Enhancing Android Malware Detection in Internet of Vehicles using Self-Attention Transformer Model

Hassan A. Alterazi[1,*]

[1] *Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; haalterazi@kau.edu.sa*

Abstract

The current trend for vehicles to be significantly correlated with vehicles, unspecified devices, and organization upsurges the latent for exterior attacks on vehicle's cyber-security. The main network security function is intrusion detection with open connectivity, like connected cars and self-driving. Particularly, when a vehicle is associated with an exterior device over a device in the vehicle or when it connects with an exterior structure, cybersecurity is mandatory to defend the network of software inside the vehicle. Present technique with this concern comprises intrusion detection and a vehicle gateway system. Conversely, it is challenging to block mischievous code based on behaviors of application. This study presents an Enhancing Android Malware Detection using Self-Attention Transformer Model (EAMD-SATM) model in Internet of Vehicles. The projected EAMD-SATM model categorizes and recognizes the Android malware efficiently and accurately. To attain this, the EAMD-SATM approach endures a min-max approach utilizing data pre-processing at the initial stage. Furthermore, the EAMD-SATM method employs self-attention-based transformer (SA-T) technique for the detection of Android malware. To improve the SA-T technique solution, the EAMD-SATM technique applies the improved mother optimization (IMO) technique for the parameter tuning process. The simulation validation of the EAMD-SATM algorithm can be established on a benchmark Android malware dataset. The experimental outcomes highlighted the important performance of the EAMD-SATM approach in the Android malware recognition method.

Keywords

Improved Mother Optimization, Min-Max; Android Malware, Self-Attention Transformer, Internet of Vehicles

1. Introduction

In recent times Autonomous Vehicular System (AVS) should have spotted a huge development in a varied range of characteristics through the improvement of smart cities to construct Intelligent Transport System (ITS) [1]. Including, the vivid usage of embedded schemes and wireless communication viz., 5 G and 4 G LTE in recent vehicle internet that finally increases users' well-being and security [2]. Still, developing curiosity during the expansion of Connected Autonomous Vehicle (CAV and ITS has presented unique security

Proceedings of SNSFAIT 2024: International Symposium on Securing Next-Generation Systems using Future Artificial Intelligence Technologies, Delhi, India, August 08-09th, 2024

[1*] Corresponding author: Hassan A. Alterazi

haalterazi@kau.edu.sa

https://orcid.org/0000-0003-0137-9422

tasks and susceptibilities in AVSs, which had a major influence on the smart surroundings for smart cities [3]. On the other hand, traditional computer security results aren't valid in automated industrialized criteria for vehicle-to-vehicle (V2V) communication, vehicle-to-everything (V2X) communication, and in-vehicle communications mostly due to the real-time presentation requests, controlled computing resources, and dissimilarities between heterogeneous networks and their installations [4]. Malware detection is the main task in ITSs for several different applications and IoT devices are applied. Such as, self-driving vehicles are more susceptible to hacking these are linked to the Internet and may obtain diverse commands from mobile applications. Nevertheless, ancient cars don't have this innovative feature [5].

Those hacks are life-threatening for travelers in the vehicle, some other persons in another vehicle, and also, pedestrians. In real-time it is a tedious task to find out illegal activity [6]. Though, several machine learning (ML) and deep learning (DL) methods have been applied to detect this behavior. In addition, it presently provides "full self-driving" to proprietors of personal vehicles and offers "self-driving mode" in its vehicles [7]. Therefore, this incident is a notable landmark in AV improvement. Furthermore, as there a plentiful high-quality data sets presented and there a similar severe performance necessities, the academic community helps DL methods for ML-related responsibilities in Avs [8]. Meanwhile, Hinton released a unique deep-structured learning architecture, a deep belief network (DBN), and important developments were completed in DL. Present AVs trust intensely DL techniques for example image classification (IC), semantic segmentation (SS), and object detection (OD) for its execution [9]. Traffic sign recognition (TSR) is a vital DL application in AVs. It utilizes the DL model to classify the traffic sign image that is attained by the sensor camera after that employs the intelligent control system for controlling the car under the classification outcomes [10].

This study designs an Enhancing Android Malware Detection utilizing the Self-Attention Transformer Model (EAMD-SATM) model in Internet of Vehicles. The projected EAMD-SATM model categorizes and recognizes the Android malware efficiently and accurately. To attain this, the EAMD-SATM approach endures a min-max approach utilizing data pre-processing at the initial stage. Furthermore, the EAMD-SATM method employs self-attention-based transformer (SA-T) technique for the detection of Android malware. To improve the solution of the SA-T technique, the EAMD-SATM method applies the improved mother optimization (IMO) technique for the hyperparameter tuning method. The simulation validation of the EAMD-SATM approach can be established on a benchmark Android malware dataset.

2. The Literature Review

Ferrag *et al.* [11] propose SecurityBERT, a new structure, which leverages the Bi-directional Encoder Representation from Transformers (BERT) method. This method integrated a new privacy-preserving encoding method named Privacy Preserving Fixed Length Encoding (PPFLE). The technique effectually represents the network traffic data in a structural format by uniting PPFLE with the Byte level Byte Pair Encoder (BBPE) Tokenizers. In [12], the author examines the innovative ML method applications, particularly in Bi-directional LSTM (BiLSTM) and LSTM structures, enhanced by the word embedding methods. The study begins with a systematic study of stringent data processing methods and basic ML principles, creating a robust basis for sequential stages. The research initiates the refinement and formation of a specific DL method that is elaborately intended for the precise recognition of hidden malware in execution files. Islam *et al.* [13] present accurate, practical, and robust systems to recognize medical plants from smartphone seized plant imageries in the plant sites. The presented method used a cascade structure to mine the features by utilizing a pre-trained ResNet50 method that is enhanced by utilizing a Particle Swarm Optimizer (PSO) to identify the plants.

Ullah *et al.* [14] introduce a new network IDS for VANET that levers Spark-based big data optimizer and transfer learning (NIDS-VSB). At initial, a packet parser is utilized to crawl the filter required flow event and network traffic. Then, a Spark-based optimizer technique is executed to process the huge quantities of data effectively. Additionally, a transfer learning method is created to study extensive feature representation by utilizing their semantic anchor. Then, a stack generality ensemble method utilizes deep feature to identify many assaults. Liu *et al.* [15] propose MalIRL to design a model-free inverse reinforcement learning (IRL) method. Especially, MalIRL examines 6 representative group activities of malware and uses sliding windows to essentially separate the large malware implementation event streams into many attacks' phases, attaining a lower state and action spaces. To perfect dynamic malicious atmospheres, MalIRL presents a prompt dynamic heterogeneous graph represented by learning methods.

3. The Proposed Model

This study proposes an EAMD-SATM model. The presented EAMD-SATM model categorizes and recognizes the Android malware efficiently and accurately. To attain this, the EAMD-SATM approach comprises min-max-based data preprocessing, SA-T-based Android malware detection, and IMO-based hyperparameter tuning processes. Fig. 1 illustrates the workflow of the EAMD-SATM model.

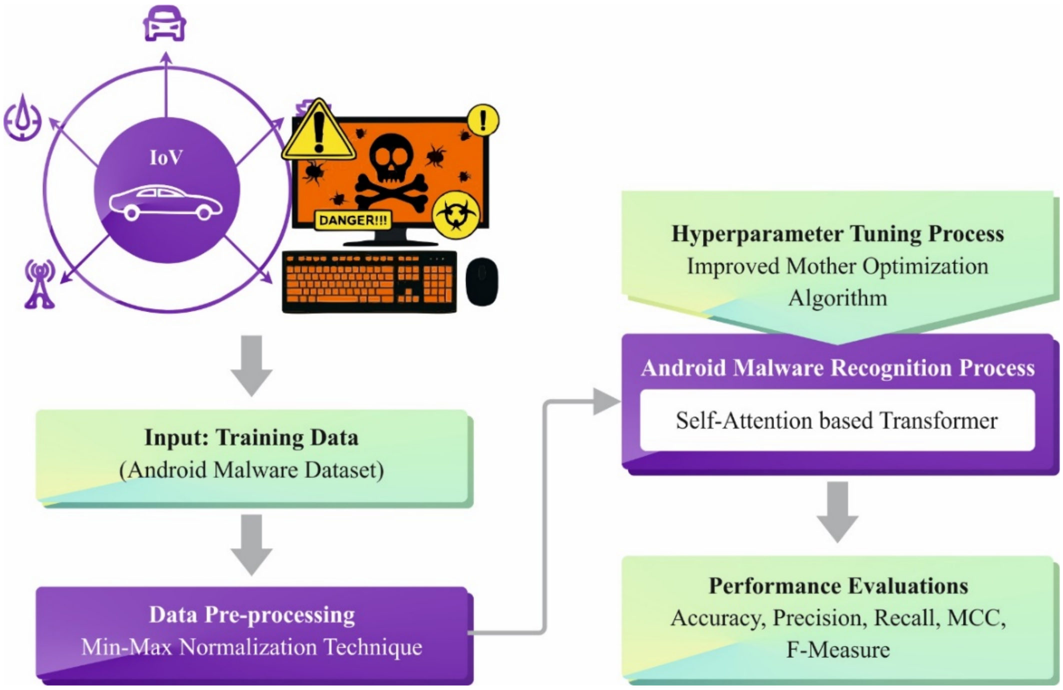

IoV
DANGER!!!
Hyperparameter Tuning Process
Improved Mother Optimization Algorithm
Android Malware Recognition Process
Self-Attention based Transformer
Input: Training Data
(Android Malware Dataset)
Data Pre-processing
Min-Max Normalization Technique
Performance Evaluations
Accuracy, Precision, Recall, MCC, F-Measure

Figure 1: Workflow of EAMD-SATM model

3.1. Data Preprocessing

The presented EAMD-SATM method utilizes the min-max approach for the data pre-processing process [16]. The min-max normalization is influential in data pre-processing for Android malware recognition, certifying that values of a feature are adapted to a constant range among 0 and 1. By converting data in this method, min-max normalization allows for impartial comparison and effectual training of machine learning (ML) methods, allowing precise classification of malicious behaviors and patterns within Android applications. This standardized method improves the abilities of detection, making the network more robust against developing malware attacks in the IoV context.

3.2. Classification Process

The SA-T technique is employed for the classification process of the proposed model [17]. These methods are signified by an input feature sequence that is employed to encrypt every case in the database. In this research paper, we contain $X = [x_1, x_2, \dots, x_n]$, signify the sequence of input features, whereas n represents the length of sequence. The self-attention method identifies the relationship among numerous features in the series and provides a weight depending on how significant it is for other attributes. To take numerous kinds of relations and enhance the efficiency of the model, numerous equivalent layers of self-attention are employed. The outputs are served into feed-forward neural networks for recognizing the non-linear relations and deliver last forecasts, This attention-based method structure with multi-head attention and self-attention methods effectively attains dependencies and connections within the series of inputs. Fig. 2 represents the structure of the transformer method.

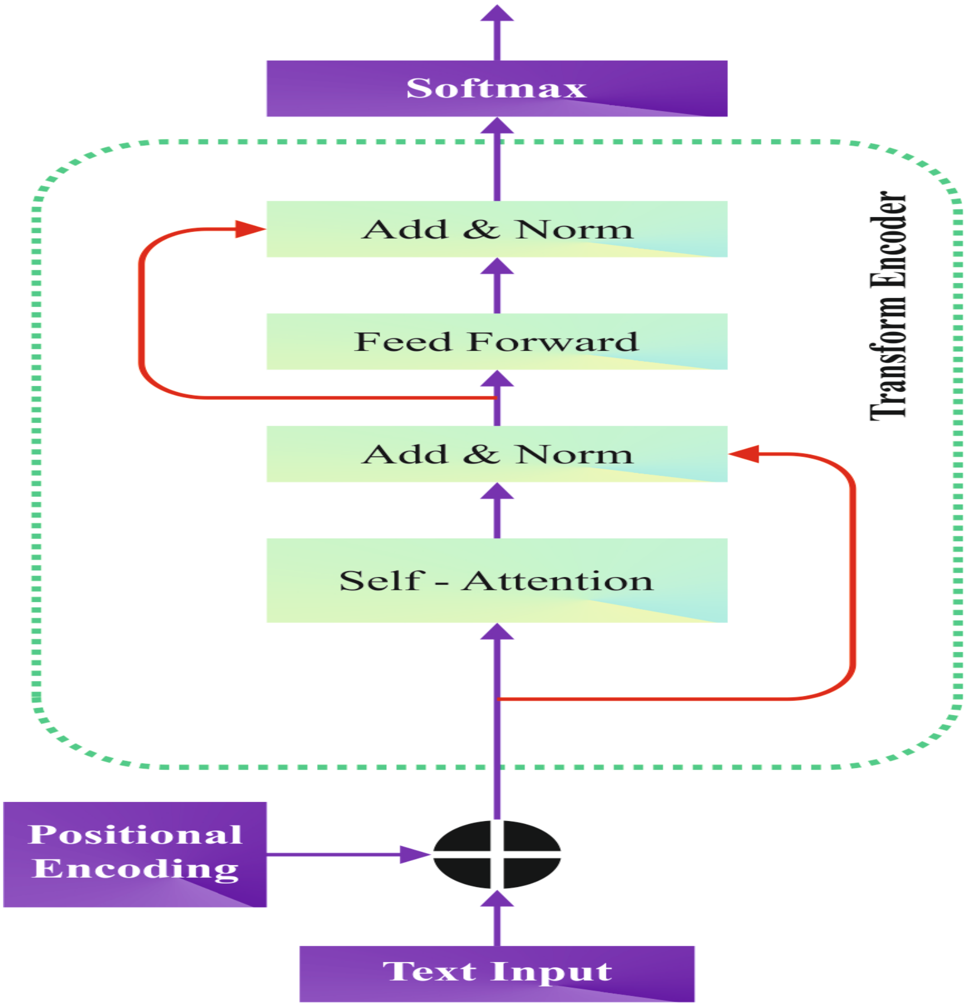


Figure 2: Structure of Transformer model

In the self-attention-based transformer method, location encoding and input embedding are dual vital methods. The series of inputs is signified by utilizing these phases, which is suitable for the following self-attention layer. The numerical features and definite variables of every case are decoded to constant vector representation over input embeddings. While, $e(x_i)$ signifies the embedding of the case (x_i) and $f(x_i)$ epitomizes the measured mathematical features of (x_i). The concatenated input embedding (x_i) is calculated as: $x_i' = [e(x_i), f(x_i)]$. The method understands the series of instances by utilizing location encoded that inserts position data to the sequence of input. The self-attention-based transformer method effectively handles the series of inputs, gathering both positional information and feature representation by implementing location encoding and input embedding phases.

Transformer Encoder: A sequence of embedded features and positional encoding. To detect relationships and obtain significant representation from the series of inputs, use a load of the Transformer encoder layer. It is calculated as, $E(i) = [e_1(i), e_2(i), \ldots, e_n(i)]$, whereas every $e(i)$ signifies the representation of output for the equivalent location in the series.

Self-Attention: The Transformer method's ability to discover links among features, which go away from adjacency of sequence is a new stimulating characteristic of this method. The self-attention method was employed to acquire the relationship amid numerous points in every Transformer Encoder layer. The resemblance amongst the vectors of key and query is employed to define the attention weight (AW) for every point. The AW demonstrates the relative significance of every location. The AW computation is given below:

$$AW = soft - \max\left(\frac{Q_u^R K_e}{\sqrt{dk}}\right) \tag{1}$$

Here, K_e and Q_u denotes the key and query related to input embedding $(e_1, e_2, \dots, e_i)$. By employing the attention weight matrix AW, we build a weighted sum of the value vector as the later value vector:

$$Attention\,(AW, V_a) = AW \cdot V \tag{2}$$

Here, V_a signifies the input embedding. Moreover, we tackle the problem of the variable length by using the similar padding mask model as the Transformer. Over the embedding layer usage, we hold the core of every feature in the assumed input series x.

$$E_{vi} = V_{vi} \cdot x \tag{3}$$

The visited embedding E_{vi} and learning parameters V_{vi} are intricate in the procedure. This layer acts as the drive for incorporating and preserving sequential data into the method.

Follow the self-attention tactic to improve the representation via using feed-forward neural networks to every point distinctly. An activation function of non-linear splits the dual linear layers, which compose the feed-forward networks. Attach the input features to the output of the self-attention device and the feed-forward network output to generate the remaining connections. After that, the features of every sub-layer are regularized utilizing the layer normalization process.

The Transformer Decoder layer output feeds over a fully connected (FC) layer. To define the likelihoods of the last output, utilize the activation function of softmax.

$$y = soft\max\,(V_a + e) \tag{4}$$

3.3. Hyperparameter Tuning Process

The MOA approach is employed for the hyperparameter tuning process EAMD-SATM approach [18]. The MOA model is a metaheuristic technique simulated by the population; it addresses the optimization problems through the iteration process. The MOA includes candidate solutions in the problem space. The population is initialized based on Eq. (6) at the initial stage of the optimization process and modelled using a matrix in Eq. (5). The values of decision variable can be described by all the members based on the search space location. Additionally, the search ability of population to discover an optimal solution.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_j \\ \vdots \\ X_N \end{bmatrix}_{N\times d} = \begin{bmatrix} x_{1,1} & x_{1,i} & x_{1,d} \\ x_{j,1} & x_{j,i} & x_{j,d} \\ x_{N,1} & x_{N,1} & x_{N,1} \end{bmatrix}_{N\times d} \tag{5}$$

$$x_{j,i} = lb_i + rand(0,1) \times (ub_i - lb_i), j = 1,2, \dots., N, i = 1,2, \dots., d \tag{6}$$

Now, X denotes the population matrix, N is the number of population participants, d is the quantity of decision parameters, $X_j = x_{j,1}, \dots., x_{j,i}, \dots., x_{j,d}$ denotes the j^{th} solution of a candidate, the $x_{j,i}$ is the i^{th} a variable that the random function within [0,1], and ub_i and lb_i are upper and lower limitations of the decision i^{th} parameters.

The members of the population provide solutions to these problems, which is enhanced. The function of cost is described by the population individual for the decision variable.

$$V = \begin{bmatrix} V_1 \\ \vdots \\ F_j \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} V(X_1) \\ \vdots \\ F(X_j) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{7}$$

Now, V and V_j are the vector cost function value for the j^{th} individuals.

The value of cost function evaluates the solution quality generated by the population members. In every iteration, the individual locations and best individual of the population are upgraded. Therefore, the best individual in population resolves the problems in the last iteration.

Consider the mathematical modeling of raising children by mother through interaction. In the MOA, the population can be upgraded in three different stages as follows:

Education or exploration stage: This stage is based on the children's education in the proposed MOA. The objective is to increase the global search and exploration capabilities by making huge alterations in the distinct position. Since the behavior of mother's during the children's training is noble, and considered as fittest member. A new location for every individual is generated by using Eq. (9). If the values of the benchmark function increase in the updated position, then it is demonstrated as a corresponding member place as follows

$$x_{j,i}^{P1} = x_{j,i} + rand(0,1) \times (D_i - \ rand\ (2)\ \times x_{j,i}) \tag{8}$$

$$X_j = \begin{cases} x_j^{P1}, V_j^{P1} \leq F_j, \\ X_j, else, \end{cases} \tag{9}$$

Where D_i is the i^{th} size of the mother's position, $x_{j,i}$ is the i^{th} size of j^{th} individual location, X_j and X_j^{P1} are the updated locations calculated for the j^{th} individuals, $x_{j,i}^{P1}$ shows its i^{th} dimension, F_j^{P1} is the cost function value, and the $rand$ is a uniformly generated integer within [0,1] and [1,2].

Advice or Exploration Stage: A mother's responsibility is raising the children, which is of great importance to guide their children and not allow them to misbehave. This allows global search and exploration by creating huge alterations in the member location. If an individual position in the population is exceeded by other individuals with the highest value of cost function is assumed as a rare method that must be prohibited. Every individual's bad behavior (BB_i) is determined by the comparative review of the cost function value. The members are arbitrarily selected from the set of worst behaviors for X_i, using a uniform distribution. Firstly, a new location is generated for each individual using Eq. (10). This keeps the child far from the bad behavior. If there is an increase in cost function value, then a new position replaces the earlier one based on Eq. (11).

Exploitation and upbringing stage: Mother uses dissimilar approaches to encourage their kids to enhance their abilities in the learning method. On the other hand, upbringing assists individuals to recover their capability in exploitation and local search by making small changes in individual locations. To stimulate this, a new location is generated for every individual based on the behavior development of children. If the cost function value improves, it replaces the preceding location, as follows:

$$x_{j,i}^{P3} = x_{j,i} + \left(1 - 2 \times rand(0,1)\right) \times \frac{ub_i - lb_i}{t} \quad (10)$$

$$X_j \begin{cases} X_j^{P3}, V_j^{P3} \leq V_i; \\ X_j, else, \end{cases} \quad (11)$$

Where X_j^{P3} denotes the updated location, which is evaluated for the j^{th} individuals, $x_{j,i}^{P3}$ is its i^{th} dimension, V_j^{P3} denotes the cost function value, the $rand$ is a randomly generated integer within [0,1]; t denotes, the iteration counter.

The fitness range is the extensive aspect influencing the achievement of the MOA method. The hyperparameter assortment method includes the solution-encoded method to compute the value of the candidate solution. In this research work, the MOA esteems accuracy as the main feature for inventing the fitness function that is expressed below.

$$Fitness \ = \ \max(P) \quad (12)$$

$$P = \frac{TP}{TP + FP} \quad (13)$$

Whereas, TP and FP portray the true and false positive values.

4. Experimental Validation

The performance assessment of the EAMD-SATM approach is analyzed using the Andro-AutoPsy database [19, 20]. This database has 7500 samples with 2 class labels as specified in Table 1.

Table 1 s
Details on Dataset

Classes	**No. of instances**
Benign	5000
Malware	2500
Total instances	**7500**

Table 2, reports an android malware detection result of EAMD-SATM technique under 70%TRAP and 30%TESP. In Fig. 3, the average outcomes presented by the EAMD-SATM approach on 70% of TRAS is emphasized. This figure displayed that the EAMD-SATM system attains efficient results. With 70%TRAP, the EAMD-SATM approach achieves average $accu_y$ of 97.30%, $prec_n$ of 97.60%, $reca_l$ of 96.33%, $F_{measure}$ of 96.93%, and MCC of 93.93%.

In Fig. 4, the average outcomes provided by the EAMD-SATM technique on 30% of TESP are underlined. The figure portrayed that the EAMD-SATM system obtains proficient results. With 30%TESP, the EAMD-SATM methodology achieves average $accu_y$ of 97.69%, $prec_n$ of 97.82%, $reca_l$ of 96.93%, $F_{measure}$ of 97.36%, and MCC of 94.75%.

Table 2
Android malware detection outcome of EAMD-SATM technique under 70%TRAP and 30%TESP

Classes	$Accu_y$	$Prec_n$	$Reca_l$	$F_{measure}$	MCC
			TRAP (70%)		
Benign	97.30	96.76	99.25	97.99	93.93
Malware	97.30	98.44	93.41	95.86	93.93
Average	**97.30**	**97.60**	**96.33**	**96.93**	**93.93**
			TESP (30%)		
Benign	97.69	97.46	99.14	98.29	94.75
Malware	97.69	98.18	94.72	96.42	94.75
Average	**97.69**	**97.82**	**96.93**	**97.36**	**94.75**

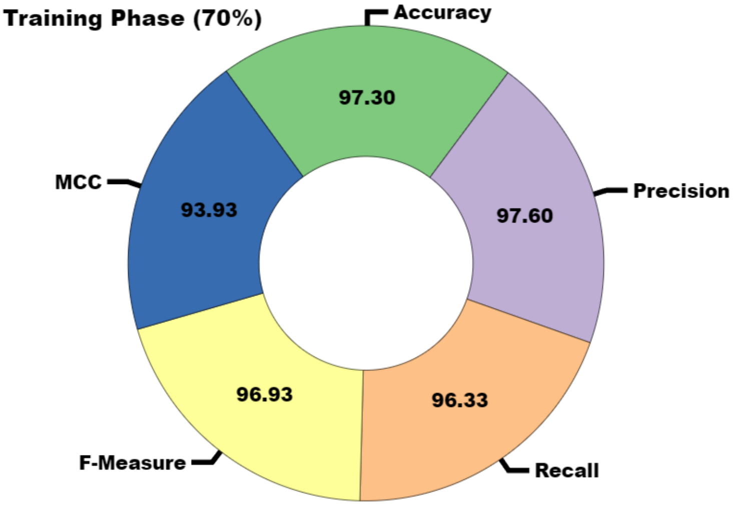


Figure 3: Average outcome of EAMD-SATM technique under 70% TRAP

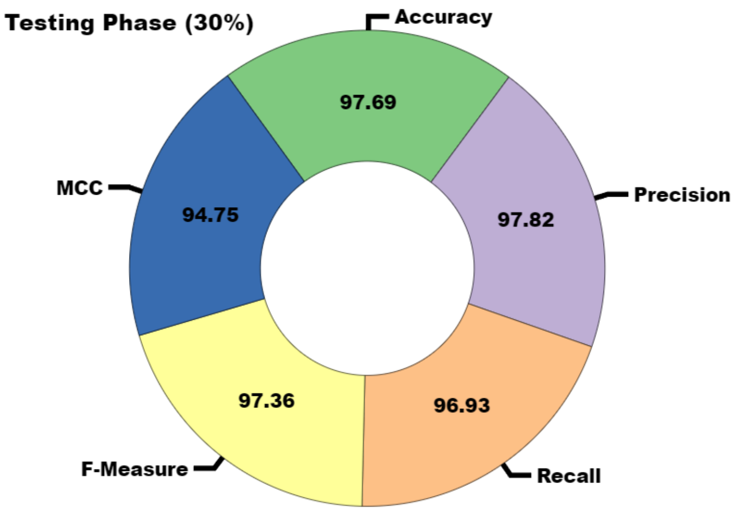
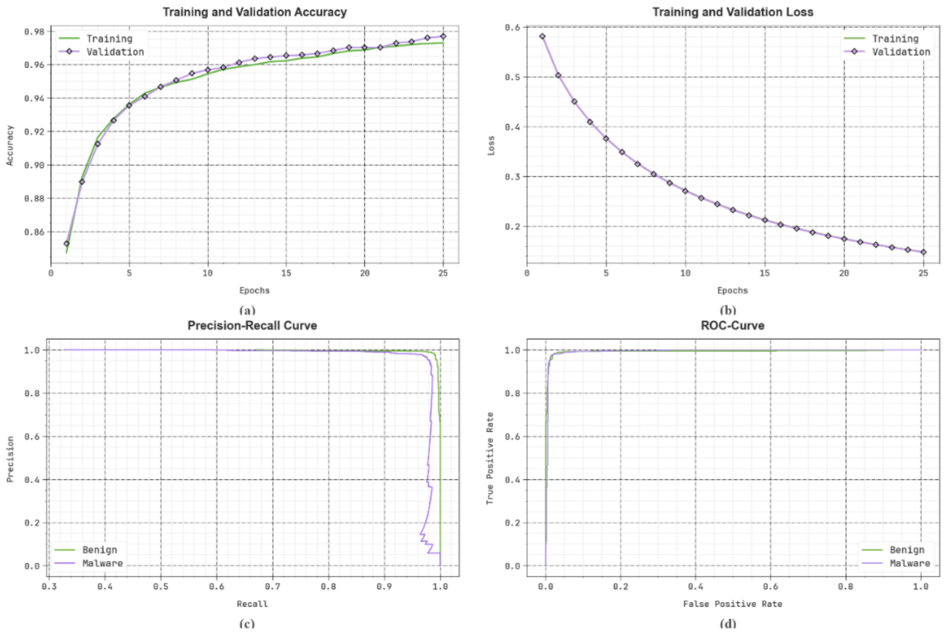


Figure 4: Average outcome of EAMD-SATM technique under 30% TESP



(a)

(b)

(c)

(d)

Figure 5: EAMD-SATM technique Curves of (a) Accuracy (b) Loss (c) PR and (d) ROC

Fig. 5 illustrates the classifier outcomes of EAMD-SATM approach. Fig. 5a shows the accuracy study of the EAMD-SATM approach. This figure shows that the EAMD-SATM method achieves growing values over increased epoch counts. Then, Fig. 5b demonstrates the loss study of the EAMD-SATM technique. The outcomes specify that the EAMD-SATM methodology achieves adjacent outcomes of training and validation loss. Fig. 5c reported the study of PR in the EAMD-SATM system. The outcomes indicated that the EAMD-SATM method outcomes in growing PR values. At last, Fig. 5d shows the ROC examination of the EAMD-SATM technique. The figure represented, that the EAMD-SATM approach results in enhanced values of ROC.

Table 3
Comparative analysis of EAMD-SATM technique with other models

Algorithm	$Accu_y$	$Prec_n$	$Reca_l$	$F_{measure}$
EAMD-SATM	97.69	97.82	96.93	97.36
J48 Model	96.80	95.20	95.42	97.24
Decision Table	94.60	91.60	95.52	97.07
Naive Bayes	69.10	61.00	96.02	96.71
SMO Classifier	96.40	94.60	95.07	96.26
Logistic Algorithm	96.30	94.40	95.74	96.58
AdaBoostM1	88.40	81.70	91.83	94.25

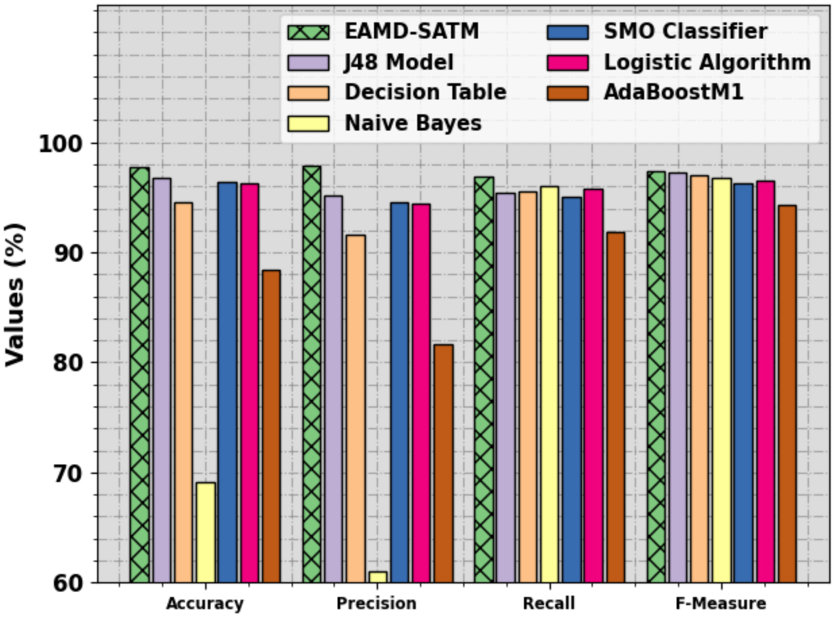


Figure 6: Comparative analysis of EAMD-SATM technique with other models

In Table 3 and Fig. 6, the efficient results of the EAMD-SATM technique were experienced compared with recent techniques [21-23]. The outcomes indicated, that the Naive Bayes & AdaBoostM1 method displayed inferior outcomes. Together, the J48, Decision Table, SMO, and Logistic Algorithm approaches have depicted nearer results. However, the EAMD-SATM model handled reporting the highest outcomes with higher $accu_y$, $prec_n$, $reca_l$, and $F_{measure}$ of 97.69%, 97.82%, 96.93%, and 97.36%, appropriately.

5. Conclusion

This study proposes an EAMD-SATM model has been developed. The presented EAMD-SATM model classifies and recognizes the Android malware efficiently and accurately. To attain this, the EAMD-SATM approach endures a min-max approach utilizing data pre-processing at the initial stage. Furthermore, the EAMD-SATM method employs SA-T technique for the detection of Android malware. To improve the solution of the SA-T technique, the EAMD-SATM algorithm applies the IMO technique for the parameter tuning method. The simulation validation of the EAMD-SATM technique can be established on a benchmark Android malware dataset. The experimental results highlighted the important performance of the EAMD-SATM approach in the Android malware recognition method

References

[1] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, H. Liu, A Review of Android Malware Detection Approaches Based on Machine Learning, IEEE Access,8, (2020) 124579–124607.

[2] E.T. Elkabbash, R.R. Mostafa, S.I. Barakat, Android malware classification based on random vector functional link and artificial Jellyfish Search optimizer, PloS one, 16.11, (2021).

[3] M. Ragab, Hybrid firefly particle swarm optimisation algorithm for feature selection problems, Expert Systems, 41.7(2024) e13363.

[4] E. M. Dovom, A. Azmoodeh, A. Dehghantanha, D. E. Newton, R. M. Parizi et al., Fuzzy pattern tree for edge malware detection and categorization in IoT, Journal of Systems Architecture,97, (2019) 1–7.

[5] R.M. Sharma, C.P. Agrawal, MH-DLdroid: A Meta-Heuristic and Deep Learning-Based Hybrid Approach for Android Malware Detection, Int. J. Intell. Eng. Syst, 15, (2022) 425-435.

[6] A. S. Alkarim, A. S. Al-Ghamdi, M. Ragab, Ensemble Learning-based Algorithms for Traffic Flow Prediction in Smart Traffic Systems, Engineering, Technology & Applied Science Research, 14.2 (2024) :13090-4.

[7] O.A. Alzubi, J.A. Alzubi, A.M. Al-Zoubi, M.A. Hassonah, U. Kose, An efficient malware detection approach with feature weighting based on Harris Hawks optimization, Cluster Computing, 25.4, (2022) 2369-2387.

[8] S. Zhao, S. Li, L. Qi, L. D. Xu, Computational Intelligence Enabled Cybersecurity for the Internet of Things, IEEE Transactions on Emerging Topics in Computational Intelligence, 4.5, (2020) 666-674.

[9] P. Ahirao, Proactive Technique for Securing Smart Cities against Malware Attacks Using Static and Dynamic Analysis, International Research Journal of Innovations in Engineering and Technology, 5.2, (2021) 10.

[10] S.K. Smmarwar, G.P. Gupta, S. Kumar, P. Kumar, An optimized and efficient android malware detection framework for future sustainable computing, Sustainable Energy Technologies and Assessments, 54, (2022) 102852.

[11] M. A. Ferrag, M. Ndhlovu, N. Tihanyi, L.C. Cordeiro, M. Debbah, T. Lestable, N.S. Thandi, Revolutionizing Cyber Threat Detection With Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices, in IEEE Access,12, (2024) 23733-23750.

[12] A. Girones De La Fuente, Enhancing Malware Detection in Executable Files using LST., BiLSTM-based Deep Learning Models with Word Embedding, Doctoral dissertation, Politecnico di Torino, (2023).

[13] M.T. Islam, W. Rahman, M.S. Hossain, K. Roksana, I.D. Azpíroz, R.M. Diaz, I. Ashraf, M.A. Samad, Medicinal Plant Classification Using Particle Swarm Optimized Cascaded Network, IEEE Access, 12, (2024) 42465-42478.

[14] F. Ullah, G. Srivastava, S. Ullah, K. Yoshigoe, Y. Zhao, NIDS-VSB: Network Intrusion Detection System for VANET Using Spark-Based Big Data Optimization and Transfer Learning, in IEEE Transactions on Consumer Electronics,70.1, (2024) 1798-1809.

[15] C. Liu, B. Li, X. Liu, C. Li, J. Bao, Evolving malware detection through instant dynamic graph inverse reinforcement learning, Knowledge-Based Systems, (2024) 111991.

[16] M. Mazziotta, A. Pareto, Normalization methods for spatio-temporal analysis of environmental performance: Revisiting the Min–Max method, Environmetrics, 33.5, (2022).

[17] A.U. Rahman, Y. Alsenani, A. Zafar, K. Ullah, K. Rabie, T. Shongwe, Enhancing heart disease prediction using a self-attention-based transformer model, Scientific Reports, 14.1, (2024) 514.

[18] S. Li, X. Fang, J. Liao, M. Ghadamyari, M. Khayatnezhad, N. Ghadimi, Evaluating the efficiency of CCHP systems in Xinjiang Uygur Autonomous Region: an optimal strategy based on improved mother optimization algorithm, Case Studies in Thermal Engineering, 54, (2024) 104005.

[19] https:// ocslab.hksecurity.net/andro-autopsy

[20] J.-W. Jang, H. Kang, J. Woo, A. Mohaisen, H. K. Kim, AndroAutoPsy: Anti-malware system based on similarity matching of malware and malware creator-centric information, Digit. Invest. 14, (2015) 17–35.

[21] F. Kateb, M. Ragab, Archimedes Optimization with Deep Learning Based Aerial Image Classification for Cybersecurity Enabled UAV Networks, Computer Systems Science and Engineering , 47.2 (2023) 2171-2185.

[22] H. Alamro, W. Mtouaa, S. Aljameel, A. S. Salama, M. A. Hamza, A. Y. Othman, Automated Android Malware Detection Using Optimal Ensemble Learning Approach for Cybersecurity, in IEEE Access,11, (2023) 72509-72517.

[23] L. A. Maghrabi, S. Shabanah, T. Althaqafi, D. Alsalman, S. Algarni, A. AL-Ghamdi, M. Ragab, Enhancing cybersecurity in the internet of things environment using bald eagle search optimization with hybrid deep learning, IEEE Access, 12, (2024) 8337-8345.