A Detection Android Cybercrime Model utilizing Machine Learning Technology

Fahad M. Ghabban
Information System Department, College of Computer Science and Engineering, Taibah University, Madina 42353, Saudi Arabia
fghaban@taibahu.edu.sa (corresponding author)

Received: 9 March 2024 | Revised: 28 March 2024 | Accepted: 30 March 2024

 DOI: https://doi.org/10.48084/etasr.7218

ABSTRACT

The present study developed a Detection Android cybercrime Model (DACM), deploying the design science approach to detect different Android-related cybercrimes. The developed model consists of five stages: problem identification and data collection, data preprocessing and feature extraction, model selection and training, model evaluation and validation, and model deployment and monitoring. Compared to the existing cybercrime detection models on the Android, the developed DACM is comprehensive and covers all the existing detection phases. It provides a robust and effective way to spot cybercrime in the Android ecosystem by following Machine Learning (ML) technology. The model covers all the detection stages that are normally included in similar models, so it provides an integrated and holistic approach to combating cybercrime.

Keywords-detection model; machine learning; android system; design science approach

I. INTRODUCTION

More and more mobile devices become connected to each other through the Internet every day, and their processes are becoming more complex and varied [1]. With technology going forward continuously, people's daily activities shift from the physical to the cyber world. Life is made easier by this transformation, but it also brings a major disadvantage: security. Owing to the unspecified structure of the Internet, cybercriminals can easily hide in the cyber world. Due to their widespread usage, mobile devices are not only a major target of these attackers, but also their local area networks and individual end users. By deploying malicious websites and programs, attackers attempt to exploit the weaknesses of the network or the human user. According to McAfee Labs report on 2020 first-quarter threats, 98% of attackers target Android devices [2].

Android-based detection and classification tools have a wide range of threats, which demands malware analysis techniques that can effectively detect and classify the latter [3]. Malware programs are harmful computer programs, such as worms, backdoors, viruses, spyware, and Trojan horses, which are designed to damage the computer in different ways. Several malware-based attack techniques have been developed to send private information and attack systems (specifically on Android platforms) without the victim's knowledge. The best way to detect these attacks would be to conduct static and dynamic analyses. Figure 1 displays the malware distribution over the past five years [4].

Detection systems in static analysis generally focus on the assets of the software since they investigate both its implementation and its source code and identify possible threats. The signature of the attack will be examined in this analysis, as well as the permissions used by the bytecode targeted by the attack. This type of detection mechanism suffers from a major limitation in that it does not detect zero-day attacks, which have never been seen before by the network [5]. The most effective way of pinpointing this type of malware is dynamic analysis. Dynamic models tend to define the normal behaviors of a system by educating with earlier data transfers or permission requests that correspond to the previous normal behaviors of the system. To catch abnormal behaviors, the system then attempts to identify them by pinpointing their suspicious requests and block them.

Moreover, according to [6], malware, ransomware, phishing, zero-day attacks, and Denial-of-Service (DoS) are some of the most common attacks, as illustrated in Figure 2. Several factors might play a crucial role in the high frequency of these attacks, but one of the most basic reasons is that the defense measures have not been perfected to such an extent to prevent them [7]. Many of today's detection strategies require manual investigation by analysts to detect advanced threats, malicious users' behaviors, and other dangerous behaviors for the detection of these threats [8]. ML, over the course of its development, has proven to be more competent than human intelligence at recognizing and predicting specific patterns. As a result of the highly dynamic, sophisticated network systems that fail to meet security requirements, security decisions, and policy changes that could affect the whole system have been made. ML technology and intelligent decision-making capabilities have permitted the automation of the decision making process.

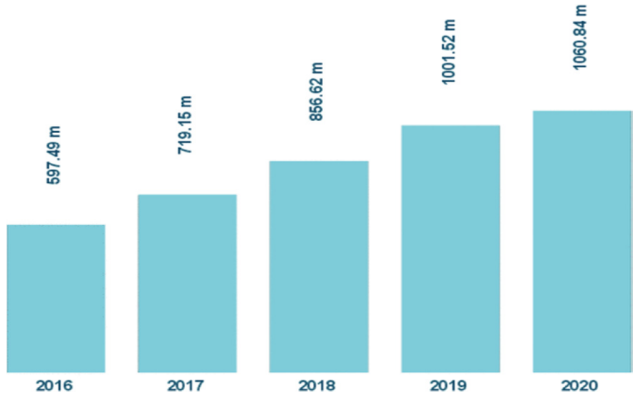

Fig. 1. Malware distribution over the past five years.

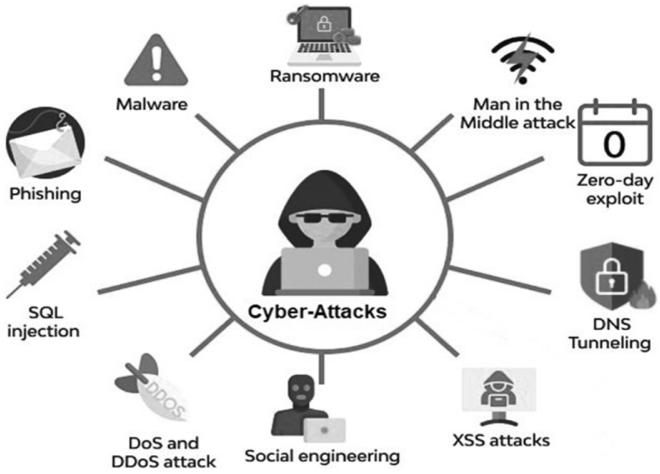

Fig. 2. Common cyberattacks.

The concept of automatic learning and experience upgrading is referred to as Machine Learning (ML). ML can enhance the decision-making process [9]. In recent years, machines have been implemented to perform tasks, such as sentiment analysis, pattern recognition, malware detection, and network intrusions detection [10-12]. Classes of ML algorithms are supervised, unsupervised, semi-supervised, and reinforcement ML algorithms. In supervised ML, the things learned prior to the new data are turned into labels in order to predict upcoming events. As a result, inferred functions can be engaged to predict what outcomes will be generated from the training dataset after analysis. In addition, these algorithms can compare the existing output with the intended output and adjust the model accordingly. Unsupervised ML algorithms, on the other hand, do not train the information using either classifications or labels. They study how unlabeled data can be used to infer the function associated with a hidden structure. Furthermore, the system identifies unlabeled data hidden structures while interpreting inferences drawn from datasets to express them. Semi-supervised ML algorithms apply labels and unlabeled data to train the model. In other words, they stand somewhere in between supervised and unsupervised algorithms and combine the best features of both. Finally, reinforcement ML algorithms interact with their environment. As a consequence, machines and software agents can involuntarily determine the optimal behaviors in a specific context to maximize their performance without being aware of it.

This study proposes a Detection Android cybercrime Model (DACM) deploying ML. The proposed model consists of five main stages: problem identification and data collection, data preprocessing and feature extraction, model selection and training, model evaluation and validation, and model deployment and monitoring. Each stage has several steps that should be taken into action to detect cybercrimes in the Android systems. The design science approach is followed in this study to develop the detection model. This approach provides a systematic framework for designing and developing effective detection models [7, 14, 15].

II. RELATED WORKS

This section reviews the existing studies conducted on ML-based malware detection in Android systems over the past few years, which will help develop future models applicable to this domain. In [16], the malware in question was detected adopting the standard Artificial Neural Network (ANN) structure that was applied to the classification just before installing the application using the classification algorithm. Authors in [17] examined the data collected from several different systems and networks to identify potential security problems and vulnerabilities. Then thery trained an intrusion detection system by combining ML algorithms, known attack styles, and data from a server-based attack method.

Several efforts have been proposed to examine cybercrimes. These contain database schemes, mechanization, cyberbullying prevention, wireless webs, cloud protection, intelligent IoT drone area, mobile zone, and health area [18-36]. As a part of fraud detection, companies and organizations monitor users' patterns to identify, spot, and prevent fraudulent activity. Authors in [37] evaluated the CICID2017 dataset. Numerous files were merged to identify many types of dual classification incidents under the same name from a dataset that included numerous classifications. Authors in [38] developed a new Android malware dataset, called CICAndMal2017, utilizing genuine smartphones. They showed that 80 types of traffic could be detected, and malicious families could be classified employing traffic analysis. According to [39], the Long Short-Term Memory (LSTM) based on static and dynamic features can be implemented to detect Android malware. In [3], an algorithm was developed for detecting malware based on factorization. The data utilized in this study were derived from the DREBIN and AMD databases. The evaluations performed demonstrated that the proposed method achieved detection results with 100% precision. Authors in [40] constructed a permission-based model for security and privacy. The model was aimed at determining whether some mobile applications can use spare permissions for suspicious activities even when they do not need them. In [41], the architecture of inter-process communication was proposed as a feature to detect Android malware. The architecture of the system consists of three modules: Android application framework, detector, and utilizer interface. In [42], MobiDroid, which can be deployed to detect malware on Android devices, was developed. MobiDroid provides a deep learning-based, real-time, safe, and fast response environment that is established on

Deep Learning (DL). Authors in [43] introduced a framework that engages supervised ML models to distinct and classify Android ransomware apps from benign apps. Based on a static analysis of unknown ransomware apps, the proposed framework extracts novel features to detect them. To compare the computation time that ML models take to detect Android ransomware, the framework was implemented on GPUs and CPUs. An approach based on ML was presented in [44] to improve malware detection on Android platforms. In that study, datasets are pre-processed, features are engineered, model-building is completed, models are evaluated, and applications are performed. In [45], a novel DL approach was proposed concentrated on detecting first-time appearing malware efficiently by offering a higher level of performance than conventional methods. Authors in [46] proposed a novel approach to Android malware detection, which employs Sensitive Function Call Graphs (FCGs), called SeGDroid. Based on a multilevel architecture, in [47], a novel classifier fusion approach, called Droid Fusion, was introduced. This approach allows for an efficient combination of ML algorithms to increase accuracy. A base classifier is trained by Droid Fusion at a lower level, which then applies ranking-based algorithms to determine the predictive accuracy of the base classifiers at a higher level. This occurs for combination schemes to be generated and then be used to build a final classification model. According to [48], SVMs and Active Learning technologies can be adopted to detect Android malicious applications. The authors employed dynamic data extraction features and attached timestamps to some of them to enhance malware detection accuracy by following a novel time-dependent behavior-tracking method. Authors in [49] conducted a survey examining the use of ML methods for the malware analysis and claimed that ML is the most common technique used to analyze complex malware in literature. In [50], two approaches for analyzing Android malware statically based on ML were presented. An Android permission analysis approach was first deployed, while an Android source code analysis technique using a bag-of-words representation was also implemented. Authors in [51] combined heterogeneous classifiers putting into service static features extracted from 6,863 app samples to develop a composite classification model. In [52], a research on the use of ML algorithms to detect Android malware was conducted. To produce the proposed ML Detection Model, different static features of applications were compared and analyzed. In [53], the authors studied the detection of Android malware using both static and dynamic off-device analysis. Features were extracted from manifest files and the code was disassembled. Among these high-dimensional features are permissions, file operations, intents, API calls, components, network statistics, phone events, packages, and serial numbers.

III. METHODOLOGY

This section describes the methodology employed in this study. The design science method [54] was followed to design the detection model for cybercrime in Android systems utilizing ML technology. Design science approaches focus on creating innovative products and solutions that meet market demands by integrating scientific knowledge and practical application. Design science processes are iterative, which makes them unique among other approaches [55]. The process of developing, implementing, and evaluating the solution is a continuous one, allowing for refinement and improvement as each step is completed. An iterative approach can produce a user-focused and well-designed solution. During design science sessions, the emphasis is placed on rigor, repeatability, and evidence. To ensure repeatability and reliability, scientific principles and methods are applied to the design process. Consequently, this scientific rigor ensures that solutions can be applied across contexts and to similar problems that share similar characteristics [55]. Figure 3 displays the methodology used in this study, which consists of two main stages:

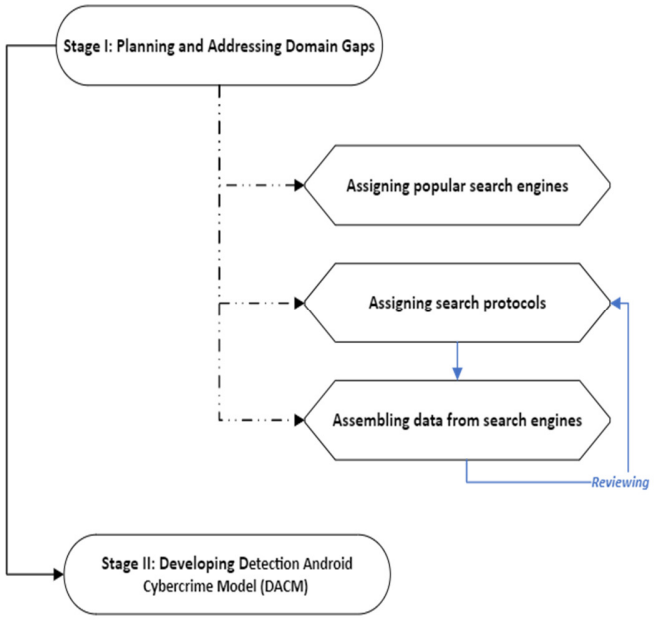

Fig. 3. The methodology used in this study.

1) Stage I: Planning and Addressing Domain Gaps

This stage provides a detailed description of the domain, involving the research directions, limitations, advantages, and hot topics associated with the latter. To achieve these points, several tasks are required at this stage of the process:

- Task1: Assigning popular search engines**:** The research databases assigned to this task are IEEE Xplore, Web of Science, Springer Link, ACM, Scopus, ScienceDirect, and Google Scholar. These databases are commonly utilized and known for their extensive collections of academic resources. They provide access to a wide range of academic resources, facilitating thorough research to be conducted and relevant and reliable information to be gathered.
- Task2: Assigning search protocols: The author is responsible for assigning the search protocols to be utilized in research databases so that search behaviors can be governed by those protocols. A search protocol identifies the keywords, the language, and the date of publication. A few of the keywords employed in this study were "Malware Detection", "Android", and "Machine Learning". The

language of the study was set to English, and the publication dates to 2015–2024.

- Task3: Assembling data from search engines: Based on the search protocols assigned in the previous task, the data were collected from the selected databases (see Table I).

TABLE I. DATA COLLECTED FROM EXISTING ML-BASED MALWARE DETECTION MODELS ON ANDROID.

Year	Ref	Description
2014	[47]	Proposed a novel classifier fusion approach called Droid Fusion.
2015	[53]	Studied the detection of Android malware using both static and dynamic off-device analysis.
2016	[53]	Conducted research on the use of ML algorithms to detect Android malware.
2017	[41]	Proposed the architecture of inter-process communication as a feature to detect Android malware.
2017	[50]	Proposed two approaches for analyzing Android malware statically based on ML.
2018	[38]	Developed CICAndMal2017, which was the first Android malware dataset, using genuine smartphones and called it the CICAndMal dataset.
2018	[39]	Android malware detection employing LSTM by using static and dynamic features. Also, static analysis of Android malware was conducted
2018	[51]	Combined heterogeneous classifiers using static features extracted from 6,863 app samples to develop a composite classification model.
2019	[16]	Used a classification algorithm to detect malware deploying the standard ANN structure.
2019	[17]	Investigated data from multiple systems and networks to identify security vulnerabilities. They also combined ML algorithms with known attack techniques and data from server-based attacks to train the intrusion detection system.
2019	[37]	Merged several files to identify multiple types of incidents with two classifications under the same name in a dataset containing many classifications.
2019	[3]	Developed an algorithm for detecting malware based on factorization.
2019	[40]	Constructed a permission-based model for security and privacy.
2019	[42]	Developed MobiDroid that can be used to detect malware on Android devices.
2019	[48]	Applied SVMs and Active Learning technologies to detect Android malicious applications.
2019	[49]	Survey examining the use of ML methods for malware analysis.
2020	[43]	Developed a framework that uses supervised ML models to distinguish Android ransomware apps from benign apps and classify them.
2021	[56]	Described how ML models have been used to create intelligent solutions to curb cybercrime threats in the past decade. Using published materials from notable databases, an exploratory approach is adopted.
2021	[57]	Used forensic analysis and reverse engineering of Android ransomware to extract static features. Developed the novel Ransom Droid framework to address problems related to mislabeling historical targets and detecting unforeseen Android ransomware using clustering-based unsupervised ML techniques.
2022	[58]	Developed an algorithm for malware detection using ML and DL for Android combining static and dynamic analysis.
2024	[44]	Suggested an ML approach to improving malware detection on Android.
2024	[45]	Proposed a novel deep learning approach, with a focus on detecting first-time appearing malware effectively and efficiently by offering a higher level of performance than conventional methods for detecting malware.
2024	[46]	Presented SeGDroid which uses sensitive FCGs.

As the literature review clarifies, one of the major limitations of the current Android cybercrime detection models is the lack of a comprehensive model that encompasses all the phases of the detection process. The current models often focus on only a small amount of the detection process, such as identifying malware samples or detecting abnormal activities, but they fail to capture the entire detection process. As a consequence, the system is unable to perform as effectively as it could. This weakness has encouraged the present study to develop a comprehensive detection model for Android cybercrimes.

2) Stage II: Developing the Detection Android Cybercrime Model (DACM)

At this stage, DACM was developed using the design science research method. The developed model consists of five stages as shown in Figure 4.

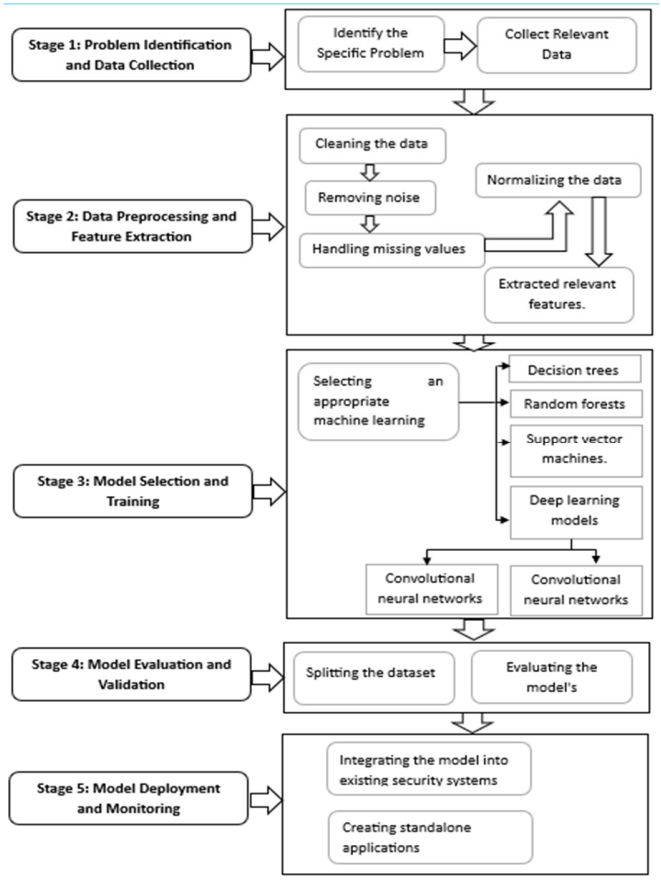

Fig. 4. The detection model for cybercrimes in Android using ML.

- Stage 1: Problem Identification and Data Collection: This stage aims to identify the problems associated with cybercrime in Android systems and the solutions to those problems. This helps recognize the types of cybercrime that may have been perpetrated on these systems, including malware, phishing attacks, and unauthorized access. Once a problem is identified, the next step is to collect data that can be used to develop training and evaluation plans for the

problem. Besides collecting data from these malicious apps, the data could be also collected from the network traffic logs, the system logs, and other datasets that may be of assistance in the investigation process.

- Stage 2: Data Preprocessing and Feature Extraction: To ensure the quality of the collected data and their compatibility with the ML algorithms, these data are pre-processed. The type of preprocessing technique applied depends on what type of data are going to be used, e.g. data cleaning, noise removal, filling missing values, or normalization. Then, the relevant features in the data are extracted from the pre-processed data. This extraction can be performed by employing a variety of techniques, depending on the nature of the data, such as statistical analysis, text mining, or even image processing, based on the nature of the data.
- Stage 3: Model Selection and Training: The next step is to pick an ML model suitable for the detection task. Data characteristics and the problem at hand must be considered when choosing a model. Several ML algorithms are utilized to detect cybercrimes, including decision trees, random forests, support vector machines, convolutional or recurrent neural networks, and DL. The selected model is then trained on the pre-processed data implementing a suitable training algorithm.
- Stage 4: Model Evaluation and Validation: During this stage, the dataset is split into two sets, the training, and the testing datasets. The performance of a model can be assessed through numerous metrics, including accuracy, precision, recall, and the F1 score. When a model has performed well on a test set and on real data, it may be possible to consider deploying it. As a result, adjustments might be needed, such as fine-tuning the model or exploring different algorithms.
- Stage 5: Model Deployment and Monitoring: In real-life scenarios, once a detection model is found effective and reliable, it can be deployed to detect and prevent cybercrime in Android systems in real-life situations that can be observed regularly. Models are either deployed as partσ of existing security systems or created from scratch to cover a particular set of requirements. To keep up with cyber threats, model performance must be continuously monitored and updated.

IV. FINDINGS AND DISCUSSION

Table III and Figure 4 compare the developed model with the existing malware and cybercrime detection models designed for Android. The comparison of the developed DACM with the existing cybercrime detection models on Android revealed that while the developed model has five stages, the existing models have a maximum of three stages. The problem identification and data collection stage has been covered in 7 studies [17, 37, 40, 42, 56, 57, 58], whereas the data preprocessing and feature extraction in six [3, 42, 48, 56, 59]. The model selection and training and model evaluation and validation processes have been covered by eight and nine authors, respectively, as exhibited in Table II. Finally, the model deployment and monitoring has been covered by 12 studies. The results disclose that the developed DACM is comprehensive and covers all of the existing detection models.

V. CONCLUSION

To detect different Android cybercrimes, this paper developed the Detection Android Cybercrime Model (DACM) using the design science approach. The developed DACM consists of five stages: problem identification and data collection, data preprocessing and feature extraction, model selection and training, model evaluation and validation, and model deployment and monitoring. In future research, experiments could be carried out to verify the effectiveness of the developed DACM.

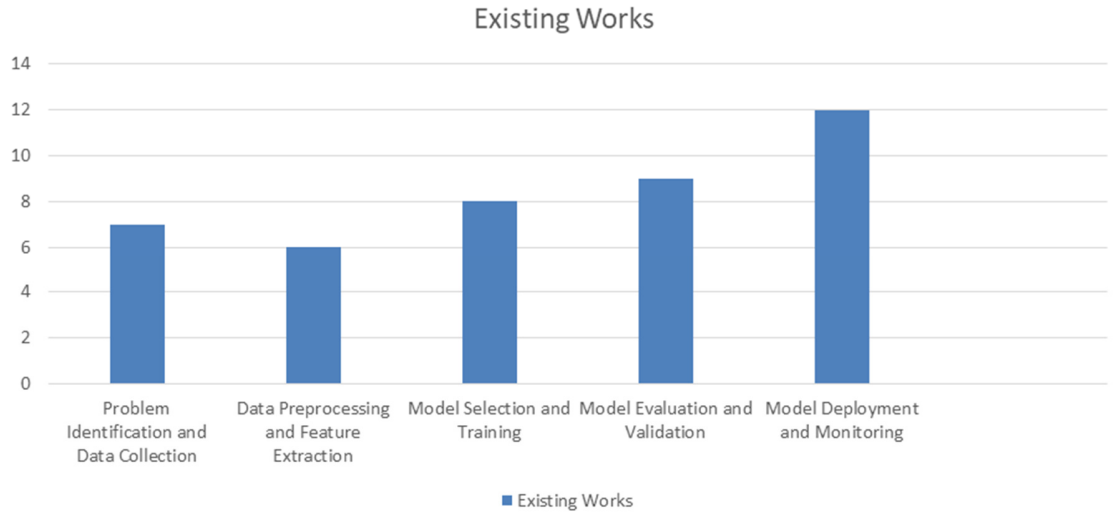

Fig. 5. Comparison between the developed DACM and the existing cybercrimes detection models on Android.

TABLE II. COMPARISON BETWEEN THE DEVELOPED DACM AND EXISTING CYBERCRIME DETECTION MODELS FOR ANDROID

DACM (proposed)	Existing Cybercrime Detection Models on Android																
	[16]	[17]	[37]	[38]	[59]	[39]	[3]	[13]	[41]	[40]	[42]	[48]	[49]	[43]	[56]	[57]	[58]
Problem Identification and Data Collection	✘	✓	✓	✘	✘	✘	✘	✘	✘	✘	✓	✘	✘	✘	✓	✓	✓
Data Preprocessing and Feature Extraction	✘	✘	✘	✘	✓	✘	✓	✘	✘	✘	✘	✓	✓	✘	✓	✘	✘
Model Selection and Training	✓	✘	✘	✘	✓	✘	✘	✘	✓	✓	✘	✘	✓	✘	✓	✘	✓
Model Evaluation and Validation	✓	✓	✘	✘	✓	✓	✘	✓	✘	✓	✓	✘	✘	✓	✘	✓	✘
Model Deployment and Monitoring	✓	✓	✓	✓	✘	✓	✓	✓	✓	✓	✓	✘	✓	✘	✘	✓	✘

REFERENCES

[1] F. Alotaibi, A. Al-Dhaqm, and Y. D. Al-Otaibi, "A Conceptual Digital Forensic Investigation Model Applicable to the Drone Forensics Field," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11608–11615, Oct. 2023, https://doi.org/10.48084/etasr.6195.

[2] P. Weichbroth and Ł. Łysik, "Mobile Security: Threats and Best Practices," *Mobile Information Systems*, vol. 2020, Dec. 2020, Art. no. e8828078, https://doi.org/10.1155/2020/8828078.

[3] C. Li, K. Mills, D. Niu, R. Zhu, H. Zhang, and H. Kinawi, "Android Malware Detection Based on Factorization Machine," *IEEE Access*, vol. 7, pp. 184008–184019, 2019, https://doi.org/10.1109/ACCESS. 2019.2958927.

[4] E. C. Bayazit, O. Koray Sahingoz, and B. Dogan, "Malware Detection in Android Systems with Traditional Machine Learning Models: A Survey," in *International Congress on Human-Computer Interaction, Optimization and Robotic Applications*, Ankara, Turkey, Jun. 2020, pp. 1–8, https://doi.org/10.1109/HORA49412.2020.9152840.

[5] N. I. Che Mat, N. Jamil, Y. Yusoff, and M. L. Mat Kiah, "A systematic literature review on advanced persistent threat behaviors and its detection strategy," *Journal of Cybersecurity*, vol. 10, no. 1, Jan. 2024, Art. no. tyad023, https://doi.org/10.1093/cybsec/tyad023.

[6] I. H. Sarker, "Machine Learning for Intelligent Data Analysis and Automation in Cybersecurity: Current and Future Prospects," *Annals of Data Science*, vol. 10, no. 6, pp. 1473–1498, Dec. 2023, https://doi.org/10.1007/s40745-022-00444-2.

[7] A. S. Alraddadi, "A Survey and a Credit Card Fraud Detection and Prevention Model using the Decision Tree Algorithm," *Engineering, Technology & Applied Science Research*, vol. 13, no. 4, pp. 11505–11510, Aug. 2023, https://doi.org/10.48084/etasr.6128.

[8] A. Alshammari, "A Novel Security Framework to Mitigate and Avoid Unexpected Security Threats in Saudi Arabia," *Engineering, Technology & Applied Science Research*, vol. 13, no. 4, pp. 11445–11450, Aug. 2023, https://doi.org/10.48084/etasr.6091.

[9] S. Y. Yerima, S. Sezer, and I. Muttik, "High accuracy android malware detection using ensemble learning," *IET Information Security*, vol. 9, no. 6, pp. 313–320, 2015, https://doi.org/10.1049/iet-ifs.2014.0099.

[10] K. Wagstaff, "Machine Learning that Matters." arXiv, Jun. 18, 2012, https://doi.org/10.48550/arXiv.1206.4656.

[11] O. V. Lee *et al.*, "A malicious URLs detection system using optimization and machine learning classifiers," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 3, pp. 1210–1214, Mar. 2020, https://doi.org/10.11591/ijeecs.v17.i3.pp1210-1214.

[12] N. S. Zaini *et al.*, "Phishing detection system using machine learning classifiers," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 3, pp. 1165–1171, 2019.

[13] D. Abel, A. Barreto, B. Van Roy, D. Precup, H. P. van Hasselt, and S. Singh, "A Definition of Continual Reinforcement Learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 50377–50407, Dec. 2023.

[14] A. Al-Dhaqm, S. A. Razak, S. H. Othman, A. Nagdi, and A. Ali, "A generic database forensic investigation process model," *Jurnal Teknologi*, vol. 78, no. 6–11, pp. 45–57, Jun. 2016, https://doi.org/ 10.11113/jt.v78.9190.

[15] A. A. Alhussan, A. Al-Dhaqm, W. M. S. Yafooz, A.-H. M. Emara, S. Bin Abd Razak, and D. S. Khafaga, "A Unified Forensic Model Applicable to the Database Forensics Field," *Electronics*, vol. 11, no. 9, Jan. 2022, Art. no. 1347, https://doi.org/10.3390/electronics11091347.

[16] H. R. Sandeep, "Static Analysis of Android Malware Detection using Deep Learning," in *International Conference on Intelligent Computing and Control Systems*, Madurai, India, Dec. 2019, pp. 841–845, https://doi.org/10.1109/ICCS45141.2019.9065765.

[17] M. Takaoglu and C. Ozer, "Saldiri Tespit Sistemlerine Makine Ogrenme Etkisi," *Uluslararası Yonetim Bilisim Sistemleri ve Bilgisayar Bilimleri Dergisi*, vol. 3, no. 1, pp. 11–22, Jun. 2019, https://doi.org/10.33461/ uybisbbd.558192.

[18] A. Al-Dhaqm, W. M. S. Yafooz, S. H. Othman, and A. Ali, "Database Forensics Field and Children Crimes," in *Kids Cybersecurity Using Computational Intelligence Techniques*, W. M. S. Yafooz, H. Al-Aqrabi, A. Al-Dhaqm, and A. Emara, Eds. New York, NY, USA: Springer, 2023, pp. 81–92.

[19] M. Q. Mohammed *et al.*, "Deep Reinforcement Learning-Based Robotic Grasping in Clutter and Occlusion," *Sustainability*, vol. 13, no. 24, Jan. 2021, Art. no. 13686, https://doi.org/10.3390/su132413686.

[20] W. M. S. Yafooz, A. Al-Dhaqm, and A. Alsaeedi, "Detecting Kids Cyberbullying Using Transfer Learning Approach: Transformer Fine-Tuning Models," in *Kids Cybersecurity Using Computational Intelligence Techniques*, W. M. S. Yafooz, H. Al-Aqrabi, A. Al-Dhaqm, and A. Emara, Eds. New York, NY, USA: Springer, 2023, pp. 255–267.

[21] I. U. Onwuegbuzie, S. A. Razak, I. F. Isnin, A. Al-dhaqm, and N. B. Anuar, "Prioritized Shortest Path Computation Mechanism (PSPCM) for wireless sensor networks," *PLOS ONE*, vol. 17, no. 3, Mar. 2022, Art. no. e0264683, https://doi.org/10.1371/journal.pone.0264683.

[22] A. Al-dhaqm, M. Bakhtiari, E. Alobaidi, and A. Saleh, "Studding and Analyzing Wireless Networks Access points," *International Journal of Scientific & Engineering Research*, vol. 4, no. 1, pp. 1–8, 2013.

[23] R. Al-Mugerrn, A. Al-Dhaqm, and S. H. Othman, "A Metamodeling Approach for Structuring and Organizing Cloud Forensics Domain," in *International Conference on Smart Computing and Application*, Hail, Saudi Arabia, Feb. 2023, pp. 1–5, https://doi.org/10.1109/ ICSCA57840.2023.10087425.

[24] A. A. Zubair *et al.*, "A Cloud Computing-Based Modified Symbiotic Organisms Search Algorithm (AI) for Optimal Task Scheduling," *Sensors*, vol. 22, no. 4, Jan. 2022, Art. no. 1674, https://doi.org/10.3390/s22041674.

[25] B. E. Sabir, M. Youssfi, O. Bouattane, and H. Allali, "Towards a New Model to Secure IoT-based Smart Home Mobile Agents using Blockchain Technology," *Engineering, Technology & Applied Science Research*, vol. 10, no. 2, pp. 5441–5447, Apr. 2020, https://doi.org/ 10.48084/etasr.3394.

[26] M. Saleh *et al.*, "A Metamodeling Approach for IoT Forensic Investigation," *Electronics*, vol. 12, no. 3, Jan. 2023, Art. no. 524, https://doi.org/10.3390/electronics12030524.

[27] A. E. Yahya, A. Gharbi, W. M. S. Yafooz, and A. Al-Dhaqm, "A Novel Hybrid Deep Learning Model for Detecting and Classifying Non-Functional Requirements of Mobile Apps Issues," *Electronics*, vol. 12, no. 5, Jan. 2023, Art. no. 1258, https://doi.org/10.3390/ electronics12051258.

[28] K. N. Qureshi *et al.*, "A Blockchain-Based Efficient, Secure and Anonymous Conditional Privacy-Preserving and Authentication Scheme for the Internet of Vehicles," *Applied Sciences*, vol. 12, no. 1, Jan. 2022, Art. no. 476, https://doi.org/10.3390/app12010476.

[29] A. M. R. Al-dhaqm and Md. A. Nagdi, "Detection and Prevention of Malicious Activities on RDBMS Relational Database Management Systems," *International Journal of Scientific & Engineering Research*, vol. 3, no. 9, Sep 12, [Online]. Available: https://www.ijser.org/paper/

Detection-and-Prevention-of-Malicious-Activities-on-RDBMS-Relational-Database-Management-Systems.html.

[30] I. U. Onwuegbuzie, S. A. Razak, I. F. Isnin, T. S. J. Darwish, and A. Al-dhaqm, "Optimized backoff scheme for prioritized data in wireless sensor networks: A class of service approach," *PLOS ONE*, vol. 15, no. 8, Jul. 2020, Art. no. e0237154, https://doi.org/10.1371/journal.pone.0237154.

[31] S. Abd Razak, N. H. Mohd Nazari, and A. Al-Dhaqm, "Data Anonymization Using Pseudonym System to Preserve Data Privacy," *IEEE Access*, vol. 8, pp. 43256–43264, 2020, https://doi.org/10.1109/ACCESS.2020.2977117.

[32] W. A. H. Altowayti *et al.*, "The Role of Conventional Methods and Artificial Intelligence in the Wastewater Treatment: A Comprehensive Review," *Processes*, vol. 10, no. 9, Sep. 2022, Art. no. 1832, https://doi.org/10.3390/pr10091832.

[33] M. Rasool, N. A. Ismail, A. Al-Dhaqm, W. M. S. Yafooz, and A. Alsaeedi, "A Novel Approach for Classifying Brain Tumours Combining a SqueezeNet Model with SVM and Fine-Tuning," *Electronics*, vol. 12, no. 1, Jan. 2023, Art. no. 149, https://doi.org/10.3390/electronics12010149.

[34] M. Q. Mohammed *et al.*, "Review of Learning-Based Robotic Manipulation in Cluttered Environments," *Sensors*, vol. 22, no. 20, Jan. 2022, Art. no. 7938, https://doi.org/10.3390/s22207938.

[35] I. U. Onwuegbuzie, S. A. Razak, and A. Al-Dhaqm, "Multi-Sink Load-Balancing Mechanism for Wireless Sensor Networks," in *IEEE International Conference on Computing*, Kuala Lumpur, Malaysia, Nov. 2021, pp. 140–145, https://doi.org/10.1109/ICOCO53166.2021.9673578.

[36] D. M. Bakhtiari and A. M. R. Al-dhaqm, "Mechanisms to Prevent lose Data," *International Journal of Scientific & Engineering Research*, vol. 3, no. 12, pp. 1–4, 2012.

[37] H. Ahmetoglu and R. Das, "Derin Ogrenme ile Buyuk Veri Kumelerinden Saldiri Turlerinin Siniflandirilmasi," in *International Artificial Intelligence and Data Processing Symposium*, Malatya, Turkey, Sep. 2019, pp. 1–9, https://doi.org/10.1109/IDAP.2019.8875872.

[38] A. H. Lashkari, A. F. A. Kadir, L. Taheri, and A. A. Ghorbani, "Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification," in *International Carnahan Conference on Security Technology*, Montreal, QC, Canada, Oct. 2018, pp. 1–7, https://doi.org/10.1109/CCST.2018.8585560.

[39] R. Vinayakumar, K. P. Soman, P. Poornachandran, and S. Sachin Kumar, "Detecting Android malware using Long Short-term Memory (LSTM)," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1277–1288, Jan. 2018, https://doi.org/10.3233/JIFS-169424.

[40] R. S. Arslan, I. A. Dogru, and N. Barisci, "Permission-Based Malware Detection System for Android Using Machine Learning Techniques," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 1, pp. 43–61, Jan. 2019, https://doi.org/10.1142/S0218194019500037.

[41] A. Feizollah, N. B. Anuar, R. Salleh, G. Suarez-Tangil, and S. Furnell, "AndroDialysis: Analysis of Android Intent Effectiveness in Malware Detection," *Computers & Security*, vol. 65, pp. 121–134, Mar. 2017, https://doi.org/10.1016/j.cose.2016.11.007.

[42] R. Feng *et al.*, "MobiDroid: A Performance-Sensitive Malware Detection System on Mobile Platform," in *24th International Conference on Engineering of Complex Computer Systems*, Guangzhou, China, Nov. 2019, pp. 61–70, https://doi.org/10.1109/ICECCS.2019.00014.

[43] S. Sharma, C. R. Krishna, and R. Kumar, "Android Ransomware Detection using Machine Learning Techniques: A Comparative Analysis on GPU and CPU," in *21st International Arab Conference on Information Technology*, Giza, Egypt, Nov. 2020, pp. 1–6, https://doi.org/10.1109/ACIT50332.2020.9300108.

[44] H. A. Al-Ofeishat, "Enhancing Android Security: Network-Driven Machine Learning Approach For Malware Detection," *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 2, pp. 737–750, 2024.

[45] K. Shaukat, S. Luo, and V. Varadharajan, "A novel machine learning approach for detecting first-time-appeared malware," *Engineering Applications of Artificial Intelligence*, vol. 131, May 2024, Art. no. 107801, https://doi.org/10.1016/j.engappai.2023.107801.

[46] Z. Liu, R. Wang, N. Japkowicz, H. M. Gomes, B. Peng, and W. Zhang, "SeGDroid: An Android malware detection method based on sensitive function call graph learning," *Expert Systems with Applications*, vol. 235, Jan. 2024, Art. no. 121125, https://doi.org/10.1016/j.eswa.2023.121125.

[47] S. Y. Yerima, S. Sezer, and I. Muttik, "Android Malware Detection Using Parallel Machine Learning Classifiers," in *Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, Oxford, UK, Sep. 2014, pp. 37–42, https://doi.org/10.1109/NGMAST.2014.23.

[48] B. Rashidi, C. Fung, and E. Bertino, "Android malicious application detection using support vector machine and active learning," in *13th International Conference on Network and Service Management*, Tokyo, Japan, Nov. 2017, pp. 1–9, https://doi.org/10.23919/CNSM.2017.8256035.

[49] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security*, vol. 81, pp. 123–147, Mar. 2019, https://doi.org/10.1016/j.cose.2018.11.001.

[50] N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, "Machine learning aided Android malware classification," *Computers & Electrical Engineering*, vol. 61, pp. 266–274, Jul. 2017, https://doi.org/10.1016/j.compeleceng.2017.02.013.

[51] S. Y. Yerima and S. Sezer, "DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 453–466, Oct. 2019, https://doi.org/10.1109/TCYB.2017.2777960.

[52] S. Hahn, M. Protsenko, and T. Müller, "Comparative evaluation of machine learning-based malware detection on android.," in *Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit*, 2016, pp. 79–88, [Online]. Available: https://dl.gi.de/items/c8d84289-435d-413a-affc-abc26ff184eb.

[53] M. Lindorfer, M. Neugschwandtner, and C. Platzer, "MARVIN: Efficient and Comprehensive Mobile App Classification through Static and Dynamic Analysis," in *39th Annual Computer Software and Applications Conference*, Taichung, Taiwan, Jul. 2015, vol. 2, pp. 422–433, https://doi.org/10.1109/COMPSAC.2015.103.

[54] F. M. Alotaibi, A. Al-Dhaqm, W. M. S. Yafooz, and Y. D. Al-Otaibi, "A Novel Administration Model for Managing and Organising the Heterogeneous Information Security Policy Field," *Applied Sciences*, vol. 13, no. 17, Jan. 2023, Art. no. 9703, https://doi.org/10.3390/app13179703.

[55] A. Al-Dhaqm *et al.*, "Categorization and Organization of Database Forensic Investigation Processes," *IEEE Access*, vol. 8, pp. 112846–112858, 2020, https://doi.org/10.1109/ACCESS.2020.3000747.

[56] P. U. Chinedu, W. Nwankwo, F. U. Masajuwa, and S. Imoisi, "Cybercrime Detection and Prevention Efforts in the Last Decade: An Overview of the Possibilities of Machine Learning Models," *Rigeo*, vol. 11, no. 7, pp. 956–974, Aug. 2021, https://doi.org/10.48047/rigeo.11.07.92.

[57] S. Sharma, C. R. Krishna, and R. Kumar, "RansomDroid: Forensic analysis and detection of Android Ransomware using unsupervised machine learning technique," *Forensic Science International: Digital Investigation*, vol. 37, Jun. 2021, Art. no. 301168, https://doi.org/10.1016/j.fsidi.2021.301168.

[58] M. S. Hossain and M. H. Riaz, "Android Malware Detection System: A Machine Learning and Deep Learning Based Multilayered Approach," in *International Conference on Intelligent Computing & Optimization*, Hua Hin, Thailand, Oct. 2022, pp. 277–287, https://doi.org/10.1007/978-3-030-93247-3_28.

[59] L. Taheri, A. F. A. Kadir, and A. H. Lashkari, "Extensible Android Malware Detection and Family Classification Using Network-Flows and API-Calls," in *International Carnahan Conference on Security Technology*, Chennai, India, Oct. 2019, pp. 1–8, https://doi.org/10.1109/CCST.2019.8888430.