An Efficient Approach of Deep Learning for Android Malware Detection

Mujaddad Sheraz Ahmad[1], Danish Javeed[2*], Muhammad Shoaib[3], Naqash Younas[4], and Adil Zaman[5]

[1,2]Software College, Northeastern University, Shenyang 110169, China

[3,5]Abdul Wali Khan University, Mardan 23200, Pakistan

[4]Dalian University of Technology, Dalian 116024, China

Email: [1]mujaddad@live.com and [2]thedanishkhn@gmail.com*

***Abstract*—** Android plays a very important role in the development of mobile technology as it is one of the famous operating system in smartphones. Its popularity makes it a target of different cyber-attacks that result in money loss and data loss. There is a severe need to protect the android operating system from such attacks. This paper implements a detection system using efficient DL algorithms (i.e., LSTM and BLSTM) by employing latest publically available "CICAndMal2017" dataset in order to protect the android systems against numerous attacks. Further, it uses standard evaluation metrics for the measurement of system's performance. Finally, this paper aims to compare the results with the current state-of-the-art detection techniques to show the efficiency of the proposed model.

***Keywords*—** Android, Threat Detection, Deep Learning, IDS.

I. INTRODUCTION

The mobile phone has become an important part of our daily life since mobile apps provide a lot of services. They changed the way of communication, as they provide a lot of applications in smart devices. Smart devices contain micro-phones, cameras, and sophisticated sensors [1]. For users, a whole new world has been opened due to these sensors and it generates a huge amount of sensitive information. Android is one of the famous operating systems and Google Play plays an important role in its popularity. The popularity of android makes it a target for attacks, which may result in confidential data loss and leads to a serious threat to user's privacy and security. Consequently, this raises a serious need for security solution and malware detection techniques to protect the user from malicious applications, that leads to the exploitation of sensitive data of the user. However, the IoT systems also become equally important as they are not only installed in smartphones but also used in critical systems i.e., industrial IoT devices [2]. According to numerous reports, the number of devices of IoT will be more than 16 billion by the end of the year 2021. The security solutions should not only defend the smartphones from malicious attacks but should defend the IoT devices too from such attacks. Android is having about 85 percent shares in the mobile industry, making it the biggest operating system. In mobile platforms, the detection of malware is a serious problem. Due to new applications, it is very hard to manually examine each of the applications for malicious behavior. Thus, there is a serious need for an automatic detection technique for the detection of malware activities. Machine learning plays a very important role in automatic malware detection. Different detection mechanisms have been proposed by different authors [3, 4]. Convolutional neural networks have also shown very good performance on different natural language processing tasks [5]. Machine learn-ing consists of numerous research areas. Deep Learning is one of the most widely used research areas in machine learning and due to its increasing demand, it has gained serious attention as well as motivated a high number of applications related to image processing as well as speech recognition [6]. The deep learning-based security solutions show high efficiency, as well as accuracy in many cases of the android platform, which is an attractive target to attackers due to its rapid growth. Using Cu-LSTM-enabled DL classifier, we effectively perform detection of several DDoS attacks targeting android networks, our dataset includes diverse categories of android attacks for the assessment of the proposed experimentation.

II. MOTIVATION

Android infrastructure is facing numerous cyber threats and attacks i.e., adware, ransomware, scareware, and SMS mal- ware, etc. So, there is a severe need of enhancing the protection mechanisms from the numerous evolving malware attacks and threats. The proposed work is cost-effective, scalable, and adaptive detection framework.

A. Contribution

The main contribution of this research work is given as follow:

- The latest and up to date, publically available data set have been employed named "CICAndMal2017".
- A CuLSTM enabled deep learning-based threat detection model has been proposed that is

having the ability to achieve high accuracy as well as low False positive rate.
- Cross-validation has been done by using 3-fold cross-validation.
- Comparison of model performance with existing state-of-the-art research.

B. Organization

The remaining part of this research article is organized as follows: section 2 comprises background and related work. Section 3 comprises the proposed methodology, dataset description with the explanation. Section 4 comprises Experimental setup and performance evaluation metrics. Results and discussions have been done in section 5. Finally, the conclusion of this research work is discussed in section 6.

III. BACKGROUND AND RELATED WORK

This section comprises the necessary data that is related to this research work. An artificial neural network is a composition of multiple layers and each of the layer consist of artificial neurons. There are multiple layers in ANN. The first layer of any ANN is the input layer, the output layer is the last, while other layers are known as hidden layers. Current literature is evident that there is a huge increase in android malware. A variety of approaches for the detection of static malware has been manually used for derived features, such as commands, intents, permissions, and API calls, with numerous classifiers [7]. Different malware detection uses static features that are derived from the permissions which have been requested by the application. NLP and static malware analysis have too much in common; therefore, CNN technique can be used for the detection of malware in android. Numerous approaches to neural architectures are widely used for the detection of malware. Different authors proposed multiple methods based on learning [9]. The deep learning techniques can be used to detect malware in various platforms i.e., IoT's, fog to things, etc. [15]. The difference between the previous literature and this proposed work is that previous literature used virtual machines for the capturing of behavioral features. However, some other existing literature used hand-design methods for malware detection [10]. Some approaches monitor the power usage of an android device [11] and send a report if it finds anomalous consumption in the battery. However, some other monitors the calls and report unusual patterns. The general area of malware detection hosts a wider range of detecting approaches i.e., comparing programs based on the code of a program to look for signatures [12], using approaches of data mining as well as machine learning for the detection of malware [13], However, byte string trigrams are also used for the detection of boot sector viruses. In [14], the authors proposed the use of permission behavior for the detection of the new android malware, and for the detection of unknown malware, heuristic filtering is applied. This method resolves the problem of lacking the ability of unknown malware detection. All the existing methodologies and approaches have advanced the detection of malware in android, but these detection mechanisms are not adaptive to new malware and always in need of signature updates. The proposed work is motivated by the existing mechanisms, and to deliver a DL driven architecture for malware detection in the android platform.

IV. METHODOLOGY

A. Proposed Hybrid DL-driven architecture

The proposed deep learning driven architecture consists of BLSTM and CuLSTM for the detection of malwares in android platform. The proposed architecture is shown in the figure 1. The aim of this mechanism is to detect a higher accuracy as well as reduced false rates. The description of the proposed system model can be found in table 2.

B. Dataset Description

Dataset plays a very important role in the performance of any detection system. In this research, a publically available dataset "CICAndMal2017" has been used [8]. The malware of this dataset has been further divided into four categories i.e., Adware, Ransomware, Scareware, and SMS malware. Furthermore, every category of this dataset contains different attacks. The explanation of the dataset is shown in table 1.

C. Dataset Preprocessing

As the dataset comprises blanks, nans, etc., firstly, all these blanks from the dataset have been dropped using data transformation because these blanks can impact the quality of the data.

As the label comprises non-numeric data, all the non-numeric data has been transformed to numeric values. Hot decoding has been done, as the numeric order can impact the results which may lead to unexpected results.

D. Sampling Dataset

The dataset comprises multiple attacks, however, this re-search work labeled the dataset with 8 attacks from the four main categories (i.e., Adware, Ransomware, Scareware, SMS malware) as shown in table 1.

The imbalance problem has also been observed in the dataset. The total number of records in the dataset is huge, it has been sampled to 41,223 (i.e., attack and normal) for better detection accuracy.

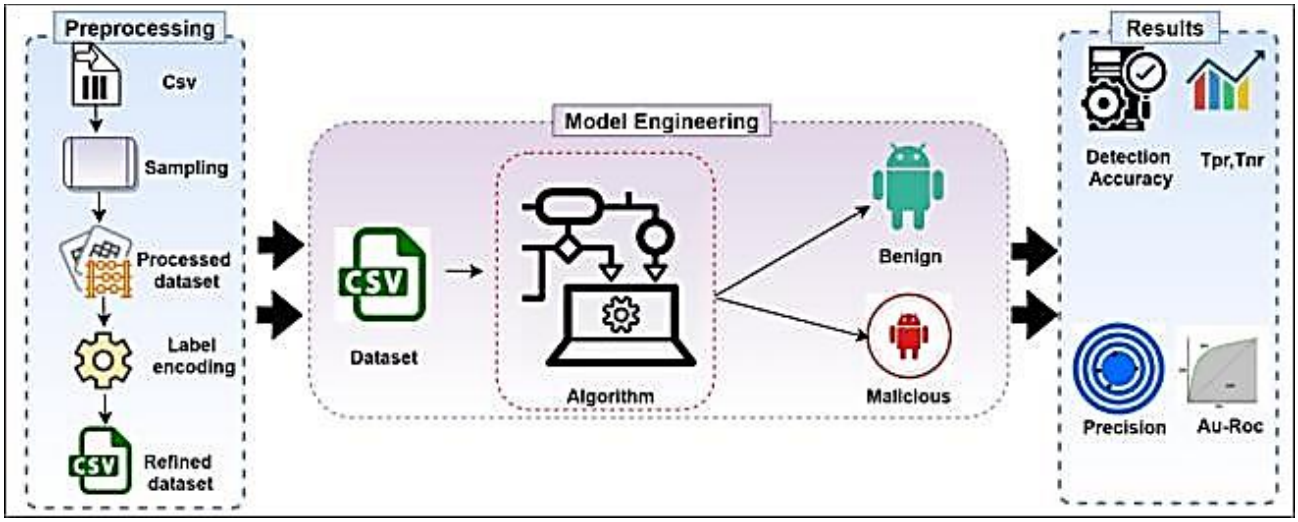

Fig. 1: Proposed Hybrid Detection Model

Table 1: Dataset Description

CATEGORIES	ATTACK TYPES
Adware	Dowgin, Ewind, Feiwo, Gooligan,Youmi, Shuanet, Kemoge, koodous, Mobidash, Selfmite.
Ransomware	Charger family, Koler family, Simplocker family, PornDroid, Svpeng, Jisut, LockerPin, Pletor, RansomBO, WannaLocker.
Scareware	AndroidDefender, AndroidDefender, AVpass, FakeApp.AL, FakeJobOffer, Penetho , AndroidSpy.277, AVpass, FakeApp.AL , FakeTaoBao, VirusShield.
SMS Malware	BeanBot, FakeInst, FakeNotify, Mazarbot, Plankton, Zsone, Biige , FakeMart, Jifake , Nandrobox, SMSsniffer.

V. EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION MATRICS

A. Experimental Setup

The proposed model has been trained using the 3.8 version of python. For fast matrix multiplication and parallel process-ing, the PC has been configured with GPU. For experimental purposes, a single PC server has been used having a core i5-7700 CPU with a 2.21GHz processor and 12 GB of RAM as well as a graphic card of Nvidia Getforce GTX 1050.

B. Performance Evaluation Matrix

Standard performance evaluation has been employed i.e., precision, accuracy, F1 score, recall, ROC, training and testing time, True positive (TP), False Positive Rate (FPR), False positive (FP), True negative (TN), False negative (FN), etc.

VI. RESULTS AND DISCUSSION

This section contains the overall results of evaluation matrix and it also contains the well-defined results with graph and fig-ures. For examine the performance of different models, some standard matrix has been selected like accuracy, precision, F1 score, recall, ROC, TP, FPR etc.

For the evaluation of DL proposed models, the performance metrices can be defined as follows, Precision depends upon the reliability of replicated measurements while the Accuracy provides the closeness of measurement to the true value. Recall predicts the positive values in a dataset while the F1 score is mostly used for evaluating the binary systems which shows the positive or negative values.

Table 2. Proposed Model Description

Algorithm	Layers	Neurons/kernel	AF/LF	Optimizer	Epochs	Batch-size
Cu-LSTM	Cu-LSTM (3)	200, 100,80				
	Dropout	-	Relu	Adam	5	64
	Dense (3)	80, 70, 50				
	Output Layer (1)	4	Softmax			
Cu-BLSTM	CuBLSTM (3)	200,100,80	Relu			
	Dropout	-		Adam	5	64
	Dense (3)	80,70,50	Softmax			
	Output Layer (1)	4				

The True Positive (TP) and True Negative (TN) indicates the most correct predicted values while the False Positive (FP) and False Negative (FN) represents the mis-sorted events. The False Positive Rate (FPR) interpret the quantity of instances which was classified in class Y and it belongs to the distinct class while these instances are not exists in this class. That is how, we discuss the experimentation results of our GPU en-abled CICAndMal2017 detection method. We have compared our proposed model with the other two models for showing the favorable results.

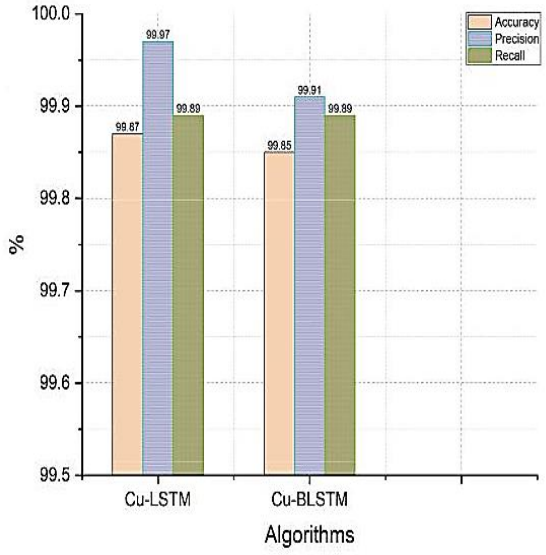

Fig 2: Accuracy of the Models

The confusion matrix is represented by the quadrangle structure consisting of rows and columns; therefore, rows are considered as the true classes of the illustrations while columns are considered as the derived classes. In software engineering, the confusion matrix is used to differentiate the derived values and true values of model components.

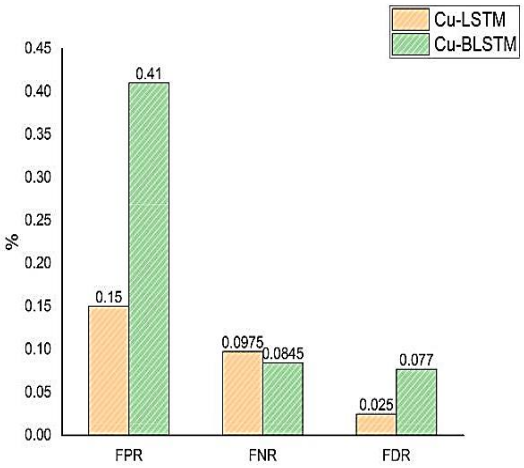

Fig 3. FPR, FNR and FDR

Compliant and non-compliant classes can be classified through four confusion matrix measures including True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Figure 5 and Figure 7 shows the confusion matrix of both algorithms. In Cu-LSTM, we get the higher accuracy rate than in Cu-BLSTM, the accuracy rate is 99.87 % in Cu-LSTM while in Cu-BLSTM the accuracy rate is 99.85%.

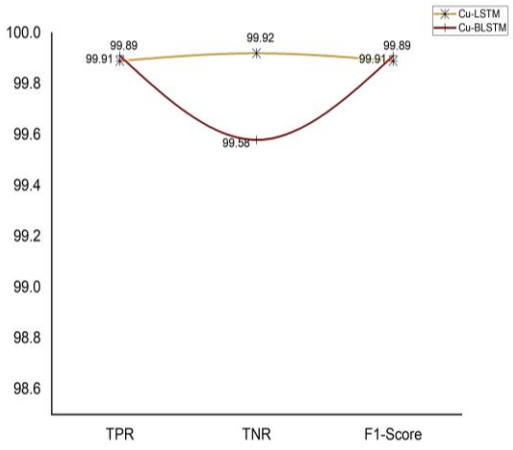

Fig 4: TPR, TNR F1-Score

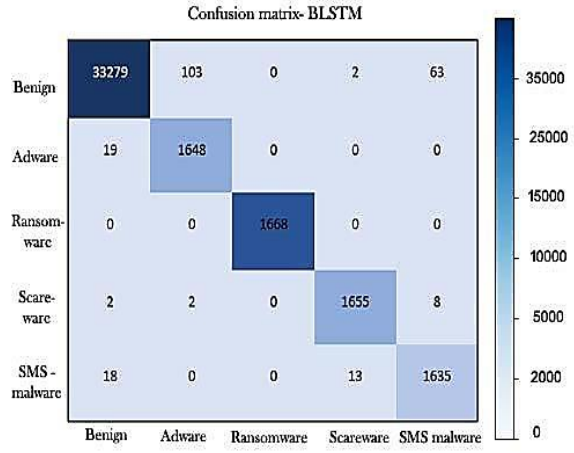

Fig. 5: BLSTM Confusion matrix

The precision rate is also higher in Cu-LSTM, its value is 99.97% while in Cu-BLSTM the precision rate is less containing the value of 99.91%. the recall rate remains same in both algorithms, "Cu-LSTM" and "Cu-BLSTM" showing the value of 99.89% in Figure 2.

In Figure 4, we performed the detailed analysis of our proposed model. The Cu-BLSTM obtained the higher True Positive Rate (TPR) of 99.91% than the Cu-LSTM. While the True Negative Rate (FNR) and F1-score is reached towards the higher rate in Cu-LSTM than Cu-BLSTM by showing the value of 99.92% and 99.91%. These results shows that the Cu-BLSTM is less efficient than the Cu-LSTM.

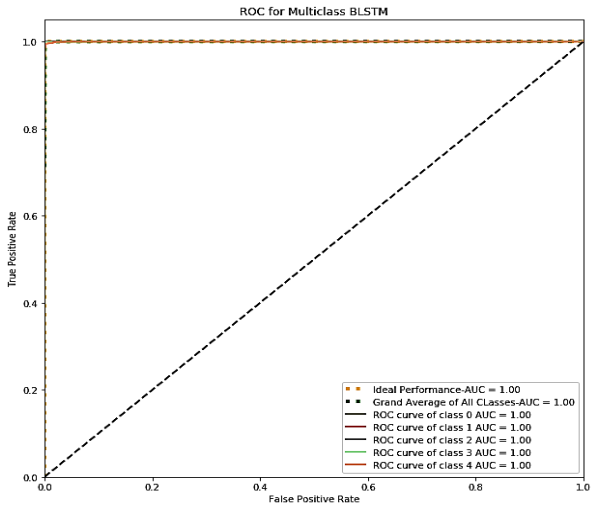

Fig 6: BLSTM Roc

For further evaluating the proposed model, we examined the other metrices which includes the FPR, FNR, False Discovery Rate (FDR) as shown in Figure 3. The measurement of incorrectly categorized positive samples gives the False Negative Rate (FNR). The False Positive Rate (FPR) indicates the ratio between those samples which are incorrectly categorized and the total number of negative samples. The values of FPR, FNR, and FDR obtained by the proposed model are present in the range of 0 to 0.4 which is suitable for the detection system.

The experimental results for both algorithms in this dataset define the effectiveness and favorable performance of this detection model. We ensure the great results of our proposed CICAndMal2017 model by comparing it with other models. Figure 6 and figure 8 comprise the Roc of the proposed algorithms.

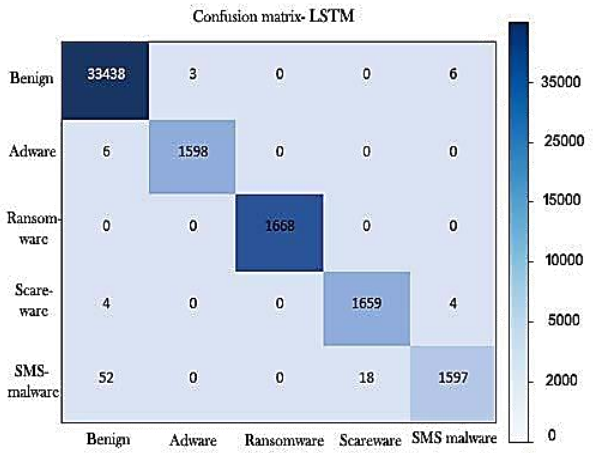

Fig7: LSTM Confsusion Matrix

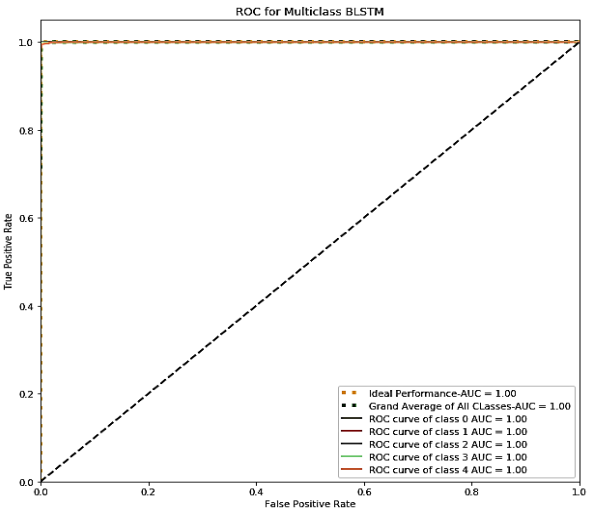

Fig. 8. LSTM Roc

VII. CONCLUSION

Android system demands a reliable, flexible and secure infrastructure. In this paper a DL driven architecture has been proposed to protect the android systems against malwares and cyber-attacks (i.e, adware, ransomware, scareware and SMS malware). Deep learning gained the attention of the world by its advancement. In this paper, two state-of-the-art algorithms, i.e., "LSTM and BLSTM" has been used for the implementation purposes. Both Gpu and Cpu has been used for experimentation purposes, it has been noticed that the speed of Gpu was much faster

than Cpu in terms of testing and training time. The proposed architecture is cost effective as well as high scalable. From implementation results, it has been concluded that the proposed work shows accuracy of 99 percent, that proves the efficiency of our proposed model in terms of detection accuracy. In future, the authors aim to use different datasets as well as different algorithms of deep learning for malware detection.

ACKNOWLEDGMENT

The authors would like to thank the people who helped them and support them through the entire journey. The authors would also like to appreciate the support of the parents and the guidance of friends.

REFERENCES

[1] Delmastro, F., Arnaboldi, V., Conti, M., People-centric computing and communications in smart cities, IEEE Communications Magazine.

[2] Yan, L., Zhang, Y., et al., 2008. The Internet of Things: from RFID to the Nextgeneration Pervasive Networked Systems. Auerbach Publications.

[3] X. Liu and J. Liu. A two-layered permission-based android malware de-tection scheme. In Mobile Cloud Computing, Services and Engineering (MobileCloud), 2014 2nd IEEE Int. Conf. on, pages 142-148, 2014.

[4] A. Sharma and S. K. Dash. Mining api calls and permissions for android malware detection. In Cryptology and Network Security, pages 191-205. 2014..

[5] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional net-works for text classication. In Advances in Neural Information Process-ing Systems, pages 649-657, 2015.

[6] Y. Bengio. Learning deep architectures for ai. Foundations and trends in Machine Learning, 2(1):1–127, 2009.

[7] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck. Drebin: Effective and explainable detection of android malware in your pocket. In NDSS, 2014.

[8] Arash Habibi Lashkari, Andi Fitriah A. Kadir, Laya Taheri, and Ali A. Ghorbani, “Toward Developing a Systematic Approach to Gener-ate Benchmark Android Malware Datasets and Classification”, In the proceedings of the 52nd IEEE International Carnahan Conference on Security Technology (ICCST), Montreal, Quebec, Canada, 2018.

[9] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas. Malware classication with recurrent networks. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE Int. Conf. on, pages 1916-1920, 2015.

[10] J. Saxe and K. Berlin. Deep neural network based malware detection using two dimensional binary program features. In 2015 10th Interna-tional Conference on Malicious and Unwanted Software (MALWARE), pages 11-20, Oct 2015.

[11] Javeed, D., Badamasi, U. M., Iqbal, T., Umar, A., & Ndubuisi, C. O. (2020). Threat Detection using Machine/Deep Learning in IOT Environments. International Journal of Computer Networks and Communications Security, 8(8), 59-65.

[12] [12] Mark A. Davenport, Richard G. Baraniuk, and Clayton D. Scott. Tuning support vector machines for minimax and neyman-pearson classifica-tion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(10), 2010

[13] Matthew G. Schultz, Eleazar Eskin, Erez Zadok, and Salvatore J. Stolfo. Data mining methods for detection of new malicious executables. In Proceedings of the 2001 IEEE Symposium on Security and Privacy, SP ’01, pages 38–, Washington, DC, USA, 2001. IEEE Computer Society.

[14] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets. In Proceedings of the 19th Annual Network & Distributed System Security Symposium, Feb. 2012.

[15] Javeed, Danish, Tianhan Gao, and Muhammad Taimoor Khan. "SDN-Enabled Hybrid DL-Driven Framework for the Detection of Emerging Cyber Threats in IoT." Electronics 10.8 (2021): 918.

[16] Javeed, Danish, et al. "A Hybrid Deep Learning-Driven SDN Enabled Mechanism for Secure Communication in Internet of Things (IoT)." Sensors 21.14 (2021): 4884.