Journal of Electrical Engineering and Modern Technology

Volume No. 11

Issue No. 2

May - August 2023

ENRICHED PUBLICATIONS PVT. LTD

S-9, IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR-5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-47026006

Journal of Electrical Engineering and Modern Technology

Aims and Scope

Journal of Electrical engineering and Modern Technology, publishes original research papers in the fields of Electrical and Electronic Engineering and in related disciplines. Areas included (but not limited to) are electronics and communications engineering, electric energy, automation, control and instrumentation, computer and information technology, and the electrical engineering aspects of building services and aerospace engineering, Journal publishes research articles and reviews within the whole field of electrical and electronic engineering, new teaching methods, curriculum design, assessment, validation and the impact of new technologies and it will continue to provide information on the latest trends and developments in this ever-expanding subject.

Journal of Electrical Engineering and Modern Technology

Journal of Electrical Engineering and Modern Technology

(Volume No. 11, Issue No. 2, May - August 2023)

Contents

Home Automation Using Raspberry PI and Android Application

[1]Dnyaneshwar J Dhangar, [2]Akshay Parekh, [3]Himanshu Patil, [4]Sakshi Patil, [5]Shubham Patil

[1]Assistant Professor, [2,3,4,5]Department of Computer Engineering, Rajiv Gandhi Institute of Technology, Juhu-Versova link road Mumbai-400053, India

E-mail: [1]dnyaneshwar.dhangar@mctrgit.ac.in, [2]akshayparekh63@gmail.com, [3]himanshupatil.97@gmail.com, [4]sakshipatil.siya@gmail.com, [5]patilshubh1996@gmail.com

A B S T R A C T

Operating manually the electrical switches every now and then is a problem for physically challenged and aged people. The proposed system solves the problem by making such a system that can operate electrical switches as well appliances from an Android device using Raspberry pi which acts as a interface. The electrical loads are controlled based on server. This input signal is received from the android device. The android device can be any android based phone or tab having an android OS. The application also provides an effective GUI for providing this functionality. The proposed system consists of a circuit that has lights and fan connected to it along with Wifi module interfaced with raspberry pi. Goal of the proposed system is to find cost efficient and flexible platform to operate, monitor home appliances using Raspberry pi.

Keywords - Smart Home, Internet of Thing, Raspberry Pi, Android App, Wi-Fi.

I. INTRODUCTION

Home automation is used to control any electrical devices in our home or office. Different automation systems are available for Home Automation in market designed differently for different purpose. But there are different issues while designing a Home automation system, such as, the GUI of application should be user friendly, as well the interface so as to make connection simple. Various smart home systems have been developed where the control is through Bluetooth, internet, Android applications and short message service based Bluetooth competence is good and most of current technology gadgets have built-in adaptor that will reduce the system's cost. However, the system limits the control to within Bluetooth range of the environment.

The proposed system working is based on Wi-Fi to remotely control appliances from anywhere using Android application. The desktop PC acts as a server. Installation cost is significantly reduced because of use of Android application. Scope of the work is to develop an efficient system using Raspberry pi as an interface between user and appliances. Live picture can be viewed by android application through wireless camera. The pi is a low cost microcomputer that is able to run on Linux and can give endless extension possibilities.

II. PROPOSED SYSTEM

Every user who is experienced in the existing system may think of a system that may add more flexibility and run with some common applications such as android. The proposed system is designed in such a way to avoid the limitations of the existing system.

The proposed system supports more flexibility, comfort, ability and security. The proposed home automation system is working with very popular android phones. It is having mainly three components; the android enable user device , a Wi-Fi router having a good scalable range, and a Raspberry pi board. Here the users have provision to control the home appliances through application. This will improve the system popularity since there is no need for a wired connection. The instruction from the user will be transmitted through the Wi-Fi network.

The raspberry pi board is configured according to the home system and it will enable the relay circuit as per user request. The relay circuit can control the home appliances also. We can add appliances to the system also add additional security features. The main objective of the proposed system is to design and to implement a cheap and open source home automation system that is capable of controlling and automating most of the house appliances through web page.

The proposed system architectures generally incorporate a Raspberry pi computer for the purposes of network management and provision of remote access. Raspberry pi can be configured according to our home system. The user will communicate Raspberry pi through Wi-Fi network. The system is flexible and scalable, allowing additional home appliances designed by multiple vendors, to be securely and safely added to the home network with the minimum amount of network with the minimum amount of effort.

In the proposed system, the Raspberry pi board is connected ton 5A power supply. Pi board is connected to relay board using female-female jumper wires and GPIO pins of Pi are noted accordingly. Different number of relay board are available based on the number of appliances that user wants to connect. Eachchannel of relay board is connected to different appliances which are controlled by an individual GPIO pin.

Application is connected to the server that the PC is running through internet. Pi has to be connected to the same server for commands from application to be effective.

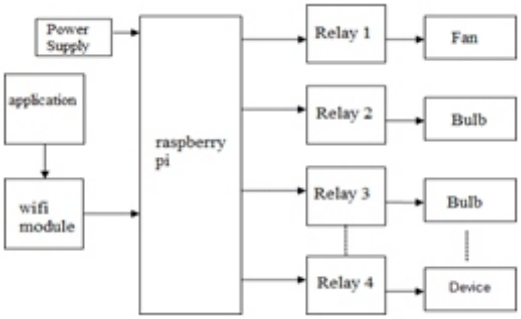

Figure 1: Block Diagram

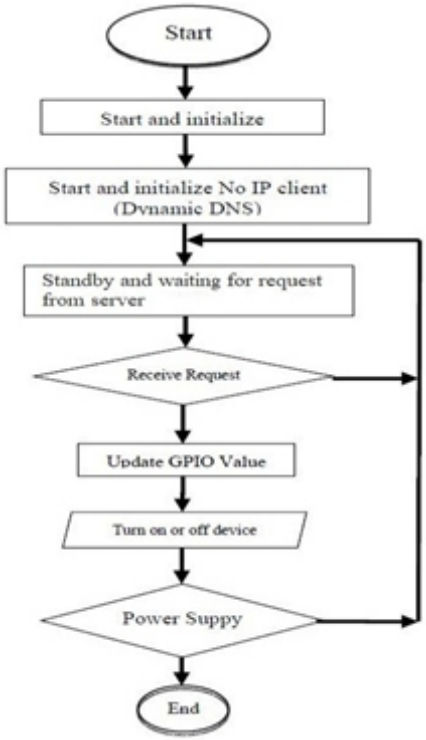

Figure 2: Flow Diagram

III. HARDWARE AND SOFTWARE REQUIREMENTS

A) Hardware Requirements

1) Raspberry pi(model 3b+):

The Raspberry Pi is a compact and smart computer that has been developed to provide low-cost computer and free software to students. Raspberry Pi has the availability of Linux software stack which easier to connect to the internet and variety of programming languages can be used. But, accessing hardware is not a real-time which interfacing with the hardware can be deferred if the CPU is busy.

2) Relay board (5v):

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically

operate a switch, but other operating principles are also used, such as solid-state relays. A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor.

B) Software Requirements

1) Android Applications:

Windows Mobile, Symbion, IOS and Android are several platforms that are used to develop smart phone application. In this work, we are using an Android platform as its main smart phone application platform, as it has a huge market and cost effective due to open source. Besides that, Android development application tools are free to download and provide flexibility to developers to extend or easily edit the source code. Android Software Development Kit (SDK) used a Java as its main language which provide tools and APIs to be used in developing the smart home application. The SDK provides a complete set of development tools such as debugger, libraries and a handset emulator. Eclipse is the officially supported integrated development environment (IDE) has been used on in conjunction with the Android Development Tools (ADT) Plug-in to develop the application. The accessory mode of the developed Android application is a feature of Android OS since version 2.3.3 (API 10) AND Kit Kat 4.4.2 (API 19).

The application screen consists of functions like light and fan controlling which user can select any function which he or she wish to control. The designed application for the smart home system provides following functionalities such as remote connection through Wi-Fi or mobile network to the raspberry pi and device control to the user. In the development of application, standard GET & POST request operations have been utilized to access the devices +through application remotely. Besides that, the android application (client) continuously access the web server to update the interface and send HTTP requests to server each time the user want to access a specific function.

IV. CONCLUSION

The system as the name indicates, "Home automation using Raspberry pi and Android application" makes the system more efficient, reliable and provides attractive user interface compared to other home automation systems. In this system mobile devices are integrated into home automation systems. A novel architecture for a home automation system is proposed using the relatively new communication technologies. The system consists of mainly three components is a Wi-Fi module, raspberry pi board and relay circuits. Wi-Fi is used as the communication channel between android phone and the raspberry pi board. We hide the complexity of notions involved in the home automation system by including them into a simple, but comprehensive set of related concepts. This simplification is needed to fit as much of the functionality on the limited space offered by a mobile.

ACKNOWLEDGMENT

The authors would like to thank the Prof D. J. Dhangar of Rajiv Gandhi Institute of Technology for the guidance, encouragement, support.

REFERENCES

[1] ElShafee and K. A. Hamed, "Design and Implementation of a Wi-Fi Based Home Automation Raspberry Pi Technical documentations from elinux.org
[2] https://circuitdigest.com/microcontroller-projects/iot- raspberry-pi-home-automation
[3] ElShafee and K. A. Hamed, "Design and Implementation of a Wi-Fi Based Home Automation System, "World Academy of Science, Engineering and Technology, vol. 68, pp. 2177- 2180, 2012. Dhami Himani Singh et al; International Journal of Advance Research, Ideas and Innovations in Technology. © 2017, www.IJARIIT.com All Rights Reserved Page | 525
[4] Magazine for Raspberry Pi users "The MagPi ".
[5] Raspberry Pi latest kit from raspberrypi.org.
[6] Android application development references from developer.android.com
[7] PuTTy a free telnet/ssh client from putty.org.

Development of Highly Secured and Improved Captcha System By Using Dynamic Word Cloud Security

[1]Divya Suvarna, [2]Sujata Pathak
[1,2]KJ. Somaiya College of Engineering, Mumbai, India
E-mail: [1]divya.suvarna@somaiya.edu, [2]sujatapathak@somaiya.edu

A B S T R A C T

Internet is a plethora of information. Today millions of people are using website to access such valuable information. Due to cyber-crime website owner should learn to protect their assets from attacker. If the attacker manages to hack the website and its users account then that could be a threat. In order to prevent cyber- attack, website can be integrated with security mechanism such as Completely Automated Turing Test to tell computers and humans apart. But unfortunately it has been proven weak and attackers have managed to bypass the system. By using Machine Learning attacker can create a bot that can mimic human and break the system. This paper presents a replacement for such weak system. A system that is capable of preventing Denial of Service attack was proposed. Dynamic Word Cloud is integrated in place of text-based CAPTCHA successfully to increase the security level of the website and has proven to be a robust solution. Also solutions are provided to perform security testing on it so that one can prevent from future automated bot attacks.

Keywords - CAPTCHA, Cyber Attack, Word Cloud, Machine Learning, Automated bots.

I. INTRODUCTION

Many websites, e-banking services, online tolling and digitized books uses CAPTCHA technology to determine whether or not the user is human. Such security mechanism is coined crucial to the success of website. But unfortunately this mechanism can be broken by the attacker due to its design flaw. The importance of studying the breaking of Text-based CAPTCHA is vital because websites using such system are under a greater threat. The evolution of Turing Test has led to many variations in it, such as image, video, audio, and No CAPTCHA based system. Still there are many websites using vulnerable Text-based CAPTCHA system in order to authenticate user.

In past few years, the battle to secure the CAPTCHA system has been broken by using sophisticated attack by bypassing the authentication flow. It is no more considered to be secured. After studying the attack pattern, the threats associated with the systems vary from authentication bypass, fake account creation to abuse of functionality. Investigation of attack scenarios on such weak implemented system gave benefit of doubt whether to use it or not as a protection from online spammers and bots. Problems associated with the existing text based CAPTCHA system can get the website owner into huge trouble hence an alternative mechanism need to be introduced in order to replace the text based security system.

The need for investigation of CAPTCHA system can minimize the literature gap as the existing research work is limited and still at the improvising stage. As the advanced system are more secured but they have originated from the base model which is text based CAPTCHA. Therefore there is a high possibility of compromising video, audio, and No CAPTCHA system.

The research work aims at replacing the current vulnerable system with a different concept but with more advanced security mechanism. Only the text based system will be replaced by another technology while the other systems are quite stable and able to prevent cyber-attacks on websites. But soon those will also be defeated by advanced attacks by using Machine Learning Algorithm. The websites using such vulnerable systems are GATE exam site, Net Banking service, registration online and many more. Hence There is a need to replace weak system with a better one.

In this paper, Chapter 1 gives the brief introduction about a topic to understand the concept behind it. In Chapter 2, literature survey is provided to support the work and idea proposed in the paper. Chapter 3 is methodology part where methods and algorithm is provided for new Security Model. İn Chapter 4, results are successfully documented on following the methods properly. Lastly the paper provides with a conclusion and references.

II. LITERATURE SURVEY

Word cloud is an optic representation of text with features like font size and color. The position of it in the cloud can be arbitrary or reflect its relation to other word. In this paper [1], author Rosa Tsegaye et al. 2015 presented a tool that generates text cloud from German company website. The main idea is to anticipate the overall work of the companies details in one word cloud based solely on their own web page.

In the paper [2], author Z. He et al. 2017 proposed a concept of Molecular Cloud (MC) from stars and galaxies. The two -stage MC based model portray word cloud into a galaxy-like cloud. The model presents an algorithm, 1st stage that stimulates the moving and colliding and 2nd stage with merging of collapsing of it. In addition, it consume long range force based disturbance from star formation. Their experiment suggests that it produce galaxy-like word cloud for users to choose from.

The properties of Class Feature Centroid (CFC) classifier by considering of rate of change of each prototype vector with respect to individual dimension [3]. There are many limitations and to overcome the author T.T Nguyen et al. 2011 [3], proposed an improved and robust centroid –based classifier that uses precise term class distribution properties instead of simple presence or absence of terms over classes in text categorization.

In paper [4], author W. Cui et al. 2010, presented a Context-Preserving Visualization for users to visually analyze a collection of document. It has two components; first component is a trend chart and second is set of word cloud distributed over time. They tried to compare their own layout with the existing layout for better document visualization with timestamp, as semantically coherent words are grouped together to ensure dimensional stability over time [4].

The importance of automated cloud extraction from satellite imagery is quite unavoidable. In the study of automatic cloud detection method author Y. Yuan et 2015, proposed an object grouping of image features. The support vector machine algorithm is applied to differentiate cloud and non-cloud regions [6]. Thereafter, the "Grab-Cut algorithm" is utilized to extract accurate cloud regions. The key of this method is to deal with the highly varying patterns of clouds. The Bag-of-Words (BOW) model is used to construct the compact feature vectors from densely extracted local features, such as dense "scale-invariant feature transform" (SIFT) [6].

Last fm is a music website that provides a multivariate dataset. However, it is diffcult to examine relationships if using purely Last fm APIs and its Web services. In this paper [7], author Dinh Quyen Nguyen et 2016, presented a visualization technique, called "Hyper Word Clouds" to examine such complex and multi-relationship dataset. Through the text-based depiction of Last.fm data items are visualized as words linked in parallel and anchor based word clouds. The users can mutually select to filter, highlight, and compare data and relationships of interest to discover further insights.

An envision method is presented by author P. Fabo et 2012, that centered around the discussion, its dynamics, intensity and topic changes [9]. The flow is divided into uniform time spans that collect the data and runs the text analyzer. Later the discussion is visualized using details of three different levels. The global view of the discussion development over time, an inside view on discussion on main topics, and display individual word clouds respectively. The method is demonstrated on data from an Internet Relay Chat (IRC) but it can be used for visual analysis of dependent text data such as email communication, search terms, keywords and similar.

The author Quim Castellà et 2014, introduced the concept of word storms to analyze mass document [11]. It is a group of word cloud in which each cloud represents a single document to allow the viewer to compare and contrast the documents. They also presented an algorithm that creates a synchronized word storm, in which word that appears in multiple document are placed in the same location, using the same color and orientation across clouds. This ensures that similar documents are represented by similar looking word cloud, making them easier to compare and contrast visually. The algorithm is evaluated using an automatic method based on document classification, and a user study. The result shows that a synchronized word storm allows for better visual comparison of documents

In paper [12], author Martin Seyfert et 2017, used an unexplored method to visualize dynamic time varying data field using word cloud. The goal was to present a novel way of generating animated word clouds to show changes in word frequency via font size. The existed method is not quite robust hence a compact layout is generated via Wordle tool using web technology.

The authors C. Binucci et al. 2016, explains that words in semantic word cloud reflect logical correlation. With a new dynamic scenario an algorithm has been proposed in the literature section to compute both static and dynamic semantic word clouds. In this paper [13], they assumed to receive a streaming text in a given time period with real time computation, also a dynamic word cloud that shows the evolution of the stream. It preserves both the logical relationships between words and the user's mental map. The author presents an algorithmic framework and the results of an experimental study.

Author Benjamin Renoust et al. 2017 [14], word cloud is a popular means for outlining text documents. The visualization is usually done by the word frequencies from single text sources or attributes. However, such visualization of several text documents in one word cloud has rarely been addressed so far. This paper presents RadCloud, a technique for text visualization based on multiple word clouds merged into a single view. The text sources are indicated by stored bar charts and the descriptive word arrangement and.

In paper [15], Multimedia data Text analysis along with computer vision combine to form two levels of extracted information. Another approach is the Face cloud. Multiplex networks can seize and combine both semantics sources. Inclined by word clouds, they allow to grasp visual text semantics information from multimedia collection all at once. Author M. Burch et al. 2014 also demonstrate their system with the inspection of a Japanese news archive.

The author Jitendra Ajmera et al. 2012, presented an apparatus to create and supply an audio cloud for audio content. Such apparatus are expected to provide a summary of the audio documents. They have wide apposite in various domains, especially for users who do not use the internet but interact with audio-based systems [16]. Detecting words from an audio content is challenging, especially if the audio is in languages for which a speech recognition system does not exist. They also give a language independent system to detect repeated words within an audio document. Authors presented four ways to supply these words that form an audio cloud.

The Author Yuan Zhang et 2010, research work on the Visual Feature of word cloud Chinese relied on user's browsing experience showed a phenomenal analysis by using eight visual features. In this paper [17], they look over the visual elements of Chinese word clouds, and compared the structure of English

and Chinese language. 37 students participated in the experiments with 40 tag clouds. The results showed that different features such as “number of strokes, reduplication or not, number of characters, font weight, and position of tag clouds” all have different impact on user's browsing behavior.

In this paper [18], author Chun Che Fung et 2011, proposed a new approach of Web Content extraction based on discovery rules and external path utility in XML. The motive was to address the issue associated with the Web evoke by generation tag cloud from Thai websites to provide an overview of the keywords in the Web Pages. Three major steps followed by the authors are Web Page portion and feature extraction, block detection, and content extraction selection. The result shows that by improvising the noise filtering and block detection one can extract the web content with accuracy of over 96% from Thai Web site.

Author Preeti Mundada et 2012, Tag clouds are collection of text representation in cluster form with varying size and color. A very efficient visualization is needed that can be achieved from tag cloud. It is useful for navigation and also retrieval of data. In this paper, a dynamic model of 2nd generation of tag cloud is presented. It is categorized in three different modules such as search criteria, time frame and location based.

III. METHODOLOGY

A. General Description

Today website has become a common platform to access variety of information. This Information can be categorized as Personal information, Healthcare Information, and Payment Card Information. From banking, e-commerce and institution, everyone has a dedicated website. Information present on the website is stored on a secured database. With the increase percentage of cybercrime every website owner should protect their own data.

In order to protect data from malicious attacker, one should keep security measures in place. In case of website authentication, authorization, confidentiality and integrity is maintained by implementing security features such as one time password, CAPTCHAs and many more. In this paper, website using weak Text- based CAPTCHA is vulnerable to many cyber-attacks like Denial of Service, spammer e-mail, code injection, username and password based attacks. It has been proven that such weak CAPTCHAs can be broken using Machine Learning. Machines are becoming smarter and can be a threat to website implementing such weak system, yet advance CAPTCHAs emerged from weak CAPTCHAs it can be possible to break security system. An alternative to such weak system is Tag

Cloud system. Tag Cloud is nothing but a combination of alpha numeric special character and can be of any shapes and size. With its inbuilt capabilities one can implement it in website in order to security from online bots.

Dynamic Word Cloud is an optic representation of text data, using metadata as keywords on website and its generation is dynamic. There are three major types of Word Cloud application in social software, distinguished by their meaning rather than appearance. Word for the frequency of each item, Global Word Cloud where the frequencies are aggregated over all items and users and in third type, the cloud contains categories with size indicating number of subcategories.

B. Algorithm for Word Cloud Model

This paper proposed an algorithm for Word Cloud System Model. The following steps are given below:

STEP 1: Initialize the code. [Start]

STEP 2: User Input as Number of Words.

STEP 3: Then Initialize the questions.

STEP 4: Make a Word Cloud with the provided Words.

STEP 5: Pick one Random Question from the Question Array.

STEP 6: User Input answer based on the selected question.

STEP 7: If the input is correct then the Word Cloud is solved successfully.

STEP 8: User Authentication approved.

STEP 9: If the input is incorrect then the Word Cloud is failed.

STEP 10: User Authentication Failed [End]

The below given Flowchart is for the above proposed Word Cloud System Algorithm. This Model can be used in Websites to increase the level of security. The flowchart explains the method followed during our implementation.

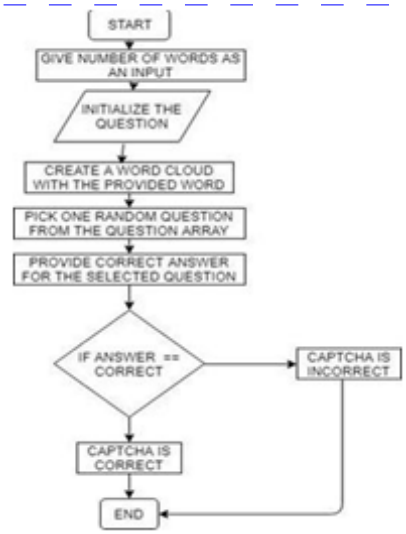

Figure 1: Flowchart for Word Cloud System Algorithm

C. Word Cloud System Design

The formation of the Word Cloud system is divided into two parts. The design flow is presented in this report. A Word Cloud system is nothing but bag of words generated randomly. Now the randomness of the word provides different form of Word Cloud.

This makes the system much better than the text based CAPTCHA generator because the Word Cloud generator is not limited while the CAPTCHA is. Therefore the 1st part of the Word Cloud is better than the Text Based CAPTCHA system.

The 2nd part is quite challenging. It consists of questionnaire where array of questions will be given and the user needs to select the appropriate answer from the 1st part of the Word Cloud. This is how a Word Cloud system comprises of two part where both the parts are crucial and need to be implemented with care.

Figure 2: Word Cloud Modules

In the above figure a Word Cloud Module is dissected into three parts. One in which the text formation will take place by the provided words, another part there will be array of question and in third part the time bound limit is implemented to control wrong input. This entire module will be presented to the users at the monitor who are using the website.

Along with the proposed secured system the system consist of a time bound feature. This feature allows preventing Denial of Service attack. The introduced feature is not present in existing Text-Based CAPTCHA system.

IV. RESULTS

The purpose of implementing a Word Cloud in website was to replace the existing weak text-based CAPTCHA system. Initially website using Text- based CAPTCHA was susceptible to various attacks carried out by using Machine Learning. It has become very easy to train the bot to crack online CAPTCHAs. Hence various advanced CAPTCHA was introduced but they all are derived from the base model, which is considered to be weak.

Therefore this paper proposed Word Cloud System to replace the Text-based CAPTCHA system. Word Cloud is a robust solution because it consists of many elements such as dynamic word formation, available in different shades of colors, sizes and combination of small and capital letters. With a motive to help website owner protect their data from getting it into the wrong hands, a secured robust solution is provided.

The implementation of the thesis work is given below. The results were recorded and the screenshot provides detailed analysis of the same. Steps are as followed

Implementation of the Word Cloud Security System

Step 1: Import the WordCloud from the workspace index.jsp in eclipse

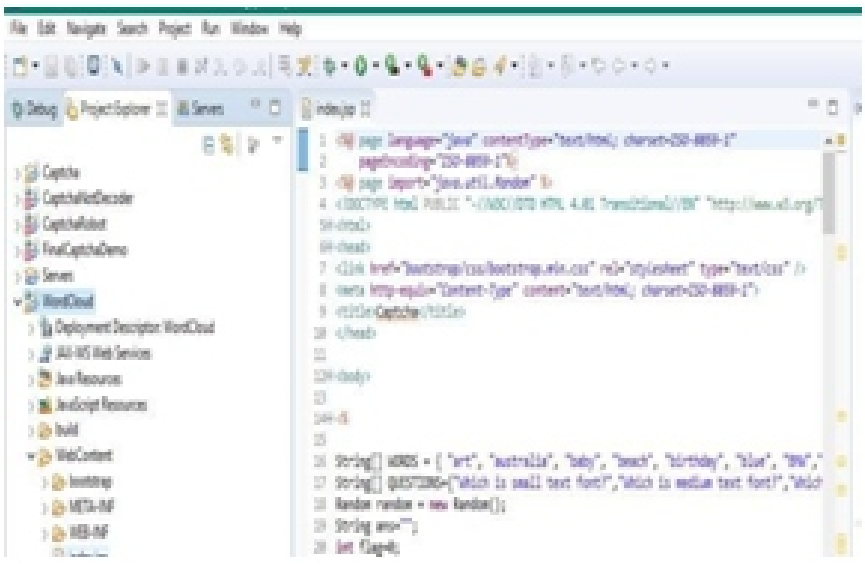

Figure 3: Import the Project in the Eclipse application and load the index.jsp file

In the above figure, once the launcher is ready then import the project file and load the index.jsp file from the workspace. This file contain logic for the thesis.

Step 2: Run the application in the console

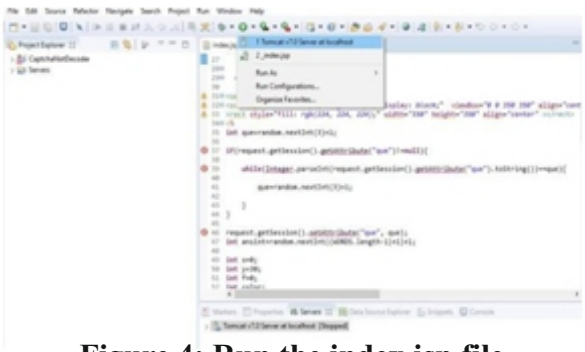

Figure 4: Run the index.jsp file

In the above figure, the index.jsp file is made to run by right clicking the run green arrow at the top of the panel.

Step 3: The local host will fire up in the browser

Figure 5: Local host display the application locally

In the above figure, the link http://localhost:8080/WordCloud/index.jsp will run in the URL and automatically get the Word Cloud System running on your machine locally. On can do the testing of the application before deploying it in the actual environment.

Step 4: The Word Cloud system will be displayed

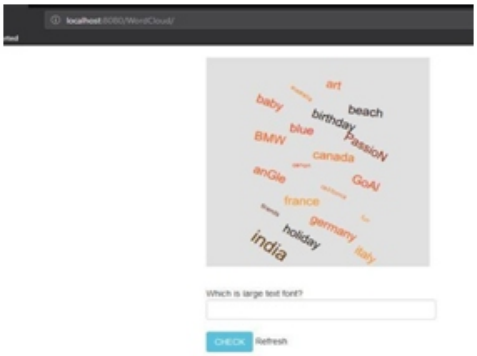

Figure 6: The Word Cloud System Display

In the above figure, the Word Cloud will display as soon as the user opens the Webpage. The application can be used in various domains as per the requirement. The Word Cloud can display both Word generator and question randomly on the screen.

Step 5: The Word Cloud System: Randomly generate a Word Cloud with a question for User 1

Figure 7: Example for the Word Cloud System for User 1

The above figure explains that with same question displayed on the screen will give users different Word Cloud options. The User 1 is provided with a Word Cloud challenge. And it is expected to give correct input. This allows dynamic use of word and gives multiple arrays of option. In case of Text-Based CAPTCHA the generation of it was limited.

Step 6: The Word Cloud system: On every user provided input 1

In the below given figure, a Dynamic Word Cloud System is provided for user. An input is expected from user. In input 1 user need to answer the following question correctly by entering the answer in the check box.

Figure 8: Dynamic Word Cloud System taking Input 1 from User

Step 7: The Word Cloud system: result for input 1

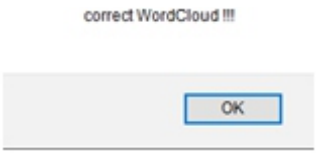

Figure 9: Correct User Input 1 for Dynamic Word Cloud System

The above figure, it shows a dialog box of correct WordCloud!! on every correct answer from user 1.

Step 8: The Word Cloud system: Randomly generate another Word Cloud with a question for User 2

Figure 10: Example for the Word Cloud System for User 2

The above figure explains that with different question displayed on the screen will give users different Word Cloud options. The User 2 is provided with a Word Cloud challenge. And it is expected to give correct input. This allows dynamic use of word and gives multiple arrays of option. In case of Text-Based CAPTCHA the generation of it was limited.

Step 9: The Word Cloud system: On every user provided input 2

Figure 11: Dynamic Word Cloud System taking Input 2 from User

In above figure, a Dynamic Tag Cloud System is provided for user. An input is expected from user. In input 2 user need to answer the following question correctly by entering the answer in the check box. But if incorrect answer is provided then the Tag Cloud system will not work.

Step 10: The Word Cloud system: result for input 2

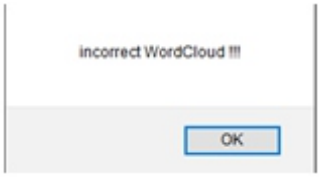

Figure 12: Incorrect User Input 2 for Dynamic Word Cloud System

The above figure, with every invalid answer from user it shows a dialog box of an incorrect WordCloud!! (Tag Cloud)

Step 11: The Word Cloud system: On Refresh

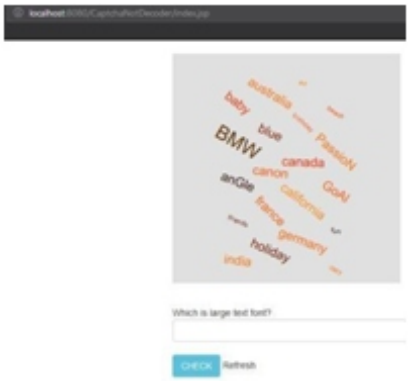

Figure 13: Refresh option

The above figure shows that the dynamic Word Cloud will give you dynamic two staged input that is 1st to solve Word Cloud and 2nd to solve the question. On every refresh option users can refresh to get new challenges.

Step 12: Time Limit Imposed

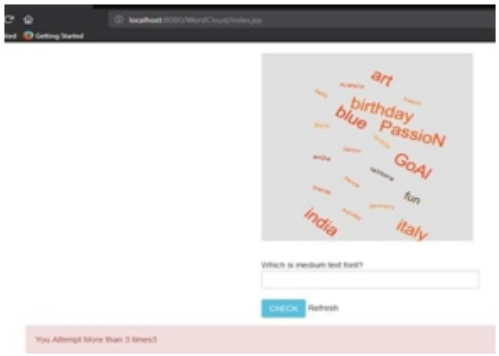

Figure 14: You Attempted more than 3 times

The above figure explains that if a user attempt more than three wrong input then the webpage will throw an error message to the user. And block the access. This is how a secured implementation of the Word Cloud system was achieved successfully. This feature is not available in the Text-Based CAPTCHA system.

In this chapter, the results were recorded after implementing the methodology presented. This not only explains how one can follow the steps but also provide a new way of implementing security mechanism to help user keep safe online. The automated attacks and threat from Machine Learning can be prevented. But research work is not limited till here. This paper provide the basic prototype for the Word Cloud System with time bound limit feature. This need to be upgraded with new features in order to overcome more advanced attacks in future.

V. CONCLUSION

This paper presents the research work on The Dynamic Word Cloud System integration within the website. The motive behind implementing such system was to increase the security level and to replace the weak text-based CAPTCHA system. The intention was to help website owner realize the importance of securing the information from the attacker. The Word Cloud System is a great way to minimize attack vectors, as it is capable of preventing Denial of Service attack. The challenges faced during the implementation phase were solved successfully and aimed to keep improvising the system. Not just limited to this, solutions are provided to every website owner, that they should carry a Threat Modeling approach or a Cyber Kill Chain in order to identify the weaknesses and attack vectors. By this approach website owner can be one step ahead of the attacker. Therefore this paper aimed to achieve a better alternative for weak Text-Based CAPTCHA and provide security approach to keep information safe online. Hence the Dynamic Word Cloud System was proven to be the robust solution with security features.

REFERENCES

[1] Rosa Tsegaye Aga and Christian Wartena. 2015. Constructing concept clouds from company websites. In Proceedings of the 15th International Conference on Knowledge Technologies and Data- driven Business (i- KNOW '15) - ACM, New York and USA.

[2] Z. He, Y. Cao and H. Xiong, "Generate Galaxy-Like Word Cloud Using Molecular Cloud Evolution", 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2017.

[3] T. T. Nguyen, K. Chang and S. C. Hui, "Word Cloud Model for Text Categorization," 2011 IEEE 11th International Conference on Data Mining, Vancouver,BC, 2011.

[4] W. Cui, Y. Wu, S. Liu, F. Wei, M. Zhou and H. Qu, "Context- Preserving, Dynamic Word Cloud Visualization" in IEEE Computer Graphics and Applications, vol. 30, Nov.- Dec. 2010.

[5] P. Wei, T. Xu, X. Qin and C. Wang, "Visualization of Police Intelligence Data Based on Word Clouds," 2014 Tenth International Conference on Computational Intelligence and Security, Kunming, 2014, pp. 539-543.

[6] Y. Yuan and X. Hu, "Bag-of-Words and Object-Based Classification for Cloud Extraction from Satellite Imagery", in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, Aug. 2015.
[7] Dinh Quyen Nguyen and Dinh Dam Le. 2016. Hyper Word Clouds: A Visualization Technique for Last.fm Data and Relationships Examination. In Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication (IMCOM '16). ACM, New York, USA.
[8] A. N. Khan, A. Muhammad and A. M. M. Enriquez, "Mining for Norms in Clouds: Complying to Ethical Communication through Cloud Text Data Mining," 2012 IEEE Fifth International Conference on Utility and Cloud Computing, Chicago, IL,2012.
[9] P. Fabo and M. Novotný, "Three-level Visualization of Internet Discussion with Extruded Word Clouds" 2012 16th International Conference on Information Visualization, Montpellier - 2012.
[10] Ji Wang, Kyle D. Dent, and Chris L. North 2013. Fisheye word cloud for temporal sentiment exploration. In CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13). ACM, New York, USA.
[11] Quim Castellà and Charles Sutton 2014, "Word storms: multiples of word clouds for visual comparison of documents", In Proceedings of the 23rd international conference on World Wide Web (WWW '14). ACM - New York, USA.
[12] Martin Seyfert and Ivan Viola 2017, " Dynamic word clouds". In Proceedings of the 33rd Spring Conference on Computer Graphics (SCCG '17), Stephen N. Spencer (Ed.), ACM. New York, USA.
[13] C. Binucci, W. Didimo and E. Spataro, "Fully dynamic semantic word clouds," 2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA), Chalkidiki, 2016.
[14] Benjamin Renoust, Haolin Ren, Guy Melançon, Marie-Luce Viaud, and Shin'ichi Satoh, 2017. "FaceCloud: Heterogeneous Cloud Visualization of Multiplex Networks for Multimedia Archive Exploration". In Proceedings of the 2017 ACM on Multimedia Conference (MM '17), New York, NY, USA.
[15] M. Burch, S. Lohmann, F. Beck, N. Rodriguez, L. D. Silvestro and D. Weiskopf, "RadCloud: Visualizing Multiple Texts with Merged Word Clouds," 2014 18th International Conference on Information Visualisation, Paris, 2014.
[16] Jitendra Ajmera, Om D Deshmukh, Anupam Jain, Amit Anil Nanavati, Nitendra Rajput, and Saurabh Srivastava. 2012. Audio cloud: creation and rendering. In Proceedings of the 2012 ACM international conference on Intelligent User Interfaces (IUI '12). ACM, New York, NY, USA.
[17] Yuan Zhang and Yihong Rong, "Research on the visual features of Chinese tag cloud based on learners' visual recognition," 2010 International Conference on Artificial Intelligence and Education (ICAIE), Hangzhou, 2010, doi: 10.1109/ICAIE.2010.
[18] W. Thanadechteemapat and C. C. Fung, "Automatic Web Content Extraction for Generating Tag Clouds from Thai Web Sites," 2011 IEEE 8th International Conference on e- Business Engineering, Beijing, 2011. doi: 10.1109/ICEBE.2011.
[19] P. Mundada and A. Ghotkar, "An approach to second generation tag cloud for assessment of business search," 2012 IEEE International Conference on Technology Enhanced Education (ICTEE), Kerala, 2012. doi: 10.1109/ICTEE.2012.
[20] J. Emerson, N. Churcher and C. Deaker, "From Toy to Tool: Extending Tag Clouds for Software and Information Visualisation," 2013 22nd Australian Software Engineering Conference, Melbourne, VIC, 2013. doi: 10.1109/ASWEC.2013.
[21] Felix, Cristian, Steven Franconeri, and Enrico Bertini. "Taking word clouds apart: An empirical investigation of the design space for keyword summaries. " IEEE transactions on visualization and computer graphics" 24, no. 1 (2018).
[22] Sheehan, Shane, Masood Masoodian, and Saturnino Luz. "COMFRE: a visualization for comparing word frequencies in linguistic tasks." In Proceedings of the 2018 International Conference on Advanced Visual Interfaces" , p. 42. ACM, 2018.
[23] Wang, Yunhai, Xiaowei Chu, Chen Bao, Lifeng Zhu, Oliver Deussen, Baoquan Chen, and Michael Sedlmair. "Edwordle: Consistency-preserving word cloud editing." IEEE transactions on visualization and computer graphics 24, no. 1 (2018)

Evolution of Android Malware Offense and Android Ecosystem Defense

[1]Nishant Pandit, [2]Deepti V Vidyarthi

[1]Student, Defence Institute of Advanced Technology, Girinagar, Pune, Maharashtra – 411025, India

[2]Assistant Professor, Defence Institute of Advanced Technology, Girinagar, Pune, Maharashtra – 411025, India

E-mail: [1]nickdiamond1234@gmail.com, [2]deepti.vidyarthi@gmail.com

ABSTRACT

Android mobile devices are used more and more in everyday life. They are our cameras, wallets, and keys. Basically, they embed most of our private information in our pocket. The paper captures the journey of android malware from being mere revenue generation incentives for the malware developer to stealing the personal data of people using these devices. It also discusses how the open source nature of Android has led to fragmentation of the core Operating System among various device manufacturer which introduces additional vulnerabilities. The non-availability of official Google Play store in some countries led to the emergence of various third party Application market which are largely unregulated in terms of the application verification. Android Operating system itself has come a long way in terms of the security features and fixed vulnerabilities over the course of a decade. Our evaluation shows that the Android System has become quite robust against malware threats and automatic installation of malware is not possible without user intervention. We explore certain simple settings on android which would protect the user from malware threats on Android device.

Keywords - Android System, Malware Analysis, Vulnerabilities, Android Fragmentation.

I. INTRODUCTION

Android is a fast growing mobile platform which powers millions of mobile devices. Based on the Linux kernel, android operating system is open and flexible enough to. run on different mobile devices having different hardware configuration [1]. This has increased the popularity and acceptance of android among the users. While Google code change will not be made available to public immediately, new version release still includes most of what community needs: factory images, source code, Over the Air (OTA) distribution channels, and Application Program Interface (API). Openness is in-arguably the main reason why Android OS gains its popularity quickly. The mobile device provides ubiquitous access to internet through WiFi and 3G/4G network which makes it a very attractive target for attackers. The GPS sensors provide accurate location of the user, websites visited on the device becomes a source of advertisement and there are organizations willing to pay for these information about users.

Android is considered as a very complex ecosystem; each device is a composition of different software (open source and closed source), different hardware (screen sizes, manufacturers, proprietary

hardware), and different distributors. This makes the exploit universally impossible but also makes the job of auditing difficult as reviewing every device and their software is a huge amount of work. The process of a security update for a specific Android device can be summarized as follows: (1) security flaw found in an Operating System (OS), (2) Google release the patch, and (3) Original Equipment Manufacturer (OEM) adopting the patch and including it in their custom built. As shown by Thomas, Daniel R et al, [18] if a security vulnerability is fixed through the release of a particular API version it will be 1230 days (3.36 years) after that until the fix is fully deployed.

Most of the malware are distributed via third party Android App store and various websites which offer the cracked version of many paid Applications otherwise available in Google Play Store. Three common ways the malware is distributed to android devices:(1) Repackaging - This technique capitalizes on adding malicious code into benign apps and replacing it android market with similar name. This way, users download the app and get infected by the malware content repackaged in the app. (2) Updation

The an infected version of update is push on the device through social engineering, leading the user to download and install an update through website (other than android market) (3) Drive-by Downloads Social engineering is used to make users download and install apps while browsing through the web. Zhou and Jiang [12] found that 86% of Android malware samples are repackaged apps produced by injecting malicious components into legitimate apps. The injected malicious components are hidden within the functionalities of popular apps and usually constitute only a small portion of the repackaged apps. Android has became a major target for mobile malware [29].

This paper attempts to discuss all techniques being used by the malware and how the Android Operating System has evolved to counter the known exploits and vulnerabilities. We will discuss the impact of Android Fragmentation owing to OEM customization which leads to additional vulnerability. Also the use freedom to choose the Apps from official and third party sources which often introduces a lack of central control over apps distribution leading to malware existence and steps being taken to mitigate its harmful effects. This paper would also suggest some security settings which would mitigate the installation of malware and keep the device safe.

II. BRIEF MALWARE HISTORY

Malware have evolved as the popularity of Android system grew larger. From the users' perspective, an exploit program can help them to bypass the security mechanism of their Android devices to achieve better control of their devices by obtaining a higher privilege, e.g., rooting their devices. On the other hand, the exploitation could also be misused to gain the control of victims' devices where the attacker

can obtain financial profit from selling users' privacy (e.g., account information). Meng et.al provides a taxonomy for android malware highlighting various attack surfaces possible on the Android system [53]. We present a brief history of malware for Android.

(a) The Begining: 2010

The first malware discovered was Fakeplayer in Aug 2010, which disguised as a video player that was not able to play any video file but was able to send SMS to premium rate numbers at the cost of victim to generate revenue. Similarly more malware with different masquerades were focused on sending SMS to premium numbers in the background without users knowledge and uploading location information of user to remote server.

(b) Backdoors and Root Exploits: 2011

This year saw the evolution of malware. Backdoors server became more common as every malware had multiple hardcoded server. All too frequently, a backdoor will be bundled with a root exploit, and if the device is infected, the malicious user will take full control of the mobile device. Russian virus writers focused on SMS Trojan for revenue and Chinese virus writers focused upon backdoor and root access to remotely control the device. Root exploit typically targeted older version of android (2.1, 2.2) were also being employed like Rage Against the Cage (RATC) and Exploid [2]. Malware also started stealing personal user data and data about the device, subscriber number, IMEI no. Sim no. etc These programs steal personal user data and/or data about the infected mobile device. Malware were discovered on Android Market available for download with no mechanism in place on the market to detect and remove malicious code. Hamandi, Khodor, et al [41] in their work showed the increased amount of SMS fraud during this period. In late december, arspan trojan appeared which was a mass mailer promoting acts of Hactivism in the middle east countries. Root exploits based on vulnerabilities of android version 2.3 [5,6] were implemented to grant root privilege.

(c) Banking Fraud: 2012

This year saw the emergence of malware that target banking institutions. The malware (Spitmo, Citmo, Zitmo etc) performs a Man-in the-Middle attack by forwarding OTP / mTAN received by user to remote location. The malware also masquerades as a banking App and displays a phishing web page of the original site on its Web-kit element and steals users credentials. All the malware now feature a Command & Control Center to upload user information. The malware also started clickfraud on infected device to generate user input to click on advertisement for revenue generation. Further, 72% of malware reported during this year asked for access to read low level system logs. Sending SMS to premium numbers still remained fairly noticeable on all malware reported. The period also saw the rise of malware requesting the INSTALL_PACKAGE permission to install additional apps in the

background. The malware were also able to hide itself by modifying its manifest file to remove the app from the launcher [4]. Obfuscation was also used for repackaged apps having malicious code. Almost 70% of the malware contained encrypted C&C server.

(d) Getting Sophisticated: 2013

User awareness towards malware was increased which led to attackers shifting their strategies towards creating scareware. The users were warned about possible virus in their mobile and were led to download a anti-virus loaded with malware. The malware writers also started convincing the user to grant device admin permission which the user had to give manually deep in the settings page. Many malware utilized the Master key vulnerability found in Android 4.3 to update legit apps on the device with malware [7,8]. This period also saw the emergence of keylogger for android disguised as keyboard with swipe feature and convinces the user to change the input settings of the device to the malware. Dynamic code loading was observed with malware writers resorting to Java Reflection, downloading malicious code from C&C which avoids static analysis technique. These were noted for change in behavior based on commands received from command and control (C&C) servers[10]. Malware were also increased its evasive nature by fingerprinting the environment and only deploying malicious code when not running under an emulator or Google Bouncer [4].

(e) OEM introduced Malware: 2014

It saw the rise of Advertisement related malware which steals user info. Also the first crypto currency miner was spotted which uses the computational power of mobile to mine con currency. Second half of the year saw the emergence of ran somware while encrypting data on EXT storage. Some were actual ran somware which resorted to encryption while other were scare-ware. Also, few malware locked the user out completely by changing the lock screen code and demanded the ransom. Furthermore, worm like behavior was observed by malware by sending SMS to all contacts on the infected device containing link to download the malware. Malware are created with lots of advertisement and often used to direct the user to play store to give better ratings for the malware developer app to establish credence on the malware. Android boot-kit was also discovered which was embedded deep in the firmware (system and boot partition) of various Chinese devices[23,24]. Malware with banking fraud continued to incre2.7 | Rise of Scareware: 2016ase during this period.

(f) Obfuscation: 2015

This period saw a reduce in number of malware resorting to sending SMS to premium numbers as revenue generation. Adware had a very high increase in its share in 2015. It seems that this kind of displaying Ads to the user and tricking him/her into clicking the Ads became very popular and a new business-model on Android. The number of banking Trojans also doubled from last year [9]. Malware

writers were also seen resorting to use code obfuscation and encryption extensively to evade Google Bouncer. Further increased use of dynamic code was noticed and most malware tends to download malicious code from C&C server after installation on the device.

(g) Rise of Scareware: 2016

This year saw the Growth in the popularity of malicious programs using super-user rights, primarily advertising Trojans. Malware also resorted to creation of botnets for automatic click fraud on android. One of the botnet was available on Google play store and had millions of download until it was finally removed [40]. Smishing campaign were utilised by The period did not saw emergence of any new tactics employed by malware writers and with the introduction of run time permission based model arriving with Android Marshmallow (6.0) update, users were more aware about the context of permission being requested by individual Apps. Although the period did noticed increase in scareware which led unsuspecting users to download malicious Apps. Kaspersky reports [42] that Android devices running versions higher than 4.4.4 have much fewer exploitable vulnerabilities. However, there are still about 60% devices running old versions of Android that are vulnerable to rooting attack.

(h) Tighter Android Security: 2017

Rooting attacks continued to nose dive as the number of vulnerable android versions decreased. This period saw the increased number of mobile banking Trojans to draw a phishing overlay on top of genuine banking apps to steal user credentials [47]. Meltdown and Spectre vulnerability were also reported for ARM and snapdragon processors [48], but its actual impact was not utilized on this platform. Malware were mostly available on third-party app stores with the official play store adopting strict measures with the introduction of Google protect play which uses adaptive machine learning to detect dynamic malware on the app store.

(i) 2018

Privacy related malware continue to dominate the Android Operating System. Malware using advertisement framework on repackaged Apps have been available on official play store and were subsequently taken down. This period did not saw any new technique being employed by malware writers. There was an increase in the number of banking frauds through phishing.

Based on this approach, the CSLA adder blocks of 2:1 mux, Half Adder (HA), and FA are evaluated and listed in Table I.

III. TREND IN MALWARE

Malware have become more sophisticated over the years with the volume of permission being sought.

Figure-1 shows the trend of top five permissions being used by malware. Internet is the most requested permission by malware, a trend which is seen equally among benign apps as well. External storage does not offer the android sandbox feature and most files there are world writable for any App that has this permission. The permission to install packages is considered dangerous as it allows apps to download malware from third party sources or from the internet. However, there was a decline in this permission and external storage with the run-time permission model from android 6.0 and the fact that root exploits were not effective with device having android 4.4.4 and greater. There is a steady trend in the use of GPS location, tracking users daily routine.

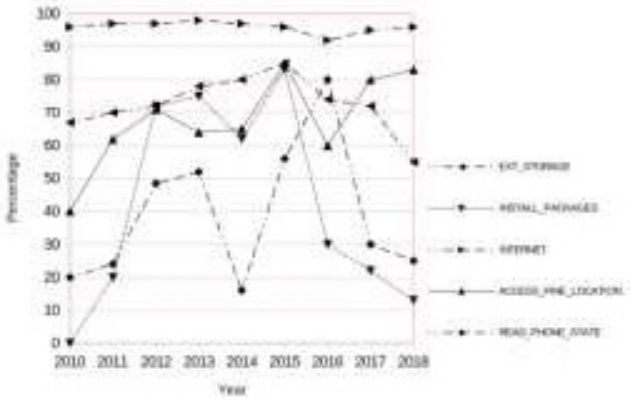

Figure 1: Permission requested by malware

Figure-2 shows that there is a sharp decline seen in malware sending premium SMS for monetary gains from 2013 onward, which was also countered by carrier and Google [55]. Malware continue to contact C&C server and the server names were being obfuscated or encrypted malware writers. New servers kept coming up with old ones being blacklisted, a trend which is observed even in 2018. Malware have benefited from the plethora of Advertisement SDK available and increased click- fraud trend was observed since 2011. Ransomware Apps started in 2012, initially limited to locking the user out by changing lock screen PIN. But the trend has shifted to locking out user and later indulging in actual encryption of files in external storage like Doublelocker [56]. Crypto-currency miners for android were first noticed in 2013 and have been steadily rising ever since.

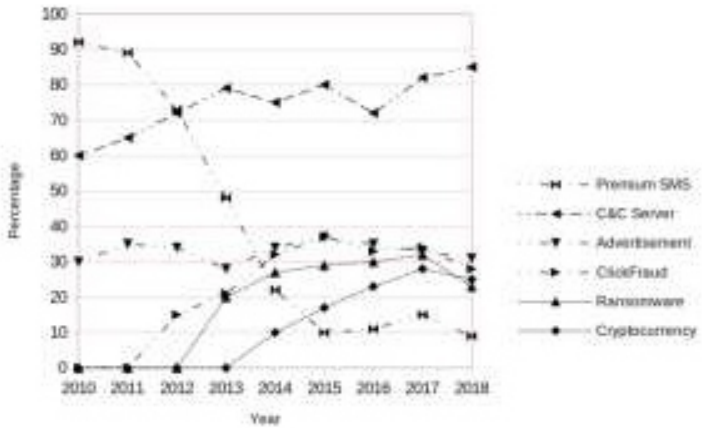

Figure 2: Types of malware

IV. PERSISTENCE TECHNIQUES USED BY MALWARE

Malware writer uses various techniques in the Android ecosystem to gain persistence. Android uses various system broadcast messages to inform Apps about events generated in the operating system. For example, ACTION_BOOT_COMPLETED indicates that the OS has been booted up. Malware regularly registers themselves to receive such broadcast messages to initiate its malicious code. Other common messages that the malware look for are RECEIVED_SMS, RECEIVED_CALL, ACTION_POWER_CONNECTED etc. Based on these initiating agents, the malware then deploy background services to carry out hidden activities and hide itself from the unsuspecting user. The background service performs a variety of malicious activities like connecting to the Command & Control Center to upload data, stealing clipboard data, Location tracking, recording ambient audio and uploading last recorded audio file subjected to geofencing and also to get dynamic code and instructions.

Android allows developers to declare the broadcast receiver or intent messages from other sources in the AndroidManifest.xml file. However, it also gives the flexibility of it being declared dynamically in java code. The former is subjected to static analysis which can look for these excessive permissions in the manifest file and flag them as potential malware. Thus malware writers have started using dynamic code and java reflection to register for these receivers at run-time. It is also seen that malware often come as a clean package but later download the malicious code from C&C [43]. Since Android 8.0 (SDK API 26), the system is able to kill idle background services, a self-protection feature was also seen in the recent code which raises a fake notification to prevent it from becoming idle. Furthermore, Android P forces the use of foreground service which has to show a notification to the user if an App is using privacy related hardware such as camera, microphone etc, which makes the Android environment more transparent [44].

Malware have been known to use root exploit to gain persistence on the system. As brought earlier, root methods are not as effective for Android version greater than 5.0 [42]. Although Proof of Concept (PoC) exists for these versions of Android which displays vulnerabilities that can be used to gain root access on these devices, its actual implementation is not visible. Instead, malware have started using the device administrator permission which has to be manually granted by user themselves through the use of social engineering. This might be in the form of a scareware, which promises to remove a non-existent Trojan from the android device by installing a fake anti-virus App which would ask for the Device Admin Permission to complete its task. The malware can then utilize elevated privileges to hide itself and perform privileged operations such as changing lock- screen pin code, locking device, wiping device data, etc.

V. ANDROID VULNERABILITIES

The Android software stack can be subdivided into five layers: the Linux Kernel and lower level tools, System Libraries, the Android Runtime, the Application Framework and Application layer on top of all. Each layer provides different services to the layer just above it. Shewale, Himanshu, et al.2014 [26] have classified various vulnerabilities present in these layers which have been exploited by malware. As demonstrated, android framework layer has the maximum number of vulnerability. Many more vulnerabilities have discovered and the most critical ones are discussed in this section.

(a) Linux Kernel Layer

The Dirty Cow (Dirty Copy-On-Write), or CVE- 2016-5195 vulnerability which existed for over 9 years was also used to root the Android OS through through a race condition bug and gain write-access to read-only memory[27,28]. Another vulnerability, CVE-2015-1805 was discovered when the failed copy command could be used to possibly gain privileges via a crafted application, aka an "I/O vector array overrun."[13]. The RowHammer uses hardware vulnerability of DRAM to flip bits by repeated flushing and can be utilised to gain root privilege [15].

(b) Android Framework Vulnerabilities

Application layer root exploits mostly include vulnerable logics introduced by setuid utilities, system applications, or services. Stagefright is a exploit which utilizes the code library for media playback in Android called libstagefright CVE 2015- 1538. The libstagefright engine is used to execute code which is received in the form of a malicious video via MMS, thus requiring only the mobile number of the victim to carry out a successful attack [17]. Furthermore vulnerability in mediaserver (CVE- 2016-6074, CVE-2016-3862) were also discovered with high CVSS score of 10. Motorola XoomFE devices have found to contain a command injection vulnerability [38]. Another instance is a backdoor- like setuid binary shipped with certain ZTE Android devices (CVE-2012-2949)

© Vendor Libraries

On Samsung Galaxy S4 through S7 devices, an integer overflow condition exists within libomacp.so when parsing OMACP messages (within WAP Push SMS messages) leading to a heap corruption (CVE- 2016-7990). Vendor libraries also include advertisement SDK which are know to steal private data like Taomike library [34]. ObjectInputStream vulnerability (CVE-2014-7911) does not verify that deserialization will result in an object that met the requirements for serialization, which allows attackers to execute arbitrary code [37].

Majority of the exposed vulnerabilities are attributed to mistakes in bound-checking, buffer overflow, incorrect pointer dereference and input not verified. Jimenez, Matthieu, et al [33] that showed that the

vulnerable android component fall under nine different category, i.e. Driver, Library, Messaging, Networking, Access Control, Browsing, Cryptography, Dalvik and Debug. It is seen from the CVSS score of various android versions that after Android 6.0, the number of vulnerabilities with CVSS score higher than 8.0 has significantly reduced[32].

VI. OEM CUSTOMISATIONS AND CUSTOM ROM

Android ROM is the basic OS firmware layer of the Phone. This is the base for phone operations. In file system generally this part is stored under /system and may have /data partition also in case it holds some user specific settings. This is the portion which contains all quintessential parts of the operating system for proper functionality of the Phone. Starting with Linux kernel along with it modules to Dalvik VM, combined with core libraries and user libraries (SQLite etc). This same portion also features the application framework which allows for seamless interaction of android applications with android core features, including the telephone. On top of all this we see the applications running in Dalvik VM. The Android Operating System is licensed under the terms of the Apache Software License 2.0 which does not mandate source code availability [30]. This also allows OEMs to package Android with binary libraries which they are not forced to make open source. The underlying Linux kernel instead is licensed under the terms of the Gnu General Public License (GPLv2). That license requires that each modification or addition of code must be made available to the customers of a device with that software.

(a) Android Fragmentation

The source code of the Android Open Source Project (AOSP) project in its original form is deployed only to a selected set of devices. For each major release of Android Google partners with one OEM to create a device of the Nexus / Pixel brand. OEMs modify the code taken from the AOSP and enhance it with their own custom code. They add new pre-installed applications, tweak the user interface and add additional functionality to stock applications to set each other apart (refer Figure-3). Taken together, these modifications to stock AOSP Android are called a Skin. Additionally, many carriers add custom applications to the devices they sell. Then, we have the Linux Kernel, which may also contain OEM customizations and drivers for controlling and interacting with various peripheral System on Chips (SoCs) found on the device's board. The next level is a chain of bootloaders which either originate from the OEM or the chipset manufacturer: At its lowest level, we have in Boot ROM the Primary Bootloader (PBL), which is written by the chipset manufacturer, and then usually a series of bootloaders that end with the late stage Android (Applications) Bootloader (ABOOT).

Figure 3: Build process of android

With every device manufacturer teaming up with different SoC manufacturer and also implements proprietary hardware to have an advantage over high competition in the field of Android customized devices. Different from the main kernel code that runs on almost every device, vendors either customize the kernel (e.g., Qualcomm's custom Linux kernel branch) or provide vendor-specific device drivers for various peripherals (e.g., camera, sound). Android phone manufacturers are under the perpetual pressure to move quickly on their new models, continuously customizing Android to fit their hardware. However, the security implications of this practice are less known, particularly when it comes to the changes made to Android's Linux device drivers, e.g., those for camera, GPS, NFC etc. It leads to additional vulnerability over and above the stock devices produced by google. Hay, Roee [35] showed inherent vulnerability in in OnePlus 3 device and various nexus devices which can be exploited in fastboot mode to gain access, for example in OnePlus 3 device, Android Debug Bridge (ADB) mode can be accessed through charger mode (when the device is switched off) and with the device connected to PC using USB cable. It also shows various low level techniques to interact with SoC using Inter-Integrated Circuit (I2C) protocol [36] to leak sensitive data. Since these drivers are produced by single-party, they are not put under the lens for auditing. However, the particular exploit is limited to these customized devices only.

Zhang et al [39] shows that from the highest to lowest, the order of impact and generality of exploits to be 1) the kernel exploits, 2) the exploits targeting libraries that are used by Android system processes, 3) exploits targeting system applications or services, and 4) exploits against vendor-specific device drivers, applications and programs. Also out of total root exploits, the ones caused by vendor customization to kernel and additional libraries account for 71% of the total vulnerabilities (others related to general vulnerability affecting all devices based on android version). We can see that the fragmentation of Android is a major source of the introducing vulnerabilities to the user.

Each new model of smartphone comes with hardware improvement and the firmware to drive it. Mid-range and low-range smartphone are known to manufacture budget phones with malware installed in the firmware [46]. These are known to contain bootkit which are impossible to remove without flashing the firmware without malware in it as it occupies the system partition. Russian and Chinese smartphone manufactures have the highest number of bootkit in their firmware. Moreover, lack of malware free custom ROM are also not availble for the budget phones which leaves its users no aftermarket choice. The vendor customization does not always introduces vulnerabilities, but some are designed towards enhanced enterprise grade security features. One such security feature introduced by Samsung is KNOX [51, 52]. It forms a hardware root of trust and forms an isolated space for user data using hardware encryption.

(b) Countering Android Fragmentation

In May 17, Google had announced Project Treble to mitigate the Android Fragmentation problem [49]. It attempts to introduce a vendor implementation layer which provides a stable platform for a particular device assembled from various SoC and hardware. Earlier, with any major android version upgrade, the device manufacturers had to rework the low-level implementation, i.e. how the operating system interacts with hardware. For e.g a device manufacturer could write all the code specific to their handsets, to power their unique camera system or biometric security, as one isolated part of the operating system. That software could then be locked, with the latest Android software applied on top.

When Google updates Android, the update could be made available day one, because none of the underlying hardware instructions have changed. It does cut down at least three months for the vendor to push new update for their device but it does not involve vendors motivation to update considering the financial aspect [50]. However, if it is forced by Google, it would put an end to outdated android versions and would have a tighter grip on the upgrade process.

VII. ANDROID APP MARKET

The official App store for android is Google Play Store (erstwhile Android Market place) allowing users to browse and install applications developed with Android software development kit (SDK). Owing to its open source development, the play store has continued to expand in terms of number of apps it offers and has grown to 2.6 million apps with more than 82 billion downloads [21,22]. Starting from 2012 (android 4.2), the official play store introduced application verification at individual device by scanning all the apps sideloaded from third party market places (known as on-device bouncer) [25,45]. This initial step seems to provide limited respite as it was able to detect only 15.32% of malware [31].

Study suggests that the newer version of Android are less susceptible to malware if the users utilize Google Play Store to download Apps [19]. This has been possible byfactors, such as continued platform and API hardening, ongoing security updates and app security and developer training to reduce apps' access to sensitive data. If apps are being side-loaded by the user, Androids verified application can scan the app for any known vulnerability and warn the user in advance or block the installation completely on the device [20].

VIII. PRACTICES UNDER SCRUTINY

Here we look on the security misconfigurations that can happen. We are focusing basically on the core layer of android, detailing about various settings and configurations which might result in a total security breach of android device and should be avoided by normal user.

(a) USB Debugging enabled

USB debugging or ADB is Google's method for debugging support this is the setting which needs to be enabled when you are doing development or debugging of application, however there is no need to keep this setting enabled when its a normal user system. ADB Bridge supports push and pull of files and folders from all the directories where adb user have access. Adb has various options which allows many more features including (1) Logcat collection, (2) Installation of software, (3) Remount of system partition with rw. ADB also allows for fastboot which in turn allows a user to run non verified or unofficial kernel without even overwriting the stock data.

(b) ADB shell over WiFi

Another variable which could be set to allow adb shell access. However this time access is over wifi network. Variable : service.adb.tcp.port = To set this variable you can either place it in build.prop or use command line #setprop service.adb.tcp.port=3355 This will mark port 3355 on phone to be usable to attach using adb. This can be avoided for daily usage.

(c) System permissions

In Android Devices, system partition is the most important partition which holds all the system critical files, as per general policy this partition is marked as RO i.e. read-only. However a general after market practice which is observed is to mark system partition as rw. The general use case is that by putting system in rw mode it is easy to work on modification of system data. The most harmful setting is if your ROM maker marks system with 777 i.e. rwx or read write execute permission for all users. When a system is marked with write permission it will allow a user to update / modify content of /system partition. Some of the crucial folders include /system/app or /system/bin. This permission is an open invitation to rootkits, malware, viruses and all similar items to start manhandling the device.

(d) Installation from unknown source

This specific setting is a security issue in itself. This check allows a user to install software which are not part of android market. Users who don't have access to android market this is the only way to install application, which holds true for Chinese users..

(e) Super User (SU) acccess and settings

Rooting of android phone is generally associated with installing SU binary. This binary allows a user for shifting the user to root. This is accompanied with superuser.apk which acts as a control agent. However there can be multiple scenarios's which need to be throughly examined. (1) SU binary installed and superuser.apk installed (2) SU binary installed but superuser.apk missing. (3) SU missing but superuser.apk installed (4) SU and superuser.apk both missing. Case 1 denotes max protection possible. Case 2 is a critical case as superuser.apk is the governing control over SU binary and if its not there then SU could be called directly without fear of user prompt.

IX. SUMMARY

In this paper we have seen the evolution of both Android Malware. The focus of malware writers shifted with the change in security policy implemented by Android System. The fragmented nature of Android system can be prone to older vulnerabilities owing to lack of upgrade but it makes android exploits more device specific. as the security mechanism of both the Android system and Linux kernel have been significantly strengthened, exploits targeting Linux kernel and Android system components experience decline; and vendors' customization becomes the prominent attack target in newly released exploits We have also highlighted certain simple methods which can be implemented by user to safeguard his android device against malware.

REFERENCES

[1] Mobile Operating System Market Share Worldwide http://gs.statcounter.com/os-market-share/mobile/worldwide (accessed on 28 Jan 19)

[2] Android Vulnerabilities https://androidvulnerabilities.org/by/year/ (accessed on 29 Jan 19)

[3] Malware in Mobile Devices https://www.gnu.org/proprietary/malware-mobiles.html (accessed on 29 Jan 19)

[4] Shan, Zhiyong, Iulian Neamtiu, and Raina Samuel. "Self- hiding behavior in Android apps: detection and characterization."Proceedings of the 40th International Conference on Software Engineering. ACM, 2018

[5] CVE-2011-1823 Detail https://nvd.nist.gov/vuln/detail/CVE- 2011-1823 (accessed on 29 Jan 19)

[6] First malware using Android Gingerbreak root exploit https://nakedsecurity.sophos.com/2011/08/22/first-malware- using-android-gingerbreak-exploit/ (accessed on 29 Jan 19)

[7] Android —Master Key‖ vulnerability – more malware exploits code verification bypass.https://nakedsecurity.sophos.com/2013/08/09/android- master-key-vulnerability-more-malware-found-exploiting- code-verification-bypass/ (accessed on 30 Jan 19)

[8] Android —Master Key‖ vulnerability more malware exploits code verification bypass https://nakedsecurity.sophos.com/2013/08/09/android-master- key-vulnerability-more-malware-found-exploiting-code- verification-bypass/ (accessed on 30 Jan 19)

[9] Mobile malware evolution 2015 https://securelist.com/mobile-malware-evolution- 2015/73839/ (accessed on 30 Jan 19)

[10] P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, —A survey of mobile malware in the wild,‖ in Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM'11), Chicago, Ill, USA, October 2011.

[11] 9 Year Old Linux Kernel bug dubbed _Dirty Cow' can Root every version of Android https://www.xda-developers.com/9- year-old-linux-kernel-bug-dubbed-dirty-cow-can-root-every- version-of-android/ (accessed on 30 Jan 19)

[12] Y. Zhou and X. Jiang, Dissecting Android malware: Characterization and evolution,‖ in Proc. IEEE S&P, May 2012, pp. 95–109.

[13] Android Security Advisory—2016-03-18 https://source.android.com/security/advisory/2016-03- 18.html (accessed on 30 Jan 19)

[14] It's Bugs All the Way Down http://vulnfactory.org/blog/ (accessed on 01 Feb 19)

[15] New Rowhammer Exploits use Hardware Vulnerabilities to Root LG, Samsung, and Motorola Devices, https://www.xda- developers.com/new-rowhammer-exploits-use-hardware-vulnerabilities-to-root-lg-samsung-and-motorola-devices/ (accessed on 01 Feb 19)

[16] These popular Android phones came with vulnerabilities pre- installed https://www.cnet.com/news/these-popular-android- phones-came-with-vulnerabilities-pre-installed/ (accessed on 01 Feb 19)

[17]Stagefright Explained: The Exploit That Changed Android https://www.xda-developers.com /stagefright-explained-the- exploit-that-changed-android/ (accessed on 01 Feb 19)

[18] Thomas, Daniel R., et al. "The lifetime of Android API vulnerabilities: case study on the JavaScript-to-Java interface." Cambridge International Workshop on Security Protocols. Springer, Cham, 2015.

[19] Newer Android versions are less affected by malwarehttps://www.zdnet.com/article/google-newer- android-versions-are-less-affected-by-malware/(accessed on 01 Feb 19)
[20] Verified Applications https://www.businessinsider.in/How- To-Protect-Your-Android-Phone-From-Harmful- Apps/articleshow/33832942.cms(accessed on 01 Feb 19)
[21] Number of available applications in the Google Play Store from December 2009 to December 2018https://www.statista.com/statistics/266210/number-of- available-applications-in-the-google-play-store/(accessed on 01 Feb 19)
[22] Number of apps available in leading app stores as of 3rd quarter 2018 https://www.statista.com/-statistics/276623/number-of-apps- available-in-leading-app-stores/ (accessed on 01 Feb 19)
[23]Modded firmware may harbour world's first Android bootkit https://www.zdnet.com/-article/modded-firmware-may- harbour-worlds-first-android-bootkit/ (accessed on 01 Feb 19)
[24]Most Sophisticated Android Bootkit Malware ever Detectedhttps://thehack-ernews.com/2014/04/most- sophisticated-android-bootkit.html (accessed on 01 Feb 19)
[25] Google describes how Android 4.2's app verification checks your downloads for malwarehttps://-thenextweb.com/google/2012/11/14/google- describes-how-android-4-2s-app-verification-checks-your- downloads-for-malware/(accessed on 01 Feb 19)
[26] Shewale, Himanshu, et al. "Analysis of Android vulnerabilities and modern exploitation techniques."ICTACT Journal on Communication Technology5.1 (2014): 863-867.
[27] VulnerabilityDetails https://github.com/dirtycow/dirtycow.github.io/wiki/Vulnera bilityDetails(accessed on 01 Feb 19)
[28] Proof of Concept Dirty Cow https://github.com/dirtycow/dirtycow.github.io/-wiki/PoCs(accessed on 01 Feb 19)
[29] Symantec Internet Security Threat Reort 2013 https://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v18_2012_21291018.en-us.pdf(accessed on 08 Feb 19)
[30] AOSP License https://source.android.com/setup/start/licenses(accessed on 01 Feb 19)
[31] An evaluation of the application verification service in android 4.2 https://www.csc2.ncsu.-edu/faculty/xjiang4/appverify/(accessed on 08 Feb 19)
[32] List of CVE for Android https://www.cvedetails.com/product/19997/Google Android.html?-vendor_id=1224 (accessed on 01 Feb 19)
[33] Jimenez, Matthieu, et al. "Profiling android vulnerabilities." 2016 IEEE International Conference on Software Quality, Reliability and Security (QRS 2016). IEEE Computer Society, 2016.
[34]18,000 Android Apps Contains Code that Spy on Your Text Messageshttps://thehackernews.com/-2015/10/android-apps- steal-sms.html
[35] Hay, Roee. "fastboot oem vuln: android bootloader vulnerabilities in vendor customizations." 11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17). USENIX} Association}, 2017.
[36] I2C bus protocol Tutorial, Interface with applications https://www.elprocus.com/i2c-bus-protocol-tutorial-interface- applications/(accessed on 08Feb 19)
[37]Android <5.0 Privilege Escalation using ObjectInputStream (CVE-2014-7911) https://cve.mitre.org/cgi- bin/cvename.cgi?name=CVE-2014-7911(accessed on 08 Feb 19)
[38] Xoom FE: Stupid Bugs, and More Plagiarismhttp://vulnfactory.org/blog/2012/02/-18/xoom-fe- stupid-bugs-and-more-plagiarism/ (accessed on 08 Feb 19)
[39] Zhang, Hang, Dongdong She, and Zhiyun Qian. "Android root and its providers: A double-edged sword." Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015.
[40] Viking Horde malware attacks Android devices https://www.cnet.com/news/viking-horde-malware-attacks-android-devices/(accessed on 08 Feb 19)
[41] Hamandi, Khodor, et al. "Messaging Attacks on Android: Vulnerabilities and Intrusion Detection." Mobile Information Systems 2015 (2015).
[42]Attack on Zygote: a new twist in the evolution of mobile threats https://securelist.com/attack-on-zygote-a-new-twist- in-the-evolution-of-mobile-threats/74032/(accessed on 08 Feb 19)
[43] Skygofree: Following in the footsteps of HackingTeam https://securelist.com/skygofree-following-in-the-footsteps- of-hackingteam/83603/(accessed on 02 Feb 19)
[44] Android P will stop malware from spying on you https://www.techradar.com/news/android-p-will-stop- malware-from-spying-on-you(accessed on 03 Feb 19)
[45] Android and Security https://googlemobile.blogspot.com/2012/02/android-and-security.html (accessed on 02 Feb 19)

[46] Khandelwal, More firmware backdoor found in cheap android phones, https://thehacker-news.com/2016/12/hacking-android-smartphone.html, 12 2016, (Accessed on 02 Feb 19).
[47] Mobile malware evolution 2017https://securelist.com/mobile- malware-review-2017/84139/ (accessed on 29 Jan 19)
[48] Meltdown Hack and Spectre Bug: How it affects Android https://www.androidcentral.com-/meltdown-spectre (accessed on 29 Jan 19)
[49] Here comes Treble: A modular base for Androidhttps://android developers.googleblog.com-/2017/05/here-comes-treble-modular-base-for.html (accessed on 10 Feb 19)
[50] Solving the Problem of Android Fragmentation- Project Treble https://medium.com/@lucideus/-solving-the-problem- of-android-fragmentation-project-treble-lucideus- 22271d989324 (accessed on 10 Feb 19)
[51] Samsung Knox: A closer look at Samsung's security platform https://www.digit.in/mobile-phones/samsung-knox-a-closer- look-at-samsungs-security-platform-41257.html (accessed on 10 Feb 19)
[52] Samsung Knox https://www.androidcentral.com/what- samsung-knox (accessed on 10 Feb 19)
[53] Meng, Huasong, et al. "A survey of Android exploits in the wild."Computers & Security 76 (2018): 71-91.
[54] Suarez-Tangil, Guillermo, and Gianluca Stringhini. "Eight Years of Rider Measurement in the Android Malware Ecosystem: Evolution and Lessons Learned."arXiv preprint arXiv:1801.08115 (2018).
[55] How Google Detects and Warns about Premium SMS Messages, https://www.xda developers.com/how-google- detects-and-warns-about-premium-sms-messages/ (accessed on 10 Feb 19)
[56] Double Locker: Innovative Android Ransomware https://www.welivesecurity.com/2017/10/-13/doublelocker-innovative-android-malware/ (accessed on 10 Feb 19)

Forecasting A South-West Monsoon Onset Using Neural Networks

[1]S. Ramanayake, [2]H. L. Premaratne
[1]Sri Lanka Institute of Advanced Technological Education, Sri Lanka
[2]University of Colombo School of Computing, Colombo, Sri Lanka
E-mail: [1]sudarshana@sliate.ac.lk, [2]hlp@ucsc.cmb.ac.lk

ABSTRACT

This paper presents the performance of an artificial neural network to forecast the arrival of South West monsoon to Sri Lanka. Ground level precipitation data of the wet zone and cloud shape stability time extracted from satellite images were the inputs of the network. A feed-forward back propagation network was used to predict the onset within the potential period of April, May and June. Output of the network is either 1 or 0 which represents whether a particular day satisfies the monsoon condition or not. A predetermined consecutive number of such occurrences appear is taken as onset day of the monsoon. The network was trained with the onset determined by the Meteorological Department of India which is available in a public web site. Data for the period from 2012 to 2016 were used to train the network and the data in 2017 and 2018 and also in previous years were used to test the network. The proposed system is able to predict the monsoon situation with an accuracy of 90.26± 10.49%.

Keywords- Neural Networks, Cloud Shape Stability, Precipitation, Monsoon Onset

I. INTRODUCTION

Sri Lanka receives a fair amount of rainfall throughout the year but some what regular rainfall is received during the monsoon seasons. The island annually experiences the two monsoon periods called the South West Monsoon and the North East Monsoon. Due to these two rainy seasons and based on the amount of rainfall received, country is divided into two zones, the wet zone and the dry zone. SW Monsoon which normally begins in end of May and beginning of June is the cause of getting more rain to the wet zone or south west part of the island (Dom roes et. al., 1993). Most of the years, a large amount of rainfall is received soon after the arrival of the SW Monsoon and in some seasons a situation of flood occur soon after the monsoon onset.

Rainfall is a crucial factor for the economy of the country due to the large extends on agricultural productions. Cultivation and harvesting times are directly linked to the monsoon activity. In addition to the agriculture, a study of onset of monsoon and precipitation received are also required to drought management, power production, drain and water resource management, disaster management etc. Hence forecasting the SW monsoon onset is significant in many facets.

Visible phenomenon of the monsoon onset in each year is a sharp increase and characteristic

Visible phenomenon of the monsoon onset in each year is a sharp increase and characteristic persistency in rainfall. However thunderstorm activities are highly pronounced and precipitation piercing on the eve of SW monsoon in this region. This leads to make a bogus onset (Gautam D. K. et. al., 2013). Therefore, only rainfall data is not sufficient to determine the onset.

In general, numerical and statistical models are used in weather forecasting in Sri Lanka and South Asian region (Tsing et. al., 1995; Sikka et. al., 1980; Wanget. al., 2004; Margaretet. al., 2009).Further, Empirical orthogonal function (EOF) analysis are utilized to investigate the horizontal wind in order to figure out Indian monsoon circulation (Tsing et. al., 1988). In addition, numerical simulation is performed to test the inter-annual variation of the Indian monsoon result from the response of this monsoon system to the inter-annual variation of the pacific sea surface temperature(SST) (Tsing et. al., 1994). In recent past, Neural Network models have been gaining popularity over weather forecasting (Weerasinghe et. al., 2010; Rathnayake et al 2011; John et. al., 2012).This is probably the simplicity of modelling on complex dynamical systems. However, no or less studies have been found in literature on forecasting SW monsoon onset in Sri Lankan region using neural networksin the best ofauthors'' knowledge.

II. METHODS AND MATERIALS

Thirty minute interval cloud images were downloaded from the "http://103.215.208.54//archive-/INSAT-3D- IMAGER/3D-ASIA-SECTOR/INFRARED-1". Images are taken from the Indian National Satellite System (INSAT) KALPANA 1 geostationary satellite. These images are derived from the emission by the earth and its atmosphere at thermal infrared wavelength of 10.5-12.5 μm and they cover the latitude range 100S-500N and the longitude range 450E – 1050E. The ground resolution at the sub satellite point is nominally 8 Km x 8 Km. Images are stored in the above web site in colour RGB JPEG format. Pixel resolution of the images is about 1200 x 1024 (72 dpi x 72 dpi) with 24bits depth including image header during the period 2012 to February 2014 and that for the period from March 2014 is about 1260 x 1580 (96 dpi x 96 dpi) with 24 bits depth including image header. These images were used to calculate the daily cloud shape stability of the potential monsoon onset months of April, May and June.

Local orientation which is used as a feature of a satellite image is a major contributor to determine the cloud shape life time. Local orientation is characterised by the least change of grey value in one direction and maximal change in the orthogonal direction (Bigun et. al., 1987; Premaratne et. al., 2002). Therefore, a linear symmetry tensor for an image is constructed with respect to the local neighbourhood for each pixel of the image. In this way, local symmetry tensors of the concerned bounded area of the 30 minute interval satellite images were constructed. By preserving the first image

as the reference image, comparison takes place with its LS tensor and that of the subsequent images till the correlation drops below a specified threshold. In determining the threshold, experiments were carried out for a range of thresholds and the value 0.9 gave the best performance. As long as this correlation of the subsequent image is higher than the threshold, it is considered as the same cloud shape with the reference image. The time period until the drop of the correlation of the LS tensors below the threshold is considered as the life time of that particular cloud shape. This process was continued by taking the next immediate image as the new reference image and mean value of the life times is taken as the cloud shape stability(CSS) time for the particular day (Ramanayake et. al., 2015).

Figure 1: Wet and dry zones, meteorological stations

The daily rainfall data of Colombo, Galle, Ratnapura & Kurunagala for the months of April, May and June used in this study were obtained from the Department of Meteorology, Sri Lanka. These stations are located within the wet zone reasonably well distributed across the zone (Figure 1). For each station, from 2012 to 2018 precipitation data of the above months were utilized.

To minimize the influence of the missing data, they were estimated using piecewise cubic Hermite interpolation. Available time series data from both end of the missing day or days were used for interpolation.

To develop the neural network model, both ground level precipitation data and satellite images have been utilised as describedabove. The proposed model uses previous three consecutive day data to forecast the following day monsoon status. Responses of training data were set according to the onset date determined by the Indian Meteorological Department which is available for public.Previous three days from the onset date and subsequent days were set to ones (1) and all other previous days were set to zeros (0). Supervised training was carried out for the data for the period from the year 2012 to 2016. The sample onset mapsfor the year 2017 which was published by the Indian Meteorological

Department are shown in figure 2. If the network outputs three consecutive 1s („true“) then the third day is taken as the onset day. Therefore, three days prior to the actual onset date determine by the authorities were also labelled as the 1sbeforefed to the network. Further, SW Monsoon onset day that has been taken from the web occurs in an Indian ocean just below the Island around latitude 50N and longitude between 790E and 840E. According to the Meteorological Department of India, the onset propagated inside into the country within 4 to 10 days after this occurring. Therefore, the model predicts the monsoon onset 4 to 10 days prior to appear within the country.

The inputs of the network which represent cloud shape stability (CSS) time and precipitation were classified and Likert scale values were assigned to each classes appropriately as shown in the table 1.

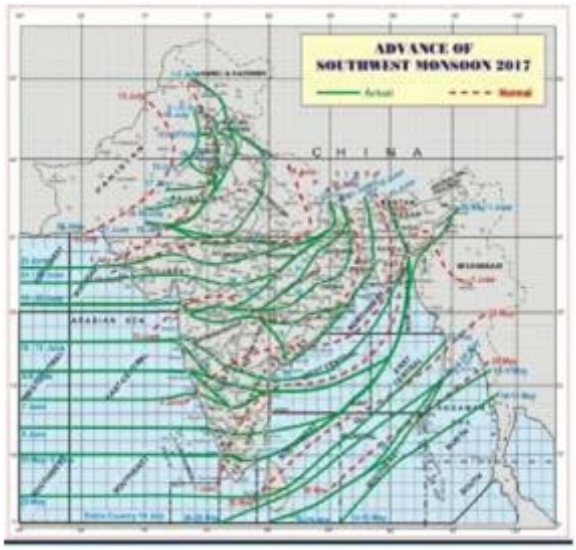

Figure 2 : South-West Monsoon onset propagation (Source: http://www.imd.gov.lk)

Boundary value of each class were determined by frequency investigation in order to equal the likely distribution of the scales. It is observed that, the cloud shape life time is decreasing when monsoon is emerging. Therefore, higher value was given to the low cloud shape life time. On the other hand precipitation is higher when monsoon is established. Therefore, higher scales were given to the higher precipitation.

Coded values are distributed reasonably well in all three months in every year. To determine whethera particular day is a monsoon day or not, the system considers the three previous consecutive day data as input and produces either one (1) or zero (0) as output, representing a monsoon day and a non-monsoon day respectively.

A feed forward back propagation neural network was used to implement the models. Several network architectures with different transfer functions and different number of neurons were tested. It was found that 15-7-1 three layer architecture with the transfer function „logsig“ and „Levenberg

Marquardt‟ MSE training algorithm showed the best performance. Graphical interpretation of the model architecture and training performance are shown in figure 3 and figure 4 respectively.

Cloud Shape (CSS) (hrs)	Stability	Precipitation (mm)		
less than or equal 1.0	5	less than 0.3		1
1.0 < CSS ≤ 2.0	4	between 0.3 1.9	and	2
2.0 < CSS≤ 3.0	3	between 2.0 6.9	and	3
3.0 < CSS ≤ 4.0	2	between 7.0 16.9	and	4
greater than 4.0	1	above 17.0		5

Table 1: Classification levels

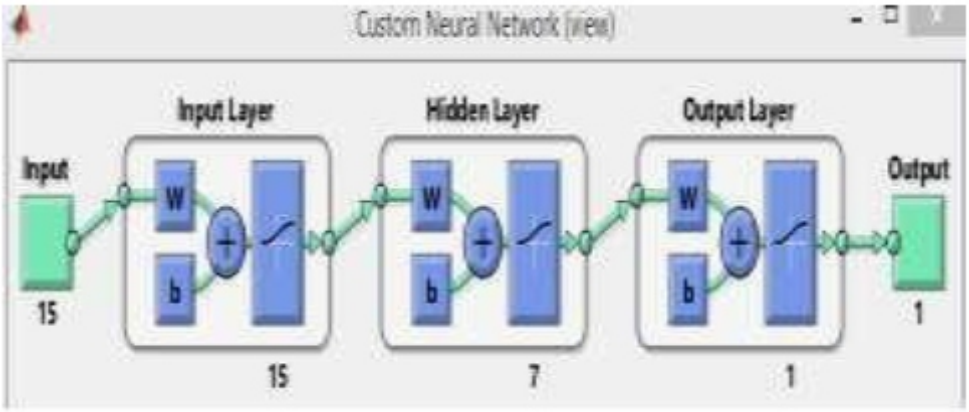

Figure 3: Neural network architecture

Table 2: Onset day forecasting		
Year	Actual	Predicted
2012	May-23	May-23
2013	May-20	May-19
2014	May-23	May-23
2015	May-21	May-21
2016	Jun – 03	Jun-03
2017	May-16	May-15
2018	May-25	May-28

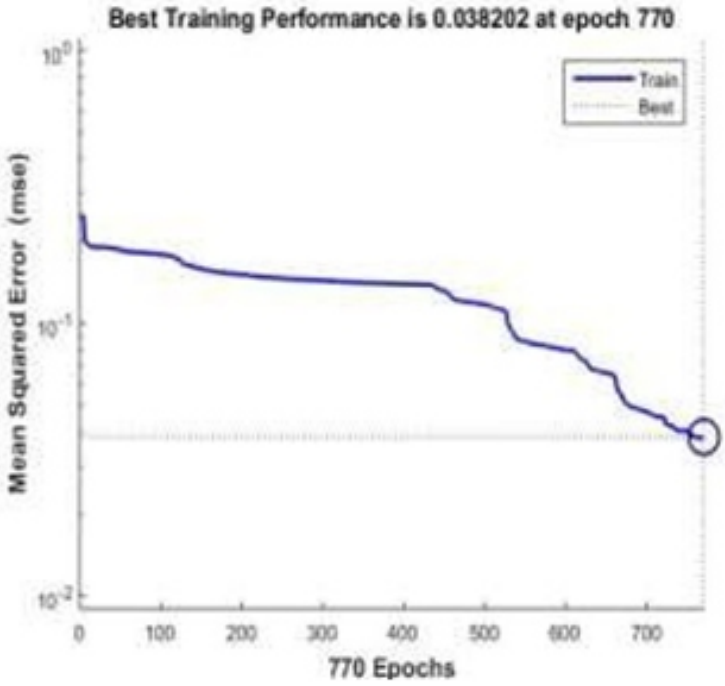

Figure 4: Network training performance

Training was done using the data for a period of five years from 2012 to 2016 with 15 x 445 input matrix and testing was done for all the years from 2012 to 2018. The prediction success rate was calculated as,

$$\text{Success Rate} = \frac{X_c}{X_{all}} \times 100\% \qquad (1)$$

where Xc is the number of correct prediction as monsoon day or non-monsoon day and X all is the total number of predictions. Further, accuracy of the model was determined using the root mean square (RMS) error which is given by the formula,

$$\text{RMS error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_{p^i} - X_{e\,i}\right)^2} \qquad (2)$$

where Xp and Xe represent the predicted and desired output respectively, and N is the total number of predictions. Success rate and the RMS error were calculated and presented year by year in order to see the year-wise performance.

III. RESULTS AND DISCUSSION

Actual onset days that were taken from the "www.imd.gov.in" and predicted onset days are presented in the table 2. Predicted onset day deviates from the actual value only in 2013, 2017 and 2018. It is one day ahead in first two years but in 2018, onset prediction is three days after the actual date. Further, from April06 to 09 and from April 17 to 19 in 2018 consecutive 1s were produced by the system and they are omitted to take as the onset date due to the larger distance to the potential onset period. All other years remain same with the adopted criteria of determining onset date that the last date of the three

consecutive 1s in the first occurrence after April 20. Summary of the prediction by the network success rate calculated using the equation 1 and the RMS calculated using the equation 2 is presented in the table 3.The lowest overall success rate and the highest RMS error are shown in 2017 next to the year 2012.

Table 3: Year wise prediction success rate

Year	Overall (%)	RMS Error	Pre Onset (%)	Post Onset (%)
2012	89.77	0.96	98.00	78.95
2013	98.86	0.11	100.00	97.56
2014	96.59	0.32	94.00	100.00
2015	98.86	0.11	97.92	100.00
2016	96.59	0.32	95.08	100.00
2017	75.00	2.35	76.74	73.33
2018	76.14	2.24	74.42	75.56
Average	90.26	0.91	90.88	89.34

IV. CONCLUSION

This study focuses to determine the arrival of SW monsoon in Indian Ocean southward to Sri Lanka. Previous studies have shown that monsoon propagates from South China Sea to north India and Nepal (Tsing et. al., 1994; Wanget. al., 2004). The proposed methodis able to forecast the onset appearing in Indian Ocean before entering the Island within 4-10 days in advance. However, the systemdoes not perform well when it trained with 3-4 years data. On the other hand, weather conditionsdoes depend on many parameters such as humidity, outgoing longwave radiation, temperature, wind direction, wind speed, cloud amount, sea surface temperature, sea level pressure etc. In the first phase of the research, it was found that the cloud shape stability and the cloud cover remarkably vary in the eve of the SW monsoon onset. Therefore, in the future,it is expected to improve the model by considering many different input data as well as for a larger duration and to predict the monsoon arrival 10- 30 days in advance.

ACKNOWLEDGEMENT

The authors appreciate the support given by the Department of Meteorology, Sri Lanka

REFERENCES

[1] Bigun, J., Gosta, H. Grandlund, (1987) Optimal Orientation Detection of Linear Symmetry, Proceedings of the IEEE First International Conference of Computer Vision. London, IEEE Computer Society Press, pp 433-438

[2] Bryan A Baum, Vasanth Tovinkere, Jay Titlow, and Ronald M Welch, (1997) Automated Cloud Classification of Global AVHRR Data Using a Fuzzy Logic Approach, Journal of Applied Meteorology, 36, 1519-1540

[3] Domroes M., and Ranatunge E., (1993), A statistical approach towards a regionalization of daily rainfall in Sri Lanka. International Journal of Climatology 13(7): 741-754

[4] Gautam D. K., and Regmi S. K., (2013)Recent Trends in the Onset and Withdrawal of Summer Monsoon Over Nepal, ECOPERSIA 1(4), 353-367

[5] John D., Sindhu K. K., and Meshram B. B. (2012) Two Stage Data Mining Technique for Fast Monsoon Onset Prediction. International Journal of Advanced Research in Computer Science and Software Engineering. 2, 6 150-154

[6] Margaret M. Wonsick, Rachel T Pinker and Yves Govaerts, (2009) Cloud Variability over the Indian Monsoon Region as Observed from Satellites, Journal of Applied Meteorology and Climatology, 48, 1803-1821

[7] Premaratne H. L. and Bigun J., (2002) Recognition of Printed Sinhala Characters Using Linear Symmetry, The 5th Asian Conference on Computer Vision, Melbourne, Australia, 23 - 25 January

[8] Ramanayake S., Premaratne H. L., (2015) A Study of Cloud Variation on the Commencement of South-West Monsoon around the Sri Lankan Region, International Conference on Advances in ICT for Emerging Regions (ICTer), 31-35

[9] Rathnayake V. S., Premaratne H. L., and Sonnadara D. U. J., (2011) Performance of Neural Networks in Forecasting Short Range Occurrence of Rainfall, Journal of the National Science Foundation of Sri Lanka, 39(3), 251-260

[10] Sikka, D. R., and Sulochana Gadgil, (1980) On the Maximum Cloud Zone and the ITCZ over Indian Longitudes during the Southwest monsoon, Monthly Weather Review, 1840-1853, 1980

[11] Tsing Change Chen and Jau-Ming Chen, (1995) An Observational Study of the South China Sea Monsoon during the 1979 Summer: Onset and Life Cycle, Monthly Weather Review, 123, 2295-2318

[12] Tsing Change Chen and Ming Cheng Yen, (1994) Interannual Variation of the Indian Monsoon Simulated by the NCAR Community Climate Model: Effect of the Tropical Pasific SST, Journal of Climate, 7, 1403-1415

[13] Tsing Change Chen, Ren Yow Tzeng and Ming Cheng Yen, (1988) Development and Life Cycle of the Indian Monsoon: Effect of the 30-50 Day Oscillation, Monthly Weather Review, 116, 2183-2199

[14] Wang B., Linho, Yongsheng Zhang, and Lu M. M., (2004) Definition of South China Sea Monsoon Onset and Commencement of the East Asia Monsoon, Journal of Climate, 17, 699-710

[15] Weerasinghe H. D. P., Premaratne H. L., and Sonnadara D. U. J., (2010) Performance of Neural Networks in Forecasting Daily Precipitation Using Multiple Sources, Journal of the National Science Foundation of Sri Lanka, 38(3), 163-170

Android-Based Legal Assistance Application Using Rule-based Inference Engine

[1]Teddie A. Custodio, [2]Benilda Eleonor V. Comendador
Polytechnic University of the Philippines, Sta.Mesa, Manila
E-mail: teddiecustodio@yahoo.com.ph

ABSTRACT

Criminal Justice System plays a vital role in maintaining peace and order within a society. It provides an evenhanded framework which dictates how criminality or lawlessness should be handled. However due to some factors like cost and personal awareness of people, justice system is somehow compromised. This study's ultimate goal is to develop an application which could be used by people to increase their personal knowledge on criminal law. This android-based application is a product of integrating expert systems into legal disciplines to provide assistance in understanding all underlying provisions under criminal law. There were surveys and interviews which were conducted on this paper to gather user's feedback on the developed application as well as their suggestions on how it could be possibly improved. Related literatures and studies coming from different resources were also cited to support the foundation of this study. The application's main features had focused on consultancy, crime reporting as well as review of the provisions covered by this study. This can be installed and used in any device running on an android operating system.

Keywords - Legal Expert System, Assistance Application, Rule-based Inference Engine

I. INTRODUCTION

Law is a communal rule every human being must abide with to intend his acts. This is a set of knowledge particular on every country to define what is statutory. It is considered complex for its interconnected subsystems of rules which may not be typical for common people. Thus, expert assistance from lawyers are needed to interpret and assert one's right in accordance to law. Law has various fields and one of which is the most common context of day- to-day news- criminal law. Criminal law is concerned with the punishment of all behaviors against human rights which relates to crime. It defines all sanctions and parameters needed to explicitly define any act of criminality against human.

Expert system, on the other hand, is a product of information technology which uses knowledge base and interconnecting rules to imitate human judgment and expertise. It uses artificial intelligence to analyze knowledge and make decisions in the same manner with human. It has the promising capability to leverage time, efficiency and effectiveness. These two different disciplines have a common denominator which an inference is extracted from facts and other relevant information relating to the subject matter. Thus, it can be considered that expert systems could be a great tool to help strengthen the

current criminal system of the country. It can be a medium to allow everyone have an accessible information and tool that they can use to reserve and assert their rights at the comforts of using their handheld devices.

Considering those aforementioned, this study primarily aims to develop an android-based Legal Assistance Application to emulate domain expert in the field of criminal law. However due to some project constraints, the study had focused only on the two chapters from Title Eight of the Book Two of

Criminal Law - ―Crimes Against Persons‖: Chapter 1

Destruction of Life and Chapter 2 - Physical Injuries. The resulting application is expected to implement artificial intelligence to draw inference from human inputs. These inputs are usually keywords that describe criminal offenses which will be later analyzed to identify what specific article(s) of law has been violated. Facts and other related information was added to the application to briefly discuss the criminal liability identified. Proper filing process was explicitly incorporated to guide users on the necessary actions which need to be taken. Also, the application was implemented in android-based mobile devices to make it more accessible and utile to common people.

This study could be a stepping stone in proving that expert systems can be applied and utilized in the current system of Philippine Criminal Law. Not only those common people can use the application but law enforcers as well because they can use this to quickly assist their law-related inquiries which traditionally needs consultation from those law experts. This study doesn't aim to replace criminal case lawyers but it aspires to become a helping tool for those law practitioners. This study is looking into the future of incorporating expert system to the other branches of law which were not included in this current study.

II. METHODS

The author used descriptive research to identify all important facts relating to the developed mobile assistance application for Philippine Criminal Law. In this type, interviews, review of related literatures and surveys were conducted to address the necessary data needed in sustaining the substance of this study. Data gathered includes but not limited to the factors considered in human decision-making, the classified information need to be encoded, the characteristics and function alities the application must inherit and the users' feedback as well. Those data were collected from a group which is composed of 35 students taking up Juris Doctor at Polytechnic University of the Philippines. The sample size was taken from the overall population of the said group which is 38 and with a margin of error set to 5%; determined through the use of Slovin's Formula which can be computed as shown below:

n = N / (1+Ne2)

Where:

n = Number of Samples N = Total Population

e = Margin of Error

By then, each respondent was requested to rate the developed application by answering a survey questionnaire. The ratings are based on Likert metric scale which uses five different levels to quantify the respondent's feedback on the application. Theselevels are shown on the following table.

III. LIKERT SCALE

All the results gathered were treated statistically by using Weighted Mean. Weighted mean is a kind of average where the sum of all scores are divided by the number of sources. The formula is shown below:

$$\mu = (\Sigma Xi) / N$$

Where:

μ = Population Mean

N = Number of Sources

Xi = Number of Occurrences

System Architecture

The developed Legal Assistance Application is made- up of interconnected components which can be shown in Figure 2 such as user interface, inference engine, and knowledge base. It is driven by different entities which either feeds-in or process information inside the system: target user, knowledge engineer and domain expert. Each entity or component has its own roles which are further explained in Figure 3.

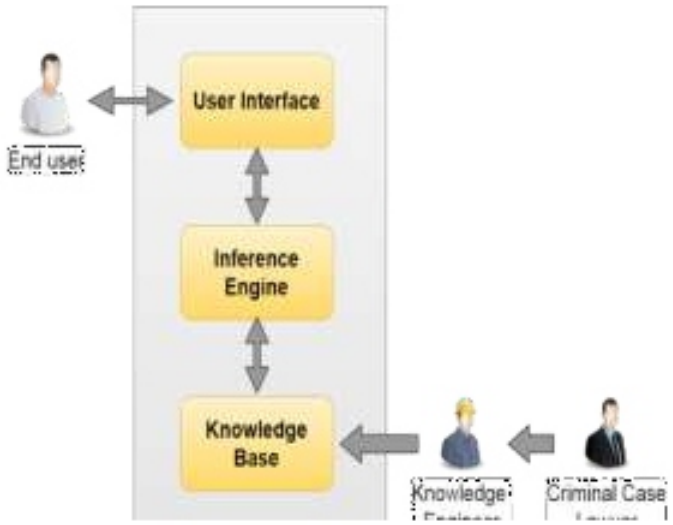

Figure 2. System Architecture of The Android-based Legal Assistance Application Using Rule-based Inference Engine

The developed application consists of three system components namely: user interface, inference engine, and knowledge base. There are key roles or entities which need to be considered as part of the system architecture as shown in Figure 2:

Figure 3 shows the detailed structure of the developed Legal Assistance Application.

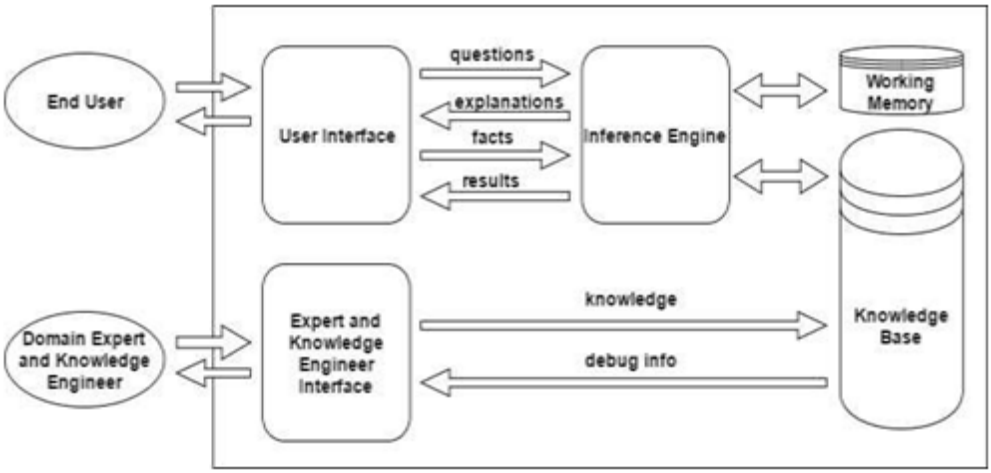

Figure 3. Detailed System Architecture of The Android-based Legal Assistance Application Using Rule-based Inference Engine

End User. Represents the target user of the developed application. Common people or those who do not have expert knowledge in the legal field are referred to as the end user of the application. From this entity where answers will be derived until a certain conclusion is reached. Domain Expert. I trefers to the Criminal Case lawyers who will be acting as validator of the facts and generated results of the application. The provisions uploaded on the application as well as the conclusions generated are the subject of validation. The facts which were validated are stored in the application's knowledge base through Knowledge Engineers.

Knowledge Engineer. Serves as the middleman between the domain expert and the developed application. This role is directly responsible in the encoding of articles, rules, and other related legal information stored on the application's knowledge base. System Interface. It is the system component that accepts input from users and knowledge engineers. It displays the provisions, conclusions generated and other related information encoded on the application. Generally, it serves as the middleware between the system and the outside entities.

Working Memory. Refers to the active recurring memory used during the process of inferencing. It usually holds the deductions of rules until the goal is reached.

Knowledge Base. Stores the articles, rules and other legal-related information used by the application. This is the primary storage where the Inference Engine will query and process rules.

The developed application used a rule-based inference engine which utilizes forward chaining reasoning. Inside the rule-based inference engine are the set of rules which were crafted from the legal provisions and which are uploaded on the knowledge base. Forward chaining on the other hand, allows the inference engine to continue working on data from user inputs and compare it on the rules stored in the working memory. During this process, deduction of rules happens in the memory until the final decision is made. Figure 4 shows the inference cycle of a Rule-based Inference Engine.

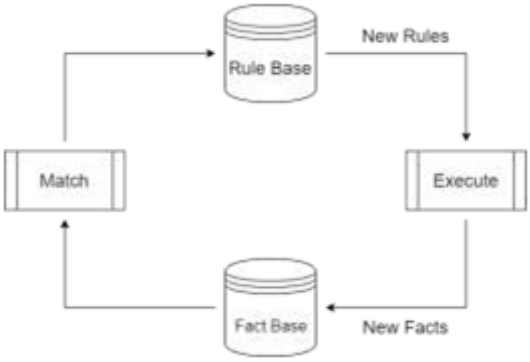

Figure 4. Inference Cycle of a Rule-based Inference Engine

Rule Base. It refers to the Knowledge Base of the developed application. This component is responsible for two main tasks: first, to identify what questions would be displayed next and second, to identify if a conclusion would be generated. Rule-base is triggered to execute ―new rules‖ whenever the user select an option on each question displayed. The option selected would be considered then as ―new facts‖. It will be searched into the Fact Base. Fact Base. This is the Working Memory of the developed application. New facts fed-in by the users are considered the antecedents of rules which are necessary to find what rule would be fired next. On the application context, whenever user selects an option from each question, all the remaining rules inside the Working Memory would be matched with the fact entered by the user. If the fact matches the antecedent of a certain rule then a new rule would be fired else, non-related rules would be removed from the working memory which are technically called as deductions. The inference cycle will continue until one single rule in the working memory would point into a definite conclusion. By that time, only one rule is expected to be found existing in the working memory.

IV. RESULTS

The following results were obtained from the above- mentioned group of respondents using survey and guided interview:

1. The Level of Acceptance of the Respondents in the Developed Android-based Legal Assistance Application Using Rule-based Inference Engine.

The level of acceptance was determined to get the assessments of the respondents on the parameters set for the developed application. These are functionality, usability and reliability which were presented on tables 1, 2 and 3 respectively. To obtain these data, the researcher handover the application to the respondents and let them use it to answer the questionnaires set on the survey. Preliminary demos were also conducted as their walkthrough on the application.

Functionality

Based on the data gathered, application's functionality is Moderately Acceptable‖ to the users with an overall mean of 4.43. It can be also reflected on that the users had highly agreed on the statement that - The application can generate conclusion using minimal number of questions‖. This functionality obtained an average mean of 4.47, interpreted as Moderately Acceptable‖ and ranked as first amongst other functionalities. Ranked second is the quality of questions displayed by the application which avoids ambiguities or confusions with an average mean of 4.44, interpreted as Moderately Acceptable‖. Last is the ability of the application to display logically sequential questions leading to a definite conclusion which got a 4.36 average mean and can be interpreted as Moderately Acceptable‖.

Usability

It was revealed that the respondents' assessment on the usability of the developed application has an overall mean of 4.70 which is interpreted as —Highly Acceptable‖. This level of acceptance means that the application possesses characteristics which makes it more usable to the users. One attribute that stand-out from the others was the clarity of instruction implemented on the application as it received an average mean of 4.83. Followed by the application's easy-to-use options and unambiguous display of questions which got 4.75 and 4.72 average mean respectively. Both interpreted as Highly Acceptable‖. On the fourth rank is the satisfaction of intended users which was scaled 4.69 or Highly Acceptable‖. This was followed by the smooth performance of application in android environment which obtained a scale of 4.67 or Highly Acceptable‖ also. Color schemes and comprehensive conclusions were ranked last which received 4.61 average mean but still considered as Highly Acceptable‖.

Reliability

It shows the respondents' assessment on how reliable the developed Legal Assistance Application, as gathered, received 4.40 grand mean which is interpreted as Moderately Acceptable‖. It shows the validity of the developed application with regards to its output. The first on rank which obtained the highest mean of 4.44 or Moderately Acceptable‖ is the reliability of facts and other information encoded on the application. Second is the consistency of the application on generating conclusions which was rated 4.42 or Moderately Acceptable‖. Next is the relevance of questions to the subject matter which got 4.39 or Moderately Acceptable‖. Lastly, the reliability of conclusions generated based on the parameters set by user. It was rated 4.36 or Moderately Acceptable‖.

DISCUSSIONS

Based on the data gathered, the results showed the following:

1. The developed Legal Assistance Application has various features which were assessed in terms of functionality, usability and reliability. With a total grand mean of 4.51, the respondents' level of acceptance on those feature is Highly Acceptable‖. It shows that the respondents are highly agreed for the overall designed, functionalities and purpose of the application. In terms of Functionality, the respondents have a Moderate Acceptance‖ on the way how the application generates conclusions and carry out other features. It received a total mean of 4.43. In terms of Usability, a total mean of 4.70 or verbally interpreted as Highly Acceptable‖ was obtained from the responses. It means that the respondents assessed the applications as indeed usable through its contained information, graphical user interface, purpose and ease of use. For Reliability, responses showed that the application is generating an accurate and reliable conclusions as tested using some parameters. Also, it displays valid information which user can always rely on. Reliability got a total mean of 4.40 or verbally interpreted as Moderately Acceptable‖.
2. Respondents had leave positive feedback as they evaluate the generated result of the application. All parameters which were asked like proper sequencing of inquiries, accuracy of result and speed of generating conclusions had received equally sound responses.
3. From the gathered responses, most of the respondents had suggested to include more chapters on the current application to cater more provisions. This will make the application more usable to the target users and precise as an actual human expert.

CONCLUSIONS

Based on the findings, the researcher came-up with the following conclusions:

1. The developed Android-based Legal Assistance Application has features and performance which make it highly acceptable for users. Respondents perceived that the application is functional, usable and reliable in generating legal conclusions as well reports and other relevant information necessary for the users.
2. Based on the positive feedback of the respondents, the results generated can be concluded as highly acceptable as this evaluation involved comparison of the result generated by the application and by the human expert. Moreover, this sort of validation from expert respondents had added more confident in using the application as legal assistance tool.
3. The developed application can be concluded as user-friendly because most of the responses showed highly acceptable rating on this attribute. This includes easy to understand questions, clear instructions and appealing graphical user- interface.

RECOMMENDATIONS

The researcher would like to recommend the following based on the findings and conclusions made:

1. Improve sequencing of questions by exploring and comparing other reasoning methods like Backward Chaining.
2. Add additional provisions as rules on the Knowledge Base to provide more mitigating and aggravating factors on the case.
3. Simplify further the words used to allow laymen to easily understand the questions displayed and the conclusions generated.

REFERENCES

[1] Laudon, K., &Laudon, J. (2014). LEARNING TRACK 2: THE EXPERT SYSTEMS INFERENCE ENGINE. In Essentials of Business Information Systems - Seventh Edition.

[2] Al-Ajlan, A. (2015). The Comparison between Forward and Backward Chaining. International Journal of Machine Learning and Computing.

[3] Avdeenko, T., Makarova, E. (2016). Integration Of Case- Based And Rule-Based Reasoning Through Fuzzy Inference In Decision Support Systems. XIIth International Symposium Intelligent Systems

[4] Baghel, A., & Sharma, T. (2013). Survey on Fuzzy Expert System. International Journal of Emerging Technology and Advanced Engineering.

[5] Balajadia, J. (n.d.). Criminal Law Procedure. Philippine Law Journal, 940-945.

[6] Basu, J. K. (2010). Use of Artificial Neural Network in Pattern Recognition . International Journal of Software Engineering and Its Applications.[7] Comendador, B. e. (2013). ATTORNEY 209: A Virtual Assistant Adviser for Family-Based Cases. Journal of Automation and Control Engineering.

[8] Conant, G., & Khan, A. (2002). Legal Expert Systems.

[9]Designing Rule Bases. (n.d.). Retrieved from: http://www.billbreitmayer.com/-rule_based_systems/rule_base d_design.html

[10] Grosan, C., &Abrahama, A. (2011). Rule-Based Expert Systems, pp. 149-185.

[11]Erdani, Y. (2012). Developing Backward Chaining Algorithm of Inference Engine in Ternary Grid Expert System. International Journal of Advanced Computer Science and Applications.

[12] Kaimal, L. B., Metkar, A., & G, R. (2014). SELF LEARNING REAL TIME EXPERT SYSTEM. International Journal on Soft Computing, Artificial Intelligence and Applications.

[13] Kalinauskas, M. (2011). Problematic Aspects of the Use of Expert Systems in Law. Social Technologies Research Journal.

[14] Kumar, S., & Prasad, R. (2015). Importance of Expert System Shell in Development of Expert System. INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT, 128-133.

[15] Maind, S. (2014). Research Paper on Basic of Artificial Neural Network. International Journal on Recent and Innovation Trends in Computing and Communication .

[16] Mansar, S. L., &Marir, F. (2008). Case-Based Reasoning as a Technique for Knowledge Management in Business Process Redesign . Electronic Journal on Knowledge Management, 113-124.

[17] Naik, V. M., &Lokhanday, S. (2012). BUILDING A LEGAL EXPERT SYSTEM FOR LEGAL REASONING IN SPECIFIC DOMAIN-A SURVEY. International Journal of Computer Science & Information Technology (IJCSIT) .

[18] Ogu, E., &Adekunle, Y. (2013). Basic Concepts of Expert System Shells and an Efficient Model for Knowledge Acquisition. International Journal of Science and Research (IJSR).

[19] Swain, M. (2013). Knowledge Engineering. In Encyclopedia of Systems Biology (p. 1074). Springer New York.

[20] Swati, G., &Singhal, R. (2013). Fundamentals and Characteristics of an Expert System. International Journal on Recent and Innovation Trends in Computing and Communication, 110-113.

[21] Grosan, C., & Abraham, A. (2011). Rule-Based Expert Systems. In Intelligent Systems (pp. 149-185). Springer Berlin Heidelberg.
[22] Pal, K., & Campbell, J. (1998). ASHSD-II: A computational Model for Litigation Support.
[23] Philippine Revised Penal Code (Act No. 3815 of December 8). (n.d.). Retrieved from wipo: http://www.wipo.int/wipolex/en/text.jsp?file_id=225306#Lin kTarget_718
[24] Popple, J. (1996). A Pragmatic Legal Expert System.
[25] Rostain, T. (2014). Designing Legal Expert System in the Classroom. Washington D.C, United States of America.
[26] Sanders, K. (2001). CHIRON: Planning in an Open-textured Domain.
[27] Winiwarter, W. e. (2005). Legal Expert System KONTERM-Automatic Representation of Document Structure and Contents. In Database and Expert Systems Applications (pp. 486-497). Springer Berlin Heidelberg.
[28] Zeleznikow, J. (2004). The Split-up Project: Induction, Context and Knowledge Discovery in Law. Law, Probability and Risk, pp. 147-168.
[29](n.d.). Retrieved from Expertise2 Go.com:http://www.expertise2go.com/e2g3g/e2g3gdoc/-e2gmod7.htm
[30](n.d.).Retrieved from Miriam-Webster Dictionary: https://www.merriamwebster.com/dictionary/-jurisprudence
[31] Forward Chaining. (n.d.). Retrieved from http://research.omicsgroup.org/index.php/-Forward_chaining
[32] Hashmi, K. (n.d.). Theories of Jurisprudence – What is the Study of Law? Retrieved from Right for Education: http://www.rightforeducation.org/all-topics/law- rights/jurisprudence-study-of-law/
[33] Inference Engine. (n.d.). Retrieved from PC Magazine: httLpe:/n/gwywelw, L.p.c(m20ag1.5c)o. mV/aelnidcaytcinlogpReduilae/-tberamse/d44A9l1g7o/riinthfemresn, c5e9--75 engine
[34] Smartphone OS Market Share, 2016 Q2. (2016, August). Retrieved from International Data Corporation: https://www.idc.com/prodserv/smartphone-os-market- share.jsp
[35] Temchenko, E. (2016, June). Criminal Law. Retrieved from Legal Information Institute: www.law.cornell.edu
[36] Temchenko, E. (2016, June). Criminal Law. Retrieved from Legal Information Institute: www.law.cornell.edu
[37] Zwass, V. (2016, February 10). Expert System. Retrieved from Britannica: https://www.britannica.com/technology/expert- system#ref705581

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).

2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);

3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);

4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Address of the Editorial Office:

Enriched Publications Pvt. Ltd.
S-9,IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR -5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-45525005