Turkish Journal of Computer and Mathematics Education *Vol.12 No.3(2021), 4676-4682*

Research Article

HEFESTDROID: Highly Effective Features for Android Malware Detection and Analysis

Shafiu Musa[a], Xiaoqiang Di[b], Hamza Mokhtar[c], Napan Dawurang[d]

[a] Changchun University of Science and Technology (CUST) Jilin province, China

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Rapid globalization and advances in mobile technology have brought about phenomenal attention and great opportunities for android application developers to contribute meaningfully to the global digital market. The android mobile platform being one of the famous mobile operating systems has the highest number of applications in the digital market with a total market share of 76.23% between August 2018 and August 2019, according to a report of global stats counter. However, the substantial number of applications on the platform has led to a great number of malware attacks on the user's privacy and sensitive documents. Consequently, a significant number of malware detection studies have been carried out to reduce the number of malware attacks. This paper analyses the impact of using highly effective android permission features to decipher the problem malware attack. The Highly Effective Features for Android Malware Detection and Analysis (HEFEST) summarises four effective android permission features to be considered in conducting malware detection analysis and classifications. The features recognized in this study are; Normal Declared Permission, Dangerous Permission, Signature-Based Permission, and Signature-or-system. The selection is based on the capabilities of the features in depicting the behaviors of android apps. The research data are drawn from Drebin open source, the dataset comprises 15,036 benign and malicious applications extracted from 215 distinct features, the records 9,026 were malicious and 6,010 benign applications. However, this research compares the detection accuracy of android permission features using machine learning-based algorithms; Support Vector Machine, and K-Nearest Neighbor to achieve a comprehensive accuracy ratio of malware detection, the classifier has a strong accuracy decision of classification and exceptional computational efficiency. The model correctly classified 2,812 out of 2,869 malicious applications appropriately with an accuracy of 98.0% and also classified 1,607 out of 1,642 accurately with a success rate of 97.9%. Generally, 98.0% of classification accuracy was archived.

Keywords:

1. Introduction

The advancement of mobile phones has become one of the most desirable and important aspects of living in this generation. This is due to its capacity to provide wide-ranging services that make life easier. The android has been described as one of the popular platforms in a mobile technology market that has today been battling with serious attacks of malware. According to the Kaspersky security bulletin 2018, 830,135 malwares capable of stealing money via online banking on android devices were blocked. Therefore, malware is one of the critical areas that dominated the usage of android mobile technology for many years.[1]The malware-infected mobile applications compromise the security and privacy of users, by allowing unauthorized access to their privacy-sensitive information, rooting devices, turning devices into remotely controlled bots, etc. [2]. Although it is not an easy task to recognize malicious apps from a benign one because the android market has an open market system whereby applications are not verified by any security agency, this makes the app platform to become a fertileground for attackers to host their repackage and malicious apps to the third-party market. So far, extensive researches have been carried out on how to develop highly effective and accurate tools that can contribute to malware detection and analysis,[3] and[4]although malware developers use several technical methods such as code obfuscation, dynamic execution, repackaging apps and stealth techniques to bypass malware detection. There is therefore an increasing concern that some malwareremained undetected probably as a result of futile features declared or the tactical diversions used by attackers. To prevent such malicious acts, this study outlines effective android permission features to be considered in conducting android malware detection analysis. Our model correctly classified 2,812 out of 2,869 malicious applications with an accuracy of 98.0% and also classified 1,607 out of 1,642 accurately with a success rate of 97.9%. The combined classification accuracy is 98.0% using the Support Vector Machine.[5] discusses SVM as an algorithm with considerable accuracy decision of classification and is very attractive in terms of computational efficiency because of the low computational cost of the algorithms.

2. Overview of Malware Detection Analysis

There are two traditional ways in which malware analysis isperformed; static or dynamic analysis. The static analysis is the process that consists of examining the executable file without viewing the actual instruction, but the

analysis can be proven based on the samples used whether the file is malicious or benign with the help of the available information. Examples are strings, imported/exported libraries, byte sequence, API calls, or other functions. The important aspects of a static approach are that they are cost-effective and less time consuming although it provides inadequate information required in the process of feature extraction. On the other hand, the dynamic analysis provides some highly effective approachestomalware detection by spotting the dynamic behavior of features in apps. e.g. sandbox, and its actual behavior to be captured in the form of API/system calls, or an instruction dump. [6] The dynamic analysis of malware is resistant to most obfuscation techniques and is more active in separating malware families. Generally speaking, the output for both static and dynamic analysis has been widely applied in the machine learning algorithms to cluster samples for demonstrating similar behavior but still, the approaches have been stalled because when using probabilistic or statistical features of the training samples, the approaches were not able to discover the similarity between more than one variants of the same family.

3. Malware Behaviors Analyses

The Android malware is described as the collection of similar malicious applications, which have inherited and derived features of malicious behavior.[7-9]Different authorities describe malware as a dangerous program which interrupts mobile operations, collects sensitive information, and get access to private data. Malware poses a significant threat to mobile devices because, the devices consist of contacts, bank account details, credit/debit numbers, private photos, messages, and lots of other sensitive documents. The research was conducted by classifying application as malicious or benign based on the proposed two independent approaches. Network-based detection and System call-based detection. In network-based detection, an app is classified as malware if it tried to connect to a malicious domain server, and the system-calls based detection is based on identifying malware by analyzing its similarity of system call frequencies to those of known malicious applications.

4. Proposed Framework

The proposed HEFESTDROID presents highly effective and scalable features of android malware detection analysis. The first stage is the feature extraction sequence in which an android app is decompiled and converted into androidManifest.xml files, this process requires tools to be executed, such as smali, backsmali and apktool which is a more effective approach to decompiled android apk. However, some researchers use DEX2Jar methods to extract android source codes from its apk which is a bit complex and time-consuming. The second stage involves behavioral analysis of the extracted apk files to select the highly sensitive and important features as defined in this research that, the semantic artifact to be used are permission feature, monitoring system events, intense API features and action permission rates as the most common features selected to be extracted in this paper. The fourth stage is the classifier model which is a major task whereby the selected features would be trained as a dataset to classify the apps as malicious or benign according to the behavioral act of the apps. Finally, the recommendation report of the apps will be notified through an onscreen alert whether the apps are benign or malicious.

5. Feature selection

The term features selection refers to a process of selecting an attribute from a large file that has been extracted from software applications [10, 11]. In this work, features were selected from an android app that is worth classification. Hence, among a large number of features in the android app, there exist some distinctly redundant features, consumed storage, and increase the application runtime. Consequently, this paper focuses on the highly effective features that provide accuracy and high precision functions in malware analysis

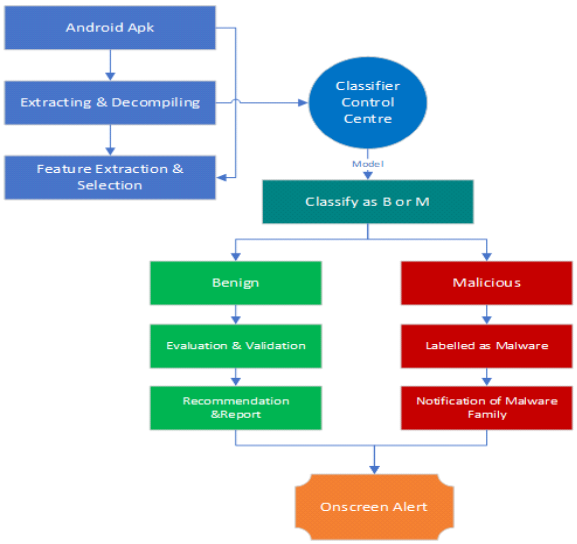

Figure 1: Proposed HEFEST Android Malware Detection framework

6. Types of Features selection

Presently, there are so many considerable feature selection statistical methods[12][10, 13, 14]Three novel methods have been used in this study. Correlation-based- Feature Selection (CFS), Information Gain (IG), and Chi-square CHI² to determine the impact of the feature's selection.Correlation-based feature selection (CFS) arranges attributes based on heuristic evaluation function. The function evaluates subsets made of attribute vectors, which are correlated with the class label, but independent of each other. The CFS method considers redundant features as low correlation with the class and therefore the algorithm disregards their processes. On the other hand, excess features should be examined, as they are usually strongly correlated with one or more of the other attributes. The criterion used to assess a subset of l features can be expressed as follows: where MS is the evaluation of a subset of S consisting of one (1) features, tcf is the average correlation value between features and class labels, and tff is the average correlation value between two features.

7. Permission features

There have been significant empirical researches that pointed out permission features as highly effective features in malware detection analysis, the android declares permission as a set of instructions that aims to protect the privacy of an Android user. A review from [1, 15-17]stated that the Android security model defends permissions features; therefore, the android permission is a restriction limiting access to a part of the code or data on the device. The limitation is imposed to protect critical data and code that could be misused to distort or destruct user's experiences. Android apps must request certain permission to access sensitive user data in Android. For instance, the permission get GPS location which gives the current GPS location of the user, if available, send_sms, this allows users to send SMS, access_ network it also gives out access to the network and many other important permissions. Therefore, in response to approving the requested permissions, the user has explicitly decided and approved the permission to use the app functions. The android permission has two (2) hierarchical models, high-level permission, and low-level permission. The high-level permission is being declared in the framework level i.e. using java programming interface to declare the requested permission and the low-level permissions are also tested in C/C++ instinctive services e.g. record_ audio when creating a socket.

8. Presentation of android permission features

The permission features have been important in android applications as they carry sufficient privileged information about the primary functions of the apps. [18, 19]However, the apps developers are responsible for determining which permission feature to be declared as a sensitive or dangerous request. Therefore, users may read and approve the permission listed in the installation process to grant access to resources files and features, failure to approve might end up canceling the installation process due to android development policy. Firstly, the Normal Declared Permission; is low-risk permission which allows applications to access API calls. Secondly, Dangerous Permission: is high-risk permission that allows apps to access potential harmful API calls. Thirdly, Signature-Based Permission: certificate-based permission in which the system grants automatic approval without

notifying the user or requesting authorization to proceed along as the system certificate becomes similar to the application. Finally, Signature-or-system: is special permissions that have sufficient protection level that developers considered especially in certain situations. The table below indicates the android permissions features and their status.

9. Classification processes and Data Evaluation

Classifying android applications into whether malicious or benign is a primary task of a classifier. However, this research compares the detection accuracy of android permission features using machinelearning-based algorithms; Support Vector Machine and K-Nearest Neighbor to achieve a comprehensive accuracy ratio of malware detection [20] the reason for selecting the two algorithms is because of its capabilities in making accurate labeling decisions. application and exceptional empiricalrealizations of data classification. This is why the study recognizes the SVM test binary classification models. Where the study classified the benign app as one (1) and malicious app as two[21]. The research data in this work are drawn from Drebin open source, the dataset comprises 15,036 benign and malicious applications extracted from 215 distinct features, the records 9,026 were malicious and 6,010 benign applications. Using 70:30 ratios. The work investigates the impact of a permission feature in malware detection analysis and classification. Although extensive research has been carried out on this, no single

Examples of permission features		
1. Normal declared Permission		
S/N	Permission feature	Status
1	android.permission.READ_CONTACTS	Normal
2	android. permission. ACCESS_NETWORK_STATE	Normal
3	android. permission. READ_PHONE_STATE	Normal
2. Dangerous Permission		
S/N	Permission feature	Status
1	android.permission.READ_SMS	Dangerous
2	android.browser.permission.READ_HISTORY_BOOKMARKS	Dangerous
3	android.permission.SUBSCRIBED_FEEDS_READ	Dangerous
4	android.permission.SEND_RESPOND_VIA_MESSAGE	Dangerous
3. Signature Permission		
S/N	Permission feature	Status
1	android.permission.READ_CALENDAR	Signature
2	android.permission.PROCESS_OUTGOING_CALLS	Signature
3	android.permission.SET_TIME	Signature
4. Signature-or-System Permission		
S/N	Permission feature	Status
1	android.permission.BIND_NFC_SERVICE	System
2	android.permission.BIND_ACCESSIBILITY_SERVICE	System
3	android.permission.BLUETOOTH_PRIVILEGED	System

Figure2: Extracted Permission Features

the study discusses the effectiveness of separate permissions as it has a comprehensive role in malware classifications.

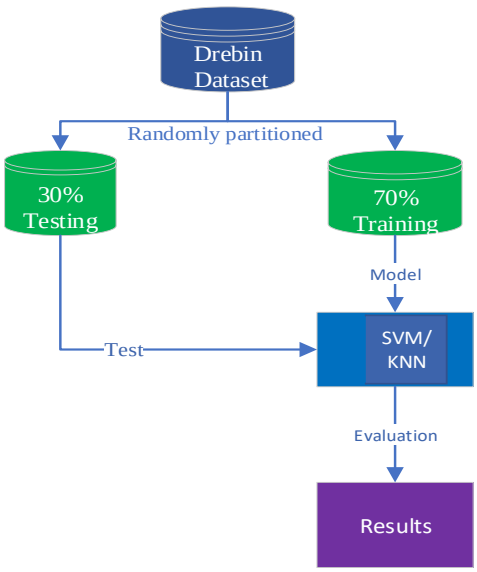

Figure 3: Dataset formulation

10. Related works

There is a large volume of published studies that investigate android malware detection using a machine learning-based algorithm [2, 22, 23]. The studies further described android malware detection as a pervasive area that requiresnoteworthy attention. [24]discusses that, the number of malicious apps and adware on the android platform is growing exponentially on mobile devices. Therefore, the research highlighted the importance of using static, dynamic, and hybrid approaches in malware detection. [25] Developed DroidCat, an app classification approach that leverage systematic profiling and supervised learning, to classify app as Benign or malicious family. The DroidCat considered both benign and malicious apps as inputs. It further computes 70 metrics as behavioral features. The inadequacy of this model is the inability to recognize a single metric as behavior distinct while HEFEST can detect based on a single distinct entity.[26] The architect of a deep autoencoder and convolutional neural network has a significant advantage to train the dataset within a shortperiod compared to other models. It takes advantage of CNN's ability in reducing complexity and time. Having DAE with CNN onboard offers a pre-training method that captures the essential features of android apps efficiently. However, the models hide some important features when if dataset training is taken place[27] Create a Computing Adaptive Feature Weights with PSO to Improve Android Malware Detection, the proposed model adopts support vector machine (SVM) classification model for Android malware detection based on a using IG and PSO feature weights. The feature weights are to replicate with other features to make similar samples more compressed to simply classify between benign and malicious. Secondly, the model signifies between features and class labels, the Information Gain is used not only to select a specified number of features but also to evaluate feature weights. The results indicate that IG weights have small amount of effect on performance. Finally, the model has an adaptive inertia weight process called fitness-based and chaotic adaptive inertia weight-PSO (FCAIW-PSO) for uncomplicated PSO that is created on both the fitness and a chaotic term to progress the particle penetration capability.[15, 28-30] comprehensively demonstrates the importance of android malware detection using various approaches of machine learning techniques

11. Results

We present the result obtained using the Drebin dataset which comprises 15,036 records of benign and malicious applications, attributed with 215 distinct features, among the records 9,026 were malicious and 6,010 benign applications. Using 70:30 ratios. The dataset was randomly partitioned into training and test set respectively. We then used 70% of the dataset 10,525 records and trained the model using SVM and KNN algorithms and tested the model using 30% 4,511 which contains 2,869 malicious and 1,642 benign. The model correctly classified 2,812 out of 2,869 malicious applications which represented 98.0% and also classified 1,607 out of 1,642 accurately with a success rate of 97.9 percent, the combined classification performance is 98.0%. The confusion matrix below detailed the result

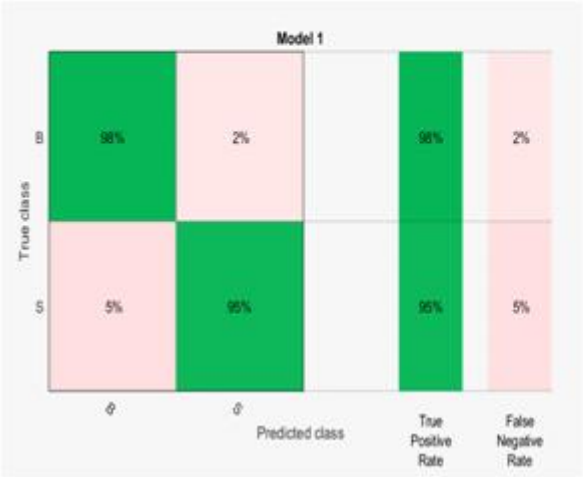

Figure 3: Result

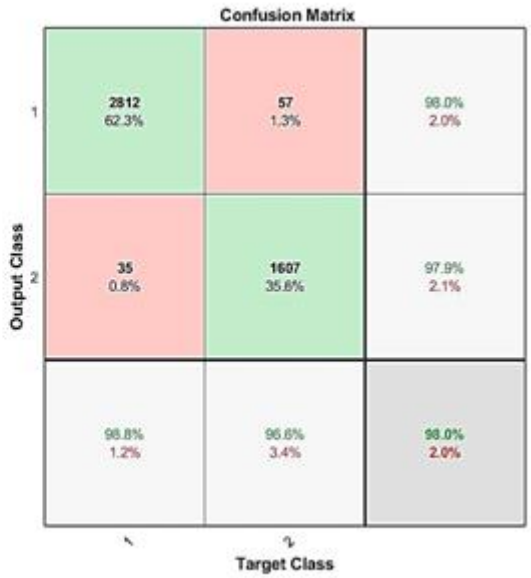

Figure 4: Result

12. Conclusions and recommendations

Despite numerous researches in that area, the malware remains undetected because the researches were carried out with a small number of a dataset and an inefficient number of features which results in disproportion in such

investigations. However, this paper critically investigated the existing android permission functions separately, and its relevance in carrying out a malware analysis and mobile security investigation. The research data in these studies are drawn from Drebin 215 open-source dataset. The dataset was randomly partitioned into training and test set respectively. We then used 70% of the dataset 10,525 records and trained the model using SVM and KNN algorithms and tested the model using 30% 4,511 which contains 2,869 malicious and 1,642 benign. The model correctly classified 2,812 out of 2,869 malicious applications which represented 98.0% and also classified 1,607 out of 1,642 accurately with a success rate of 97.9 percent, the combined classification performance is 98.0%.

References

Wu, S., et al., Effective detection of android malware based on the usage of data flow APIs and machine learning. 2016. 75: p. 17-25.

Ali-Gombe, A., et al. Aspectdroid: Android app analysis system. in Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy. 2016. ACM.

Yang, W., M. Prasad, and T. Xie. Enmobile: Entity-based characterization and analysis of mobile malware. in 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE). 2018. IEEE.

Baskaran, B. and A. Ralescu, A study of android malware detection techniques and machine learning. 2016.

Kruczkowski, M. and E.N. Szynkiewicz. Support vector machine for malware analysis and classification. in Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02. 2014. IEEE Computer Society.

Afonso, V.M., et al., Identifying Android malware using dynamically obtained features. 2015. 11(1): p. 9-17.

Makandar, A., A.J.I.J.o.T.i.C.S. Patrot, and Engineering, Malware image analysis and classification using support vector machine. 2015. 4(5): p. 01-03.

Sahs, J. and L. Khan. A machine learning approach to android malware detection. in 2012 European Intelligence and Security Informatics Conference. 2012. IEEE.

Yerima, S.Y., S. Sezer, and I. Muttik. Android malware detection using parallel machine learning classifiers. in 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies. 2014. IEEE.

Gottwalt, F., et al., CorrCorr: A feature selection method for multivariate correlation network anomaly detection techniques. 2019. 83: p. 234-245.

Hu, L., et al., Feature selection considering two types of feature relevancy and feature interdependency. 2018. 93: p. 423-434.

Jin, J., et al., Correlation-based channel selection and regularized feature optimization for MI-based BCI. 2019. 118: p. 262-270.

Bahassine, S., et al., Feature selection using an improved Chi-square for Arabic text classification. 2020. 32(2): p. 225-231.

Wang, X., et al., Input feature selection method based on feature set equivalence and mutual information gain maximization. 2019. 7: p. 151525-151538.

Martín, A., V. Rodríguez-Fernández, and D.J.E.A.o.A.I. Camacho, CANDYMAN: Classifying Android malware families by modelling dynamic traces with Markov chains. 2018. 74: p. 121-133.

Onwuzurike, L., et al., MaMaDroid: Detecting android malware by building markov chains of behavioral models (extended version). 2019. 22(2): p. 14.

Papadopoulos, H., et al., Android malware detection with unbiased confidence guarantees. 2018. 280: p. 3-12.

Bhattacharya, A., et al., A feature selection technique based on rough set and improvised PSO algorithm (PSORS-FS) for permission based detection of Android malwares. 2019. 10(7): p. 1893-1907.

Fang, Y., et al., Android Malware Familial Classification Based on DEX File Section Features. 2020. 8: p. 10614-10627.

Singh, T., et al., Support vector machines and malware detection. 2016. 12(4): p. 203-212.

Mariconti, E., et al., Mamadroid: Detecting android malware by building markov chains of behavioral models. 2016.

Allen, J., et al. Improving Accuracy of Android Malware Detection with Lightweight Contextual Awareness. in Proceedings of the 34th Annual Computer Security Applications Conference. 2018. ACM.

Damshenas, M., et al., M0droid: An android behavioral-based malware detection model. 2015. 11(3): p. 141-157.

<A Study of Android Malware Detection Techniques and Machine Learning.pdf>.

<DroidCat Effective Android Malware Detection.pdf>.

<Effective android malware detection with a hybrid model based.pdf>.

<Improve Android Malware Detection.pdf>.

Kurniawan, H., Y. Rosmansyah, and B. Dabarsyah. Android anomaly detection system using machine learning classification. in 2015 international conference on electrical engineering and informatics (ICEEI). 2015. IEEE.

Li, J., et al., Significant permission identification for machine-learning-based android malware detection. 2018. 14(7): p. 3216-3225.

Shankar, V.G., et al. AndroTaint: An efficient android malware detection framework using dynamic taint analysis. in 2017 ISEA Asia security and privacy (ISEASP). 2017. IEEE.

First Shafiu Musaholds a Bachelors' Degree in Information Technology (Hons) in Network Technology from Infrastructure University Kuala Lumpur (IUKL) Malaysia, in the year 2012. and Master's Degree in Information Technology (MIT) in 2014 at the same University. After serving Nigeria's mandatory one-year national service at National Information Technology Development Agency (NITDA) which completed in 2015. He later joined the Agency (NITDA) as a full staff working under Corporate Planning and Strategy Department as Senior Scientific Officer. He is currently a PhD Student at Changchun University of Science and Technology, China. He is a research fellow at the Department of Information and Communication Engineering. He is also a certified ISO27001 Information Security Management System, Lead Implementer, Member NITDA software Quality Assurance, Member Project Management Profession and member Cisco Network Academy. Other portfolio held includes Desk Officer – NITDA Budget Committee and three term Technical Officer at Joint Admission and Matriculation Board (JAMB) managing Computer Based Test (CBT) examinations in Nigeria

Second Xiaoqiang DI received B.Sc. degree in Computer Science and Technology in 2002 and Master's Degree in 2007 from ChangchunUniversity of Science and Technology (CUST) XIAOQIANG DI also completed his PhD in Communication and Information System at the same university in the year 2014 respectively.Hewas a visiting scholar at NorwegianUniversity of Science and Technology, Norway,from Aug. 2012 to Aug. 2013. Currently aprofessor and supervisor of Ph.D. students in ChangchunUniversity of Science and Technology (CUST) His major research interestsinclude network information security and integrated network.

Third Hamza Mokhtar received BSc. in Mathematics and Computer Science from Gezira University in 2007. M.Sc. Degree in ComputerScience from the University of Khartoum, SUDAN, in 2012. He is currently pursuing his PhD degree in Computer Science at Changchun University of Science and Technology (CUST). His research interest includesNetwork Management, Communication Software-Defined Network (SDN) and, Integrated Network

Forth Napan Dawurang was awarded B. Tech in Computer Science from Abubakar Tafawa Balewa University Bauchi in 1999. An M.Sc. Degree in Information Engineering at Robert Gordon University Aberdeen, Scotland, in 2005. He has worked with Qatar Petroleum as Systems Engineer2012 - 2015, Petroleum Technology Development Fund (PTDF) 2010 - 2012CSE Controls Ltd Aberdeen UK as Systems Engineer 2007 – 2010, Aquidata Excel JV Ltd Aberdeen UK as Systems Engineer 2005 – 2007,Intercellular Nigeria Limited as Database Administrator, Billing & Customer Accounts Department2002 – 2004, Federal Airports Authority of Nigeria as IT Support Analyst 2000 – 2002. He is currently pursuing his PhD degree in Computer Science at Changchun University of Science and Technology (CUST)