The static, the dynamic, and the hybrid analysis techniques

	Static Analysis	**Dynamic Analysis**	**Hybrid Analysis**
Essential time	Fewer	Larger	Larger
Input	Permissions, source codes, Binary files	Runtime data as API in addition to Memory snapshots	Data from both static and dynamic analysis
Resource consuming (power consumption & memory consumption)	Fewer	Larger	Larger
Efficiency	Fewer in comparison with dynamic analysis	Better than static analysis	Better than static and dynamic analysis
Executing the code	Not probable	Probable	Probable
Benefits	Little cost and require fewer analysis time	Provide analysis in deep and more detection rate	Extract both features for static and dynamic analysis, Provide more accurate results
Limitations	The known malware forms are only detected	Additional time and power consumption	Cost is high

These methods rely on app functionality and must be installed first. We, on the other hand, aim to halt the installation process through the analysis of user reviews. Intriguing results suggest that customer reviews are beneficial in understanding customer sentiments through machine learning techniques methods. In order to notify programmers exactly where to enhance across updates, this information must be extracted and described efficiently from reviews [3]. The app's security suffers greatly when it is updated. There have been very few studies done on the effectiveness of applications in terms of security.

Evaluation of mobile apps based on reviews. This part explains the research papers related to the analysis of user reviews for security evaluation. An overview of previous studies and dataset used, methods applied, and analysis results in Table 5.

Table 5. Studies of review-based app evaluation

No	Dataset Details	Method and Tools	Results/Findings
[26]	36,464 comments from 3,174 apps.	Independent Logistic Regression is used as a baseline.	Experiments showed significant improvement against Independent Logistic Regression as a baseline method.
[27]	First dataset contains 6,526 apps. Second dataset contains 6,257 apps	Crowdsourcing through Two-Coin for client Ranking-SVM.	In comparison to other cutting-edge approaches, the results of the experiments demonstrated a 6–7% gain in performance.
[28]	Dataset of 19,413 reviews from 3,174 apps.	multi-class SVM with linear kernel.	As compared to the other approaches, AUTOREB excels by a large margin with 51.36% in accuracy.
[4]	64789 reviews from 17 mobile apps.	Vader Sentiment Analyzer, Stanford Parser7.	A user survey indicates the usefulness and feasibility of the summarization of SRR-Miner.
[29]	over 87K apps, 2.9M reviews, and 2.4M reviewers.	MLP, DT, RF.	According to FairPlay, 75% of the malicious programs have been found to involve in search rank fraud.
[3]	13 apps, 1050 security related reviews, 7,835,322 functionality-related reviews.	Naive Bayes classifier	Only 23% of applications had a reputation larger than 0.5, according to the findings.
[30]	Details of 35 Apps.	MySQL database, TextBlob library.	According to the findings, average ratings aren't a valid ranking system when compared to SERS.
[31]	812,899 user reviews of 200 apps within 10 app categories.	VADER sentiment analyzer, DT, K-Nearest Neighbours, RF, LR, and SVM classifiers.	LR got the highest accuracy among other algorithms.

Authors in [26] showed that Comments with Security/Privacy Issues (CSPI) must first be recognized to eliminate all those irrelevant comments to expose the issues related to an app's security/privacy. This paper presents a label system illustrating the "What" and "When" of the occurrence of an observed CSPI. A CSPI Detection with Comment Expansion (CDCE) approach is proposed, then a multi-label supervised learning technique is applied to classify diverse kinds of CSPI.

User comments aggregation treated as a crowdsourcing challenge for inferring security risks is [27]. User feedback may be used to create a new two-stage model that automatically ranks app hazards based on latent security labels.

Authors in [28] has developed the AUTOREB framework, which uses ML algorithms to automatically assess if the app has security-related behaviors from other users' experiences. Sort user evaluations according to four distinct security-related behaviors. To make predictions about app-level security issues, it employs crowdsourcing.

For extracting reviews, [4] suggest a Security-Related Review Miner instead of utilizing ML techniques (SRR-Miner). To begin, it extracts security-related review clauses using a keyword-based technique. Using established semantic patterns, it then pulls out

phrases that reflect bad behavior, attributes, and viewpoints. It uses triples to sum up security issues as well as user sentiment.

On the other hand, the proposed FairPlay framework [29] organizes the study into the following 4 modules to define malware and search rank fraud targets in Google Play. Moduls include the Co-Review Graph (CoReG), the Review Feedback (RF) and IRR/JH Relation modules. Several features are generated by each module and then sent into a classifier to be trained. As well as the average rating, the total number of downloads, and the number of reviews, FairPlay makes use of these more general features.

Authors in [3] provides a framework called CIAA-RepDroid, a fine-grained security-related reputation based on security-related sentiment analysis and probabilistic classification model. CIAA-RepDroid breaks down reputation into reputations of confidentiality, integrity, authentication, and availability.

In order to grade security claims, the SERS ranking scheme [30] proposes to use evidence-based security-related ranking. Static and sentiment analysis are both tools used by the authors. Sentiments about confidentiality are tallied. As a result, they obtain a high app rating, indicating that users have confidence that the app will not divulge any sensitive data.

Mobile App Reviews Summarization (MARS) was introduced by authors in [31] as a mechanism for summarizing reviews and extracting privacy concerns. Their mechanism has a precision of 94.84%, a recall of 91.30%, and an F-score of 92.79%. Privacy and security are treated as keywords in this paper, and the trustworthiness of apps is determined by whether or not they pose a threat to privacy.

3 Proposal

As our dependency on smartphones rises, so ensures our experience to security threats. Hence, the security level of apps downloaded on our smartphones must be a priority for us because Applications represent the largest security and privacy risk to a device and user's data [32]. For this purpose, users tend to evaluate the app's security level primitively by using some risk indicators such as the developer's reputation, the number of downloads of the application, the app rating, and the user reviews. But, since it is common for these indicators to be manipulated and Fabricated, users can not consider it trustworthy or sufficient to trust a specific app. This is where our work comes in. The proposed framework aims to produce a helpful tool to assess the risk of android apps in google play by identifying the security issues in apps based on the sentiment analysis of genuine user reviews.

4 Methodology

The study is constructed from seven steps:

Step 1: The collection of user reviews, after search and discussion we concluded to use the dataset from [32–34].

Step 2: Apply some preprocessing on the dataset like removing irrelevant and redundant information present or noisy and unreliable data to make it suitable and reliable for further analysis.

Step 3: Detect and exclude spam reviews by analyzing the list of behavioral and textual features of the review and the application using the review content, timestamp and rating associated with each review.
Step 4: We will start filtering user reviews to extract only reviews about security-related based on a list of keywords from two research [35][36].
Step 5: Apply sentiment analysis on filtered reviews.
Step 6: Categorize the reviews into many security aspects. by evaluating the distribution of apparition of security-related keywords in each security aspects.
Step 7: Deliver an assessment of each security aspect of the app and a global assessment for the app as a whole.

5 Conclusion

Our study is useful to users who are willing to install android apps and for developers interested in making an app better. It helps to Increase the awareness of users to combat suspicious apps, Boost Google Play's security and cut down on the number of attacks. Provide a comprehensive summary of security issues to users, as well as user-generated feedback regarding the app's vulnerabilities and misbehaviors, to developers.

6 References

[1] O'Gorman, B., Wueest, C., O'Brien, D., Cleary, G. et al. (2019) Internet security threat report Accessed: 22.8.2021 [Online] Available: https://img03.en25.com/Web/Symantec/%7B1a7cfc98-319b-4b97-88a7-1306a3539445%7D_ISTR_24_2019_en.pdf
[2] Quasim, M. T., Alkhammash, E. H., Khan, M. A. et al. (2021). Emotion-based music recommendation and classification using machine learning with IoT Framework. Soft Comput 25, 12249–12260 (2021). https://doi.org/10.1007/s00500-021-05898-9
[3] Tchakounté, F., Yera Pagor, A. E., Kamgang, J. C., and Atemkeng, M. (2020). CIAA-RepDroid: A Fine Grained and Probabilistic Reputation Scheme for Android Apps Based on Sentiment Analysis of Reviews. Future Internet, 12: 145. https://doi.org/10.3390/fi12090145
[4] Tao, C., Guo, H., and Huang, Z. (2020). Identifying security issues for mobile applications based on user review summarization. Information and Software Technology, 122: 106290. https://doi.org/10.1016/j.infsof.2020.106290
[5] Etaiwi, W., and Awajan, A. (2017). The effects of features selection methods on spam review detection performance. International Conference on New Trends in Computing Sciences (ICTCS), October 2017, IEEE, pp. 116–120. https://doi.org/10.1109/ICTCS.2017.50
[6] Hassan, R., and Islam, M. R. (2019). Detection of fake online reviews using semi-supervised and supervised learning. International conference on electrical, computer and communication engineering (ECCE), February, IEEE, pp. 1–5. https://doi.org/10.1109/ECACE.2019.8679186
[7] Alqudah, N., and Yaseen, Q. (2020). Machine learning for traffic analysis: a review. Procedia Computer Science, 170: 911–916. https://doi.org/10.1016/j.procs.2020.03.111
[8] Elmurngi, E., and Gherbi, A. (2017). Detecting fake reviews through sentiment analysis using machine learning techniques. IARIA/data analytics, Nov: 65–72.
[9] Jia, S., Zhang, X., Wang, X., and Liu, Y. (2018). Fake reviews detection based on LDA. 4th International Conference on Information Management (ICIM), May 2018, IEEE, pp. 280–283. https://doi.org/10.1109/INFOMAN.2018.8392850

[10] Khurshid, F., Zhu, Y., Yohannese, C. W., and Iqbal, M. (2017). Recital of supervised learning on review spam detection: An empirical analysis. 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), November 2017, IEEE, pp. 1–6. https://doi.org/10.1109/ISKE.2017.8258755

[11] Sedighi, Z., Ebrahimpour-Komleh, H., and Bagheri, A. (2017). RLOSD: Representation learning based opinion spam detection. 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS), December 2017, IEEE, pp. 74–80. https://doi.org/10.1109/ICSPIS.2017.8311593

[12] Sihombing, A., and Fong, A. C. M. (2019). Fake review detection on yelp dataset using classification techniques in machine learning. International Conference on contemporary Computing and Informatics (IC3I), December 2019, IEEE, pp. 64–68. https://doi.org/10.1109/IC3I46837.2019.9055644

[13] Li, H., Chen, Z., Liu, B., Wei, X., and Shao, J. (2014) Spotting fake reviews via collective positive-unlabeled learning. International conference on data mining, December 2014, IEEE, pp. 899–904. https://doi.org/10.1109/ICDM.2014.47

[14] Pieper, A. T. (2016). Detecting review spam on amazon with reviewalarm (Bachelor's thesis, University of Twente).

[15] Saumya, S., and Singh, J. P. (2020). Spam review detection using LSTM autoencoder: an unsupervised approach. Electronic Commerce Research, 1–21. https://doi.org/10.1007/s10660-020-09413-4

[16] Yilmaz, C. M., and Durahim, A. O. (2018). SPR2EP: A semi-supervised spam review detection framework. ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), August 2018, IEEE, pp. 306–313. https://doi.org/10.1109/ASONAM.2018.8508314

[17] He, D., Pan, M., Hong, K., Cheng, Y., Chan, S., Liu, X., and Guizani, N. (2020). Fake Review Detection Based on PU Learning and Behavior Density. IEEE Network, 34: 298–303. https://doi.org/10.1109/MNET.001.1900542

[18] Rayana, S., and Akoglu, L. (2015). Collective opinion spam detection: Bridging review networks and metadata. 21th ACM SIGKDD international conference on knowledge discovery and data mining, August 2015, pp. 985–994. https://doi.org/10.1145/2783258.2783370

[19] Shaikh, A. (2020). Guest Editorial: Advances in Deep Learning in Mobile Interactive Algorithms and Learning Technologies. International Journal of Interactive Mobile Technologies (iJIM), vol. 14(10), pp. 4–6. https://doi.org/10.3991/ijim.v14i10.15369

[20] Liu, J., Sarkar, M. K., and Chakraborty, G. (2013). Feature-based Sentiment analysis on android app reviews using SAS® text miner and SAS® sentiment analysis studio. In Proceedings of the SAS Global Forum 2013 Conference, May 2013, (Vol. 250).

[21] Afzaal, M., Usman, M., and Fong, A. (2019). Tourism mobile app with aspect-based sentiment classification framework for tourist reviews. IEEE Transactions on Consumer Electronics, 65: 233–242. https://doi.org/10.1109/TCE.2019.2908944

[22] Areed, S., Alqaryouti, O., Siyam, B., and Shaalan, K. (2020). Aspect-based sentiment analysis for Arabic government reviews. In Recent Advances in NLP: The Case of Arabic Language Springer, Cham. pp. 143–162. https://doi.org/10.1007/978-3-030-34614-0_8

[23] Chakraborty, S., Goyal, P., and Mukherjee, A. (2020). Aspect-based sentiment analysis of scientific reviews. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, August 2020, pp. 207–216 https://doi.org/10.1145/3383583.3398541

[24] Brunova, E., and Bidulya, Y. (2017). Aspect extraction and sentiment analysis in user reviews in Russian about bank service quality. 11th International Conference on Application of Information and Communication Technologies (AICT), September 2017, IEEE, pp. 1–4. https://doi.org/10.1109/ICAICT.2017.8687070

[25] Rao, V., and Hande, K. (2017). A comparative study of static, dynamic and hybrid analysis techniques for android malware detection. International Journal of Engineering Development and Research, 5: 1433–1436.

[26] Koo, W. (2016). Usage of Smartphone Applications: A Descriptive Study of Top 100 US Retailers. International Journal of Interactive Mobile Technologies, 10(3). https://doi.org/10.3991/ijim.v10i3.5827

[27] Dar, M. A., & Parvez, J. (2016). Novel Techniques to Enhance the Security of Smartphone Applications. International Journal of Interactive Mobile Technologies, 10(4). https://doi.org/10.3991/ijim.v10i4.5869

[28] Kong, D., Cen, L., and Jin, H. Autoreb (2015). Automatically understanding the review-to-behavior fidelity in android applications. 22nd ACM SIGSAC Conference on Computer and Communications Security, October 2015, pp. 530–541. https://doi.org/10.1145/2810103.2813689

[29] Shaikh, A., Ali, S., Memon, N., & Karampelas, P. (2010). SOA security aspects in web-based architectural design. In From Sociology to Computing in Social Networks (pp. 415–430). Springer, Vienna. https://doi.org/10.1007/978-3-7091-0294-7_22

[30] Christiana, A., Gyunka, B., & Noah, A. (2020). Android Malware Detection through Machine Learning Techniques: A Review. International Journal of Interactive Mobile Technologies, 16(2). https://doi.org/10.3991/ijoe.v16i02.11549

[31] Hatamian, M., Serna, J., and Rannenberg, K. (2019). Revealing the unrevealed: Mining smartphone users privacy perception on app markets. Computers and Security, 83: 332–353. https://doi.org/10.1016/j.cose.2019.02.010

[32] Weiss, R., Reznik, L., Zhuang, Y., Hoffman, A., Pollard, D., Rafetseder, A., and Cappos, J. (2015). Trust evaluation in mobile devices: An empirical study. In 2015 IEEE Trustcom/BigDataSE/ISPA, August 2015, IEEE, pp. 25–32. https://doi.org/10.1109/Trustcom.2015.353

[33] Abozeid, A., AlHabshy, A. A., and ElDahshan, K. (2021). A Software Security Optimization Architecture (SoSOA) and Its Adaptation for Mobile Applications. International Journal of Interactive Mobile Technologies, 15(11). https://doi.org/10.3991/ijim.v15i11.20133

[34] Quasim, M. T., Khan, M. A., Algarni, F., Alharthy, A., and Alshmrani, G. M. M., (2020), Blockchain Frameworks. In: Khan M., Quasim M., Algarni F., Alharthi A. (eds) Decentralised Internet of Things. Studies in Big Data, vol. 71. Springer, DOI: https://doi.org/10.1007/978-3-030-38677-1_4

[35] Khan, M. A., and Quasim, M. T. (2020), Decentralised IoT, Decenetralised IoT: A Blockchain perspective, Springer, Studies in BigData, 2020.

[36] Shaikh, A., and Alghamdi, A. (2020). IoT, smart environments and interdisciplinary applications for technology management and sustainable development. International Journal of technology management & sustainable development, 19(3). https://doi.org/10.1386/tmsd_00025_2

7 Authors

Abeer Aljumah, Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia.

Amjad Altuwijri, Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia.

Thekra Alsuhaibani, Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia.

Afef Selmi, Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia.

Nada Alruhaily, Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia.

Article submitted 2021-09-23. Resubmitted 2021-10-21. Final acceptance 2021-10-23. Final version published as submitted by the authors.