Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Detecting Malware on Android Devices Using CNN-LSTM with FDSWM based Feature Selection

S.V. Vasantha[1], B. Kiranmai[2], B. Jyoshna[3], S. Ramakrishna[4], B. Suvarnamukhi[5], S. Hariharan[6]

[1]*CSE Department, Vardhaman College of Engineering, Hyderabad, India, s.v.vasantha@gmail.com*
[2]*CSE(DS) Department, Sreyas Institute of Engineering and Technology, Hyderabad, India, kiranmaimtech@gmail.com*
[3]*CSE(DS) Department, Sreyas Institute of Engineering and Technology, Hyderabad, India, jyoshnabejjam@gmail.com*
[4]*CSE Department, GSoT, GITAM(Deemed-to-be-University), Hyderabad, India, rsankara@gitam.edu*
[5]*CSE Department, SreyasNP Institute of Engineering and Technology, Hyderabad, India, mukhi.suvarna@gmail.com*
[6]*AI&DS Department, Vardhaman College of Engineering, Hyderabad, India, hariharan.s@vardhaman.org*

The widespread use of Android devices has made them attractive targets for cyberattacks. Traditional methods of detecting malware often struggle to keep up with evolving threats. In this research, a thorough analysis is undertaken to explore the potential contributions of deep learning and machine learning towards improving the detection of Android malware. Deep learning approaches based on GRU, LSTM, and CNN-LSTM networks are evaluated and compared with three machine learning algorithms: KNN, DT, and SVC. Using the CICAndMal2023 dataset, the effectiveness of these methods in distinguishing between malicious and benign Android apps is analyzed. The findings suggest that both deep learning and machine learning hold promise for improving Android malware detection. Specifically, the CNN-LSTM deep learning method demonstrated the highest accuracy of 99.7%. This indicates that deep learning techniques can provide a more reliable means of shielding Android users from evolving malware threats.

Keywords: Android, Malware Detection, Machine Learning, CNN-LSTM, Feature Selection.

1. Introduction

The increasing reliance on smartphones has elevated the Android operating system as a prime target for cyberattacks. Conventional malware detection methods often struggle to keep pace with emerging threats, and many Android users do not install antivirus software in the first place. As a result, Android devices are left vulnerable to malware infiltration, which is primarily due to the presence of malicious applica-tions that manage to infiltrate the Google Play Store, a popular platform for downloading and installing applications on Android devices. The sheer number of malware applications requires more sophisticated malware detection methods. A proposed solution involves a machine learning-based model designed to identify malware in Android applications, examining various aspects related to APKs. This model exhibits promising potential, as indicated by experimental studies conducted on a standard dataset M.P. Singh and H.K. Khan(2023).

Approaches for detecting Android malware commonly categorize into three main types: static analysis, dynamic analysis, and hybrid analysis. In static analysis, fea-tures are extracted from Android applications without actually executing them. Dy-namic analysis involves feature extraction by executing applications on an Android emulator or device. Hybrid analysis integrates aspects of both static and dynamic approaches. For classifying and distinguishing between malicious and legitimate APK programs using machine learning, numerous models exist. These models lev-erage various algorithms and techniques to analyze patterns, behaviors, and features extracted from the APK files. The goal is to develop effective classifiers capable of identifying and flagging potential threats, contributing to enhanced security measures on Android devices. The dynamic landscape of Android malware necessitates continuous advancements in machine learning models to keep pace with evolving threats and ensure robust protection for users. To achieve the superior detection accuracy, this study uses a feature set of 27 attributes from a recently published dataset (CICMalDroid2020) with 18,998 instances of APKs A. Droos et. al.(2022), UNB_CIC dataset(2023), I. B. Mijoya et. al.(2022).

2. Related Work

A machine learning approach for Android malware detection relies on the coexis-tence of static features. According to this approach, malicious Android applications seek a regular set of coexisting permissions and APIs, whilst malicious applications require a different set. An article generated a new dataset of co-existing permissions and API requests at four distinct combination levels—the second, third, fourth, and fifth levels—to validate this assumption A.H.E. Fiky et. Al.(2021). A novel framework for multi-view feature intelligence (MFI) is devised to acquire the representation of desired capabilities B. Tahtaci and B. Casvcay(2020). Semantic string features, API call graph features, and small opcode sequence features are among the multi-view heterogeneous characteristics extractable using this novel framework through reverse engineering E. Odat and Q. M. Yaseen(2023). Through a series of feature analysis, selection, aggregation, and encoding operations, the framework can learn the representation of a targeted capability from existing malware families to identify novel Android malware with shared target capability. This approach outperforms three state-of-the-art techniques—namely, Drebin, MaMaDroid, and N-opcode in identifying unidentified Android malware possessing specified features J. Qiu et. al.(2023).

Identifying malicious Android apps is crucial due to their widespread usage in im-plementing the Internet of Things (IoT). Firstly, an API graph embedding is created using a classifier based on graph neural networks (GNNs) to demonstrate the efficacy of graph-based classification. Secondly, a graph-based GNN Android malware classi-fier-attacking technique called VGAE-MalGAN, based on generative adversarial networks (GANs), is proposed R. Yumlembam et. al.(2023). Due to their extensive functionality and user-friendly nature, Android-based mobile devices have garnered a sizable user base, making them a prominent platform for attackers. This article introduces AMDI-Droid, an enhanced deep neural network that effectively safeguards Android smartphones from fraudulent applications. The enhanced efficacy of the suggested model is confirmed by comprehensive assessments in comparison to cutting-edge methods for accuracy, precision, recall, F1 score, and Matthews correlation coefficient (MCC) metrics P. Musikawan et. al.(2022). Such malware will be analysed using machine learning techniques supplemented by semantic analysis. A set of permissions for harmful programs will be compared with permissions extracted from the application under analysis. Ultimately, users will be able to determine the extent of harmful permissions present in the program, supported by assessments through comments R. Agrawal et. al.(2020). The research delves into various Android malwares and the deep learning techniques they employ for infiltration, alongside antivirus software safeguarding Android systems. Multiple deep learning-based methods for detecting malware on Android were discussed, including Droid Deep Learner, Droid Deep Detector, Deep Flow, Droid Delver, and Droid Deep. The system implemented a deep learning model capable of determining, without installation, whether an Android application is infested with malware or not S. Sabhadiya et. al.(2023).

Addressing the challenges associated with large, high-dimensional datasets, the study proposes a robust malware detection solution employing feature selection and deep learning techniques. The findings underscore the importance of customized strategies for diverse malware datasets, showcasing that certain feature-selected scenarios preserve performance levels comparable to the original dataset Alomari et. al.(2023). Machine learning works well for detecting malware, particularly supervised learning using static features. This work introduces Deep Q-learning-based Feature Selection Architecture (DQFSA), which is adaptable in information security applications and requires less human intervention for feature selection Zhiyang et. al.(2019).

The field of Android malware detection has witnessed significant progress, yet there exists a critical research gap in the integration of comprehensive pre-processing, feature selection, and hybrid deep learning techniques. Current models often focus on singular methodologies, either machine learning (ML) or deep learn-ing (DL), without exploiting the potential synergies that a hybrid approach may of-fer. This gap underscores the need for a holistic and innovative solution to enhance efficacy and precision in combating Android malware Dhabal G, and Gupta G(2022).

3. Proposed System

The increasing cyber threats targeting Android smartphones grow in complexity, conventional malware detection techniques, like signature-based methods, struggle to combat dynamic malware effectively. To meet the need for more adaptable and advanced solutions, this study presents a refined approach employing deep learning techniques. Utilizing deep learning, the proposed methodology strives to enhance the resilience and flexibility of Android malware detection. This approach recognizes the dynamic nature of cyber threats and aims to furnish a proactive defence mechanism capable of promptly identifying and mitigating emerging risks effectively. Fig. 1 illus-trates the procedural sequence for the detection of Android malware.

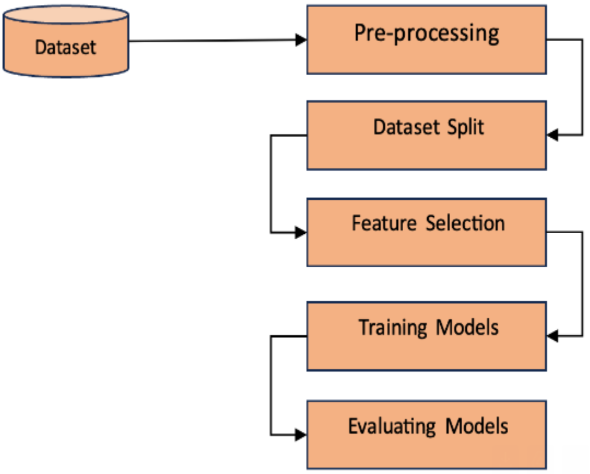

Figure 1. Procedural sequence of proposed Android malware detection system

3.1 Feature Selection based on FDSWM

Frequency Differential Selection and Weight Measurement(FDSWM) is a method aimed at improving feature selection particularly in the context of Android malware detection. It comprises two main stages:

Frequency Differential Selection: Identifies and selects the most informative features based on their frequency differences between benign and malicious data.

Weight Measurement: Assigns importance weights to these features to enhance the accuracy and effectiveness of the machine learning model in detecting malware Sun et. al.(2022), Yang and Tian(2014), Satyanegara et. al.(2022).

Frequency Differential Selection(FDS) Algorithm:

1. Initialization:

a. Let Features={f1,f2,...,fn} be the set of features extracted from the dataset.

b. Let Xb and Xm denote the quantity of benign and malicious programs that have feature fi, respectively.

c. Let Yb and Ym stand for the total number of benign and malicious applications, respectively

2. Calculate Feval:

For each feature fi:

Calculate the frequency difference between the feature in malicious and benign apps:

Feval= $\frac{X_b - X_m}{Y_b + Y_m}$

Where Xb and Xm are the quantities of benign and malicious programs having feature fi, respectively, and Yb and Ym are the total numbers of benign and malicious applications, respectively.

3. Feature Selection:

a. Rank features based on the absolute values of Fevali.

b. Select the topN features with the highest absolute differential frequencies.

Weight Measurement (WM) Algorithm:

1. Initialization:

a. Let Weights={w1,w2,...,wN} be the set of weights assigned to selected features.

b. Initialize wi=1 for all selected features.

2. Optimization:

Iterate through a training process (e.g., gradient descent):

Adjust weights based on feature importance:

wi = Optimize(Δfi)

3. Final Weighting:

The adjusted weights represent the importance of selected features for malware detection.

This algorithm iteratively refines feature selection and weight assignment, ultimately enhancing the effectiveness of ML models for Android malware detection Mahindru et. al.(2024), Neil et. al.(2024).

3.2 CNN-LSTM Model

The hybrid model blends CNN to extract spatial features and LSTM for temporal analysis. By seamlessly integrating CNN outputs into LSTM layers, the model effec-tively captures both spatial nuances and temporal intricacies, enhancing its predictive prowess. Additionally, the inclusion of fully connected layers after another stage of CNN improves classification accuracy, notably for identifying Android malware.

Convolutional Neural Network (CNN) for Spatial Feature Extraction:

In the hybrid model, the CNN component plays a pivotal role in discerning spatial patterns within the input data. When dealing with input sequences, like system or API calls, the aim is to transform them into 1D sequences. This transformation facilitates the utilization of 1D convolutional layer, which are adept at extracting nuanced patterns and distinctive features from the sequences. Additionally, pooling layers are strategi-cally incorporated into the model to streamline computation and diminish the spatial dimensions of the data, thus enhancing efficiency without compromising on pattern recognition accuracy.

Long Short-Term Memory (LSTM) Network for Temporal Analysis:

The LSTM component within the hybrid model shoulders the responsibility of understanding the sequential intricacies inherent in the input data, discerning temporal dependencies crucial for accurate analysis. To capture these temporal relationships effectively, the output derived from the CNN is seamlessly integrated into the LSTM layers. Within these LSTM layers, the focus lies on modeling long-term dependencies and capturing sequential patterns, enabling the model to grasp the underlying temporal dynamics inherent in the data, thereby enhancing its predictive capabilities.

Hybrid CNN-LSTM Model:

In order to equip the model with the capability to en-capsulate both spatial nuances and temporal intricacies within the data, a strategic modification is implemented where the output generated by the CNN layers is seam-lessly integrated into the LSTM layers. This integration ensures that the model can effectively capture and synthesize both spatial and temporal characteristics, thereby enhancing its overall predictive capacity and analytical depth. Subsequently, for the categorization process fully connected layers are introduced after the another stage of CNN layers, facilitating enhanced classification outcome for Android malware detection. The proposed architecture is depicted in Fig 2.

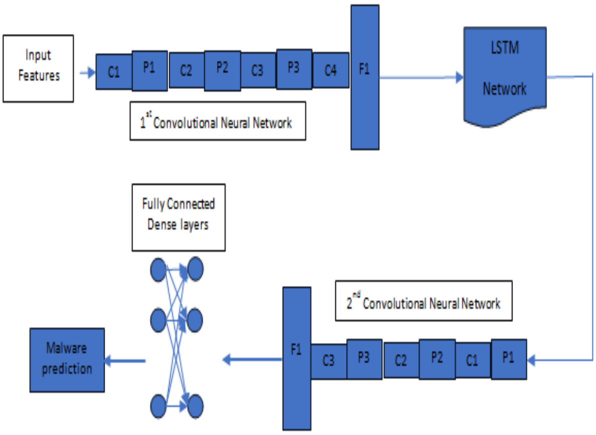
Input Features
C1
P1
C2
P2
C3
P3
C4
F1
1st Convolutional Neural Network
LSTM Network
Fully Connected Dense layers
2nd Convolutional Neural Network
Malware prediction
F1
C3
P3
C2
P2
C1
P1

Figure 2. Proposed CNN-LSTM Model Architecture

4. Results and Comparative Analysis

The study investigated five distinct methodologies for Android malware detection, including Support Vector Classifier (SVC), Decision Tree (DT), K-Nearest Neighbours (KNN), and deep learning techniques utilizing Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Convolutional Neural Networks – Long Short Term Memory (CNN-LSTM) neural networks. Each approach's efficacy was assessed through ten-fold cross-validation to ensure robustness and generalizability. Performance metrics such as accuracy, precision, F-measure, and recall were com-puted and summarized in the Table 1 below, offering a comprehensive evaluation of Android malware detection methodologies. The proposed FDS based CNN-LSTM shown outperforming accuracy of 99.7% over other models.

Table 1. Performance Evaluation of various algorithms.

Approach	Accuracy	F-Measure	Recall	Precision
SVC	0.83	0.81	0.8	0.84
DT	0.85	0.84	0.83	0.86
KNN	0.88	0.87	0.86	0.89
GRU	0.94	0.93	0.92	0.95
LSTM	0.975	0.952	1.0	0.976
CNN- LSTM	0.997	0.97	0.95	1.0

In evaluating the model on the Test data, the dataset was divided into two sub-sets—25% initially and an additional 30% for a more in-depth analysis as shown in Fig. 3 and Fig. 4 respectively. Subsequently, precision, recall, and F1 score metrics were compared to assess the model's predictive performance on the test data across different proportions. In the 25% subset, the model demonstrated exceptional preci-sion at 100%, accompanied by a commendable recall of 94%, resulting in an impres-sive F1-score of 92. However, in the 30% subset, precision remained high at 97%, while recall and the F1-score slightly decreased to 92% and 90%, respectively. These findings highlight the nuanced predictive capabilities of the model across different portions of the test data. Accuracy performance over training and validation phases is plotted in Fig. 5, shows that FDS based CNN-LSTM approach can capture more complex relationships in the data and is well-suitable for identifying hidden malware patterns in Android apps.

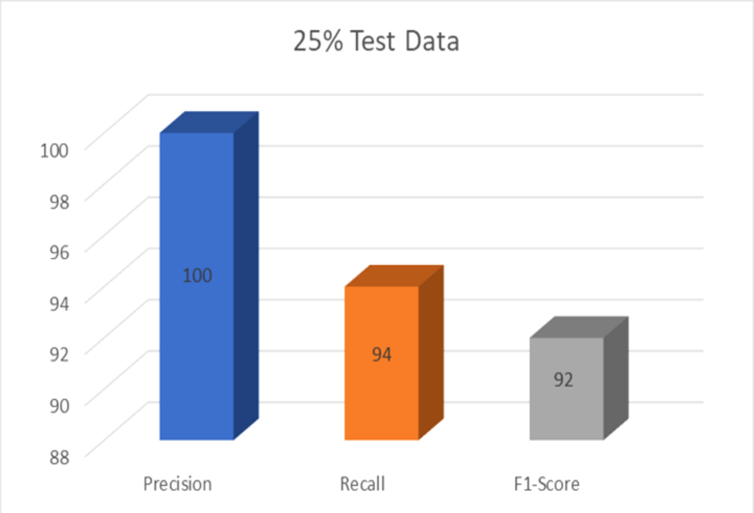

Figure 3. Performance of 25% Test data

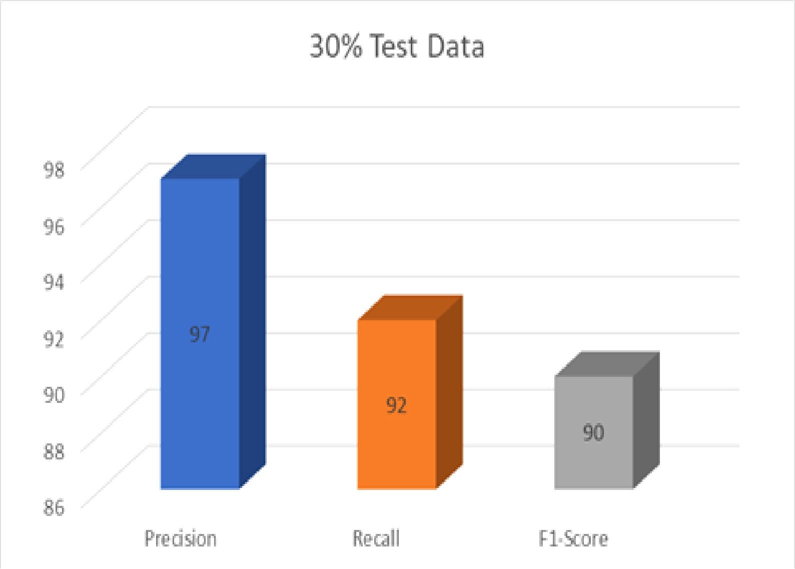

Figure 4. Performance of 30% Test data

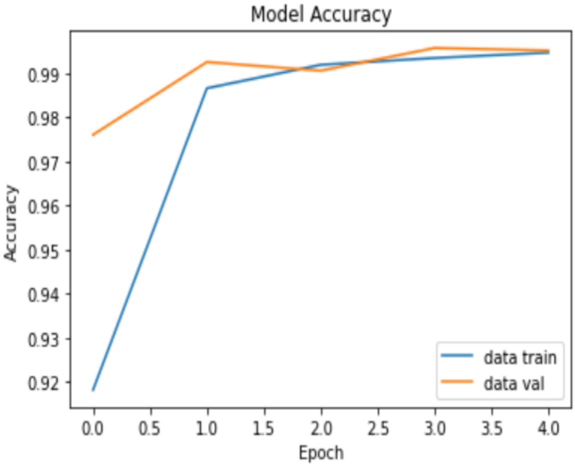

Figure. 5. Training and Validation Curves

5. Conclusion

Traditional machine learning and deep learning methods offer effective means for Android malware detection, with deep learning showing notable superiority. The proposed CNN-LSTM model notably achieved 99.7% accuracy and a perfect F-measure, surpassing both traditional classifiers and other deep learning architectures. Its resilience to varying sequence lengths and capacity to autonomously extract pivotal features from raw data positions CNN-LSTM as an ideal choice for end-to-end learning in Android malware detection. This study underscores deep learning's efficacy in this domain, suggesting CNN-LSTM's potential as a robust tool for fortifying Android security and shielding users from malicious apps. Future research avenues could explore deep learning's applicability to broader aspects of Android security, including vulnerability identification and intrusion prevention.

References

1. A. Droos, A. Al-Mahadeen, T. Al-Harasis, R. Al-Attar and M. Ababneh, "Android Malware Detection Using Machine Learning," 2022 13th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2022, pp. 36-41, doi: 10.1109/ICICS55353.2022.9811130.
2. A. H. E. Fiky, A. Elshenawy and M. A. Madkour, "Detection of Android Malware using Machine Learning," 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2021, pp. 9-16, doi: 10.1109/MIUCC52538.2021.9447661.
3. Alomari, E.S.; Nuiaa, R.R.; Alyasseri, Z.A.A.; Mohammed, H.J.; Sani, N.S.; Esa, M.I.; Musawi, B.A. Malware Detection Using Deep Learning and Correlation-Based Feature Selection. Symmetry 2023, 15, 123. https://doi.org/10.3390/sym15010123
4. B. TAHTACI and B. CASVCAY, "Android Malware Detection Using Machine Learning," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), IstaSVCul, Turkey, 2020, pp. 1-6, doi: 10.1109/ASYU50717.2020.9259834.
5. Dhabal, G., & Gupta, G. (2022). Towards Design of a Novel Android Malware Detection Framework Using Hybrid Deep Learning Techniques. In Soft Computing for Security Applications: Proceedings of ICSCS 2022 (pp. 181-193). Singapore: Springer Nature Singapore.
6. E. Odat and Q. M. Yaseen, "A Novel Machine Learning Approach for Android Malware Detection Based on the Co-Existence of Features," in IEEE Access, vol. 11, pp. 15471-15484, 2023, doi: 10.1109/ACCESS.2023.3244656.
7. https://www.unb.ca/cic/datasets
8. I. B. Mijoya, S. Khurana and N. Gupta, "Malware detection in Android devices Using Machine Learning," 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 307-312, doi: 10.1109/ICCCIS56430.2022.10037699.
9. J. Qiu et al., "Cyber Code Intelligence for Android Malware Detection," in IEEE Transactions on Cybernetics, vol. 53, no. 1, pp. 617-627, Jan. 2023, doi: 10.1109/TCYB.2022.3164625.
10. M. P. Singh and H. K. Khan, "Malware Detection in Android Applications Using Machine Learning," 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS), Bangalore, India, 2023, pp. 105-110, doi: 10.1109/ICAECIS58353.2023.10170311.
11. Mahindru, A., Arora, H., Kumar, A., Gupta, S.K., Mahajan, S., Kadry, S. and Kim, J., 2024. PermDroid a framework developed using proposed feature selection approach and machine learning techniques for Android malware detection. Scientific Reports, 14(1), p.10724.
12. Neil AM, Shabaan E, El Qout M, Emara K. Machine Learning Based Approaches For Android Malware Detection using Hybrid Feature Analysis. In2024 6th International Conference on Computing and Informatics (ICCI) 2024 Mar 6 (pp. 158-165). IEEE.
13. P. Musikawan, Y. Kongsorot, I. You and C. So-In, "An Enhanced Deep Learning Neural Network for the Detection and Identification of Android Malware," in IEEE Internet of Things Journal, vol. 10, no. 10, pp. 8560-8577, 15 May15, 2023, doi: 10.1109/JIOT.2022.3194881.

14. R. Agrawal, V. Shah, S. Chavan, G. Gourshete and N. Shaikh, "Android Malware Detection Using Machine Learning," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-4, doi: 10.1109/ic-ETITE47903.2020.491.
15. R. Yumlembam, B. Issac, S. M. Jacob and L. Yang, "IoT-Based Android Malware Detection Using Graph Neural Network With Adversarial Defense," in IEEE Internet of Things Journal, vol. 10, no. 10, pp. 8432-8444, 15 May15, 2023, doi: 10.1109/JIOT.2022.3188583.
16. S. -Q. Yang and X. -D. Tian, "A maintenance algorithm of FDS based mathematical expression index," 2014 International Conference on Machine Learning and Cybernetics, Lanzhou, China, 2014, pp. 888-892, doi: 10.1109/ICMLC.2014.7009727.
17. S. Sabhadiya, J. Barad and J. Gheewala, "Android Malware Detection using Deep Learning," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1254-1260, doi: 10.1109/ICOEI.2019.8862633.
18. Satyanegara, H. H., and K. Ramli. “Implementation of CNN-MLP and CNN-LSTM for MitM Attack Detection System”. Jurnal RESTI (RekayasaSistem Dan TeknologiInformasi), Vol. 6, no. 3, June 2022, pp. 387 -96, doi:10.29207/resti.v6i3.4035.
19. Sun, Huizhong& Xu, Guosheng& Wu, Zhimin& Quan, Ruijie. (2022). Android Malware Detection Based on Feature Selection and Weight Measurement. Intelligent Automation & Soft Computing. 33. 585-600. 10.32604/iasc.2022.023874.
20. Zhiyang, Fang & Wang, Junfeng&Geng, Jiaxuan& Kan, Xuan. (2019). Feature Selection for Malware Detection Based on Reinforcement Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2957429.