scientific reports

Check for updates

OPEN

Prediction of android ransomware with deep learning model using hybrid cryptography

K. R. Kalphana[1], S. Aanjankumar[2], M. Surya[3], M. S. Ramadevi[4], K. R. Ramela[5], T Anitha[6], N. Nagaprasad[7] & Ramaswamy Krishnaraj[8,9]✉

In recent times, the number of malware on Android mobile phones has been growing, and a new kind of malware is Android ransomware. This research aims to address the emerging concerns about Android ransomware in the mobile sector. Previous studies highlight that the number of new Android ransomware is increasing annually, which poses a huge threat to the privacy of mobile phone users for sensitive data. Various existing techniques are active to detect ransomware and secure the data in the mobile cloud. However, these approaches lack accuracy and detection performance with insecure storage. To resolve this and enhance the security level, the proposed model is presented. This manuscript provides both recognition algorithms based on the deep learning model and secured storage of detected data in the cloud with a secret key to safeguard sensitive user information using the hybrid cryptographic model. Initially, the input APK files and data are preprocessed to extract features. The collection of optimal features is carried out using the Squirrel search optimization process. After that, the Deep Learning-based model, adaptive deep saliency The AlexNet classifier is presented to detect and classify data as malicious or normal. The detected data, which is not malicious, is stored on a cloud server. For secured storage of data in the cloud, a hybrid cryptographic model such as hybrid homomorphic Elliptic Curve Cryptography and Blowfish is employed, which includes key computation and key generation processes. The cryptographic scheme includes encryption and decryption of data, after which the application response is found to attain a decrypted result upon user request. The performance is carried out for both the Deep Learning-based model and the hybrid cryptography-based security model, and the results obtained are 99.89% accuracy in detecting malware compared with traditional models. The effectiveness of the proposed system over other models such as GNN is 94.76%, CNN is 95.76%, and Random Forest is 96%.

Keywords Android, Ransomware, Deep learning, Squirrel search optimization, AlexNet, Hybrid cryptography, Blowfish

Due to the huge range of sensitive information stored both on the cloud and devices on transferring through the network, the detection of malware, specifically ransomware become a hot topic nowadays[1]. Hackers have been focused on making malicious mobile applications that could exploit the environment, devoid of existing standards on protection utilized by Google beforehand the developer releases the application[2]. A new kind of Malware like Android ransomware become an emergent one in the mobile segment. So as to protect from this application that is malicious, there is a need for an extra protection layer individually in each Android mobile device. The attack that looks like ransomware employs some stages set for infecting the system which initiates

[1]Department of Agricultural Engineering, Mahendra Engineering College, Namakkal, Tamil Nadu 637503, India. [2]School of Computing Science and Engineering (SCOPE), VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh 466114, India. [3]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India. [4]Department of Computer Science and Engineering, Mount Zion College of Engineering and Technology, Pudukkottai, India. [5]Department of Electrical and Electronics Engineering, Ultra College of Engineering and Technology, Madurai, India. [6]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India. [7]Department of Mechanical Engineering, ULTRA College of Engineering and Technology, Madurai, Tamil Nadu 625104, India. [8]Department of Mechanical Engineering, College of Technology and Engineering, Dambi Dollo University, Dembi Dolo, Ethiopia. [9]Center for Global Health Research, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, India. ✉email: prof.dr.krishnaraj@dadu.edu.et

with infection and distribution of the device. This malware searches for files to infect them. It thus encrypts the files, threatens exposure, and requests ransom for affecting the sensitive data of victims for non-payment cases. The ransomware is responsible for encrypting the victim's file from their computer in a short time so as to hijack data & ask for ransom. Typical models to discover the signature of malware do not function since the virus has consistent evolution, thus making virus action detection a challenging one. Because of this threat's several signatures, various existing signature-based detection models are not working well for detecting Android ransomware[3].

As a step to overcome those challenges, security at the cloud storage is employed and promoted by cloud providers. Cloud computing refers to a secure way for remote servers to access and store information. After that, the cloud makes it possible to access information at any cost of time over any other medium[4]. Several challenges confront the security of the cloud like accessing user control, interface safety, data separation, & data stored safely which identifies cloud storage providers or end users who might manage encryption & decryption. Once the user utilizes the cloud to store data and outcomes, they become vigilant. Also, data stored should be encrypted before it is transmitted to the cloud. This might enable consumers and users to access resources of the cloud that are shared securely and effectively which is offered that whole data are thus encrypted using various encryption approaches and models. The utilization of encryption models is regarded as a better means of securing data in the cloud[5]. Also, the detection of classification of data as malware or normal is essential before storing them on a cloud server.

A technique of cryptography is employed widely to protect public cloud data. This allows customers to access the mutual administration of the cloud in an effectual manner as whole data is protected[6]. The cryptography technique makes the plain text an unpredictable one which limits the view of data that are being transferred. Encryption is a better cryptographic model that does not compromise security & sensitive data's transferring speed. The basic component of encryption is a defense-in-depth strategy as it could mitigate the weakness of the mechanism in primary access control[7]. If the data is encrypted with a string key, it would be intricate for attackers to decrypt them as long the information is not in a similar system as such your data. Encryption mechanism aids in preventing confidential data to fall into the wrong hands. In other words, sensitive data confidentiality is thus preserved by the process of encryption. The encryption model use includes managing encrypting keys securely and safely. Also, there are three kinds of security processes like data encryption, key generation, and decryption of data.

For cloud storage and security, there were several cryptography algorithms[8], among them the most recent and effective ones Blowfish and Homomorphic ECC encryption are ensembled in this proposed work. Various researchers conducted studies to prove the efficiency of various existing cloud security approaches. The major intention of this work is to take probable benefits from the traditional computer-aided diagnosis (CAD) -aided Deep Learning (DL) model for detecting ransomware and storing data securely in the cloud.

Motivation of this research

The main motivation of this study is to define the evolution of Android malware and its types. mainly the ransomware, which targets all permissions inside Android mobile devices and encrypts all the data inside. To recover data, we are processing an advanced hybrid cryptography technique with the combination of ECC and Blowfish to secure storage data, while a deep learning model is used to detect the malware and provide data security to mobile devices. The final result of this study is to ensure the safety and privacy of information stored on mobile devices.

Methodology

- Improve feature selection with SSO-Squirrel Search Optimization; unlike the traditional model, this one uses a metaheuristic algorithm to analyze the behavior of the ransomware with more relevant features.
- With the use of adaptive deep saliency The AlexNet classifier's extraction capabilities are improved, and there is better generalization of new kinds of Android ransomware data from the dataset with increased accuracy in detection.
- To provide strong security with small keys and fast encryption, hybrid cryptography is used by combining ECC and Blowfish encryption.
- The proposed model is to produce enhanced model evaluation with 99% accuracy in the detection of malware and effective visualization compared to the already existing traditional model in Android ransomware detection techniques.

The remaining section of this article is organized as follows: "Related works" section is the description of various traditional models employed related to cloud storage security and detection of Android ransomware. The suggested work design is described briefly in "Proposed work" section. The assessment of performance and their outcomes are projected in "Experimental results" section. Finally, the conclusions are made in "Conclusion and future works" section.

Related works

A short summary of various existing models employed before related to Android ransomware detection and cloud storage security is provided here.

The work of Benil et al.[9] suggested a scheme called certificate-less Elliptical Curve Aggregate signature of cryptography for the purpose of auditing and public verification in a server of the medical cloud. It offers security in E-healthcare with the use of authorized technology blockchain. To encrypt medical data elliptic curve

cryptography (ECC) was used together with the certificate-less signature aggregate scheme for producing digital signatures to share & store them in the cloud. The blockchain technique guarantees secured storage of them in a cloud server.

A framework was developed by Velmurugadass. et al.[10] for checking the specific data actions and thus creates a cloud cloud-dependent software-defined network (SDN). SDN comprises of hundreds of mobile nodes, controllers based on blockchain and open flow switch, investigator, authentication server, and cloud server. Initially, registered users having AS cloud receive secret keys depending on Harmony search optimization (HSO) from AS. Using the Elliptical curve integrated encryption scheme approach, mobile nodes packets were thus encrypted and thus forwarded to a cloud server. A scheme of SHA-256 cryptographic hash algorithm was employed for preserving evidence of blockchain gathered from the signature and data of the user. The performance analysis shown performs better outcomes in terms of response time, accuracy, entire security change parameters, and enhancing throughput.

A method was presented by Thirumalai et al.[11] with an effective, non-sharable public key exponent secured scheme that employs a non-linear equation of Diophantine. By this, high security was attained. Not like other schemes such as ESR, and RS, the presented model ENPKESS includes three encryption stages together with decryption in both stages. Due to this, secret key extraction was complex for resolving public components. The Knapsack model ensures higher security in the cloud system. From performance analysis, it was proven that the ENPKESS model offers higher security than other existing approaches.

A cloud-based security model was presented by Masud et al.[12] for the E-healthcare system. Since, the cloud systems were scalable, inexpensive, and offered a huge access range to the patient's health records electronically by promoting security constraints. The technique offers a secured means of port for participants thereby avoiding unlicensed users from accepting cloud storage information. With the use of the key derivation function, multiple number of keys were formed in this structure. This ensures the ciphering of information in an end-wise manner thereby preventing misuse. Depending on integration and integrity among stakeholders, access rights were provided to cloud services. This model was suited for confidentiality and data security.

A method of privacy preservation and an untraceable scheme was employed by Shen et al.[13]. By the design of a proximal re-encryption approach with ORAM obvious RAM in a cloud environment, multiple users were supported in data sharing. The phase of key exchange was employed among group associates and proxies in this model for acquiring keys. By the retrieved ciphertext through the proxy re-encryption phase, group members are able to control access & thus perform sharing of data securely. A binary tree model linked with a one-way circular table was thus operated to access secret data and intractability.

In Hedaia et al.[14], a non-traditional scheme for authentication called Bio-CAPTCHA was employed. The presented technique offers random vice-dependent challenges of passwords that change each time dynamically when the user tries to log in, thereby ensuring the probability of unauthorized access. The major objective was to suggest such new solutions for the authentication of users dedicated to multi-level user authentication. It employs the authorization of users in a distributed framework to permit access to cloud databases & repositories.

Kavin et al.[15], a hybrid framework was suggested for cloud computing security. The algorithms like AES, DAS, and stenography were emphasized in this model. By using this approach, data integrity, authenticity, and security were probable. For offering high protection, DSA, and AES were combined with steganography.

The artificial intelligence techniques were made in this model by Movassagh et al.[16] for the issue related to the input coefficient. This in turn offers the demand for applications using meta-heuristic approaches. It seems to be low control in weed optimization performance in terms among typical cloud models. In the issues, of application security, open web security was identified which covers multi-tenancy, configurability, and scalability.

The author of the work Orantes Jiménez et al.[17] presented a review regarding the risks related to security that signifies cloud threat as the model of service delivery nature such as Hill, Vernan encryption, Vignere, Rail fence cipher, proxy re-encryption, and Playfair. The significant thing regarding the algorithm of cryptography was the security of data. The advantages cover flexibility, cost saving, scalability, reliability, manageability, backup, recovery, and mobility. The issues include data integrity, confidentiality, malicious insiders, and transmission of data.

In a model proposed by Ogiela[18], a new approach was presented for creating a multilevel advanced authentication protocol using hybrid CAPTCHA codes. Such a code may define a new cognitive class of CAPTCHA that requires special skills and knowledge from a user at the time of the verification process offering proper authentication. The main objective was to define how various graph formalities and linguistics might be employed to secure data division and management. The structure of the graph represents huge probabilities of application in the area of cloud computing. The suggested model was integrated with the DL approach for classifying complex patterns.

In Jabbar and Bhaya[19], a technique to place among cloud and user based on two stages was presented. The first one protects the cloud from various kinds of network attacks thus detecting abnormal and normal flow. The next one is classifying data user thereby encrypting them depending on their significance with the use of various encryption approaches. Also, the algorithm used was advanced encryption standard (AES), triple data encryption (3DES), & rivest cipher (RC4), for classified data encryption consistent with the significance that might be kept in the cloud in a secured manner.

A novel model to enhance data security was presented by Mohd et al.[20] by means of a mutual authentication that combines hybrid cryptography assets and DL model power to offer adaptable and robust solutions to secure data in the cloud. One such main intention of this model was to pre-trained CNN. The secured means of communication among involved parties thereby combining cryptography with faster time of execution. Analysis reveals that the hybrid model offers enhanced security than other models. Highlights of different traditional models are illustrated in Table 1.

Ref	Year	Method used	Accuracy	Dataset
Jabbar and Bhaya[19]	2023	LR & SGD	98%	UNSW-NB15 & BBC dataset
Attou et al.[21]	2023	RF	98.3	Bot-IoT
Ahmad et al.[22]	2022	NB, RF, KNN, and SVM	92.0%	–
Singh et al.[24]	2023	ResNet50 & VGG 16	99.1% for testing	Ransomware attack dataset
Omar et al.[23]	2022	ESOML-IDS	Denoising Autoencoder 83.09%	UNSW-NB 15
Omar et al.[25]	2023	OELSTM-MDC	97.14%	–
Omar et al.[33]	2021	SVM HHO	94%	CICmalanal2017
Alzubi et al.[35]	2024	CNN and LSTM	Above 90%	CSE-CIC-IDS-2018
Movassagh et al.[16]	2023	ANN	–	–

Table 1. Related work highlights on different traditional models.

Proposed work

A thorough explanation of the suggested design is described in this section. Primarily, the input APK files/ data are preprocessed to extract features. The selection of optimal features is supported out by means of the Squirrel search optimization (SSO) process. After that, the DL-based model-Adaptive deep saliency AlexNet classifier is presented to detect and classify data as malicious or normal ones. The detected data which are not malicious are stored in a cloud server. For secured storage of data in the cloud, the hybrid cryptographic model (Hybrid Homomorphic ECC & Blowfish) is employed which includes key computation and key generation process. The cryptographic scheme includes encryption and decryption of data after which the app response is found to attain a decrypted result upon user request. The illustration of the whole manuscript workingflow is shown in Fig. 1.

Experimental setup

The evaluation setup of the proposed system is expressed here by taking the online available dataset as https://github.com/harrypro02/Android-Malware-Permission-Based-Dataset', and maldroid-2020, which may consist of different permissions for Android malware identification on mobile devices with different parameters such as storage, image, opcode, system call, and all permissions inside the mobile device. Then the preprocessing is handled with the model to train the DL model from the available dataset. To get the desired result, the dataset may consist of 15,000 entries in 5 rows and 1204 columns with malware data, and the normal dataset is preprocessed into the training model. After training, feature extraction is done with the SSO optimization algorithm to improve model performance with an efficient learning rate of 0.01 and L2 regularization to overcome the loss after feature extraction. Cryptographic key management is processed with homomorphic ECC and Blowfish encryption to ensure security is maintained throughout the process of decrypting the affected data processed by the ransomware. After encryption, model performance is analyzed with 50 epochs of training and 80/20 testing using adaptive deep saliency. AlexNet is configured with a dense layer and an Adam optimizer with a Relu activation function to capture Android malware with high accuracy compared with other traditional deep learning models available. Finally, this model ensures the combined deep learning and cryptographic methods work very well in detecting Android malware with high accuracy, and the scientific design of the proposed model is expressed in the below sections.

Input data cleaning

At first, the input data (application to be installed) is taken and preprocessed to remove redundant and unnecessary data. The preprocessing in this model aims to evaluate the capability to protect against the embedded ransomware code in Android apps. To attain this, a special key for preprocessing the bytecodes from Android apps is used and exploited as a structure. A hybrid cryptography model was employed to determine significant features for the finding of Android malware. Before employing the detection algorithm, preprocessing is carried out for a dataset index format. This approach takes care of converting the process of dex files to a suitable APK format. This in turn includes dex-file compiling as a setup that adapts to APK. For maintaining the designing compatibility of mobile devices, modules from JVM of Omni Rom (OR) to this transformation are employed. After the completion of the conversion process, the model analyses the text segment of every APK file to extract opcode instructions. This model includes model creation for detecting ransomware and for securing cloud server data which follows the entire crypto-code. The APK image was taken for each application and thus extracts the respective bytecode from the segment of .apk. the model calculates the data occurrence and thereby sends them to the process of feature extraction.

Feature extraction & SSO (squirrel search optimization) optimization-based selection

The classification of malware depending on grey-scale image extraction is a new approach. This has proved to be an effective tool for static analysis. It is the image that is expressed in grey color. According to the logarithmic relationship, the brightness from white to black color is thus divided into 256 grades. Various physical data from graphs could cause a respective difference in greyscale, and textures which confirms the reflection in the visual field. For exploiting the malware texture difference, an interactive disassembler was employed first (IDA) for attaining binary files into smaller units each one of them comprises of eight bits and is thus converted to unsigned

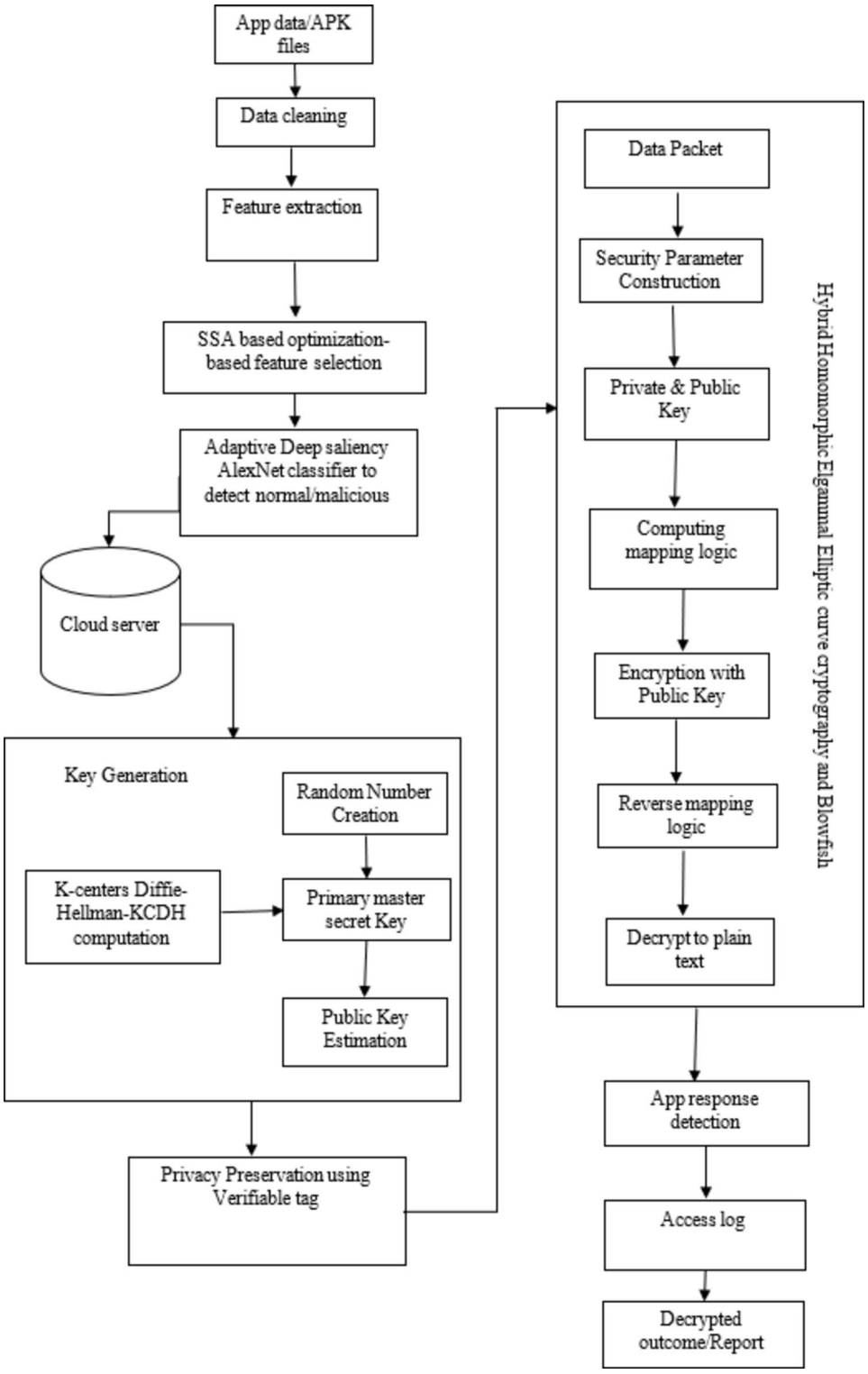

Fig. 1. Schematic flow of suggested design.

integer format in a range 0–255. In a grey-scale image, 0 and 255 signify white and black correspondingly. At last, the file transformed is thus mapped to a matrix termed 'grey scale matrix'. The matrix width is typically initialized to 2n. In this model, n is equal to 8. Moreover, the matrix is adopted to feature expression. So as to adopt this, the grey-scale matrix is thus mapped as a one-dimensional vector termed 'grey scale vector'.

The features extracted are then subjected to an optimal selection of features so as to select optimal ones. The algorithm of SSO updates the position of the individuals as per the present season, the kind of individuals, and the predator's appearance.

Initialization of population: Assume $\mathcal{N}$ as the number of individuals, and SS_U and SS_L are the bounds of exploration space. As per the formula, the individuals are produced randomly in Eq. (1):

$$SS_I = SS_L + \mathcal{R}(1, D) \times (SS_U - SS_L) \tag{1}$$

SS_I signifies ith specific ($i = 1, \ldots, \mathcal{N}$), $\mathcal{R}$ signifies the random number among 0 to 1 & D denotes the problem dimension.

Population classification. On taking the minimization issue into consideration, SSO needs one squirrel at every tree, whereas assuming the total number of squirrels are $\mathcal{N}$. The population's fitness function is thereby ranked in an ascending order. The squirrels are segregated into 3 kinds: Squirrels located at hickory trees S_H, squirrels at acorn trees S_A and squirrels at normal trees S_N. To find the finest food cause, the terminus of S_A is S_H and terminus of S_N is determined randomly as whichever S_H or S_A.

Position updation. The squirrel's position is thus updated in Eqs. (2 and 3).

$$\begin{cases} SS_I^{t+1} = SS_I^t + \mathcal{G} \times \mathcal{C} \times \left(SS_H^t - SS_I^t\right) & if\ \mathcal{R} > \mathcal{P}_{AP} \\ random\ location & Otherwise \end{cases} \tag{2}$$

$$\begin{cases} SS_I^{t+1} = SS_I^t + \mathcal{G} \times \mathcal{C} \times \left(S_{AI}^t - SS_I^t\right) & if\ \mathcal{R} > \mathcal{P}_{AP} \\ random\ location & Otherwise \end{cases} \tag{3}$$

$\mathcal{R}$ designates a random number & 't' signifies the current iteration. $\mathcal{P}_{AP}$ denotes the probability of hunter arrival whose rate is 0.1. If $\mathcal{R} > \mathcal{P}_{AP}$, then there will be predator absence & the squirrel slides into the forest for food. $\mathcal{R} \leq \mathcal{P}_{AP}$, the hunters might appear & squirrels have to decrease the activities of food forage since they are at risk. At that time, the squirrel's positions are randomly relocated. $\mathcal{C}$ specifies the constant of value 1.9 & $\mathcal{G}$ signifies gliding distance. S_{AI}^t denotes randomly selected individual squirrels from S_A. Gliding distance is considered as in Eq. (4)

$$\mathcal{G} = \frac{\mathcal{G}_H}{\tan(\theta) \times \mathcal{S}} \tag{4}$$

$\mathcal{G}_H$ denotes the persistent whose value is 8 & $\mathcal{S}$ is the relentless of value 18. $tan(\theta)$ signifies the sailing angle. Once the number of iterations exceeds the extreme amount of iterations, the individual's movement is stopped. Or else, the above steps get repeated.

Adaptive deep saliency AlexNet classifier

Once the selection of features is made, the mechanism of classification is employed so as to recognize the attacks. In this effort, the organization approach is the final stage of the detection mechanism. The detection should be made before the security mechanism. For the classification process, adaptive deep saliency AlexNet classifier is employed to classify the data as malignancy or benignity labels. The dataset is subdivided into 2 stages for estimating the regions to test and train. This phase covers dataset vector training with its respective classes, whereas the output identifies whether the input image is mild or fatal. This classifier model is trained and tested with the kernel function of RBF to attain a better outcome.

The suggested Adaptive deep saliency AlexNet classifier model detects whether the data is malicious or not. The data in the words before step t of CNN architecture is too employed as the input at the time of word processing of step t. The early cell data are gathered from cells and words are thus given as inputs. The little references sense the repeated image over one cell. The cell sequence of architecture is another reference. The amount of text presented in each example of data does not turn out to be a specific value of natural language processing issues. For executing each text, the dimensions of the arrangements were reduced to value. Once the value of the arrangement is less than the desired value, the sequence is thus filled as a value. Once the sequence size exceeds the mentioned value, the remaining are rejected.

The AlexNet CNN model comprises 5 layers of convolution, two fully connected layers that are connected completely, and one recurrent layer. The layers of CNN were employed for learning middle-level patterns of visual similar to the first 5 popular layers of AlexNet seven layer. The layer of RNN is employed for learning the dependency of space among visual patterns of the middle layer. In both final, layers, 2 fully connected RNN outputs were gathered and the representation of a global image was learned. The classification of the SoftMax layer should be applied subsequently to N-way (N signifies the class number).

Start

Step 1: *Interpret the input feature data*

Step 2: *Increase contrast type of original data using contrast stretched min − max algorithm*

Step 3: *Transform the contrast stretched transformation.*

Step 4: *Segment the data*

Step 5: *Extract the features from extracted data using EPCA*

Step 6: *Assess images using training testing patterns.*

Step 7: *Use Alex net CNN classifier to classify data.*

Step 8: *Distinguish Ground truth data to check if the region is malignant or benign.*

Step 9: *Set the classifier label as* 0 *and* 1, *then*

Apply state

if $(result == 1)$

helpdlg $('data\ is\ Encrypted)$

disp $('Malware')$

else

helpdlg $('data\ is\ healthy)$

disp $('Benign')$

end

if $(choice == 3)$

close all

return

end

Step 10: *Return the results.*

Stop

Algorithm 1. Adaptive deep saliency AlexNet classifier.

Once the data is classified as attack or normal, then the normal data is stored in a cloud server. From the cloud server, the data should be encrypted with a key so as to enable secured means of cloud storage. For the secured storing in the cloud, the computation of key or key generation is carried out followed by a hybrid cryptographic approach to enable encryption and decryption process. This is explained in subsequent sections.

Computation of key using K-Centers Diffie-Hellman (KC-DH)

For the secured means of storage in cloud, the data should be encrypted and protected with key. For key computation, the approach of K-centers Diffe-Hellman (KC-DH) is employed at which the generation of key is carried to share private key with which they could change data over insecure channel. However, the private key is not unique which generates each and every data sharing transaction at private key must be random. The algorithm for this key computation approach is shown below:

Step 1: *Let the users be named Alice* a_c & *Bob* b_b

First, they agree on two prime numbers g_i & p_j.

If p_j is large (*typically at least* 512 *bits*) & g_i is a *primitive root modulo on* p_j

Step 2: *The numbers* g_i & p_j *need not be kept secret* from *other users*

Step 3: *Generate centre node large random primary number* kc_r

Step 4: $kc_r = \left(max_i \middle| min_j - 1\right) - \frac{max_i}{2}$

Step 5: a_c *chooses large* $K -$ *centre random* (kc_r) *number* a *as her private key* (p_k) & *similarly Bob chooses* a *large number* b.

Step 6: b_b *chooses similarly a large number b*

Step 7: a_c *computes* $A = g^a(mod\ \%\ p_j)$

Step 8: b_b *computes* $B = g^b(mod\ \%\ p_j)$

Step 9: a_c *send key to* b_b $T_i \underset{a_c}{\Rightarrow} \xrightarrow{paring\ key\ (p^k)} b_b$

Step 10: b_b *send key to* a_c $T_j \underset{b_b}{\Rightarrow} \xrightarrow{paring\ key\ (p^k)} a_c$

Step 11: a_c and b_b *compute their shared key* $s_{key} = g^{a\,b}(mod\ p_j)$

Step 12: a_c *compute as* $sa_{c_{key}} = B^a\ (mod\ p_j) = (\ g^b)^a\ (mod\ p_j)$

Step 13: b_b *compute as* $sb_{b_{key}} = A^b\ (mod\ p_j) = (\ g^a)^b\ (mod\ p_j)$

Step 14: a_c & b_b *can now use their shared key* (s_{key}) *to exchange shared key. Compare their own private keys (a, b)* as $s_{key} = g^{a\,b}(mod\ p_j)$ *on* $g_1\ _{p_1}\ A = g^a(mod\ \%\ p_j)$ & $B = g^b(mod\ \%\ p_j)$

Step 15: *Compute* $C_{rs=}\ g^a(mod\ \%\ p_j)\ \cup g^b(mod\ \%\ p_j)$

Step 16: *if* $C_{rs} == TRUE\ s_{key}$ *matched*

Step 17: *else return* s_{key}

Step 18: *Break* C_{rs}

Algorithm 2. KC-DH protocol for key computation.

This is the discrete logarithm issue, which is infeasible computationally for larger p. The computation of discrete number logarithm modulo p takes a similar amount roughly the same time of amount since factoring the two prime products as similar as p, which is what the RSA cryptosystem security lies on. Therefore, this protocol ECC-DH is secured roughly as RSA.

Hybrid cryptography using Homomorphic ECC & Blowfish approach

The hybrid homomorphic ECC and blowfish-dependent cryptographic scheme was suggested in this model which the multilevel encryption to exchange data between server and client in a model of public SaaS. It is

primordial to preserve confidentiality before outsourcing or sending information in both directions from the client to the cloud & vice versa. Consequently, unauthorized access by non-allowed users might be secured to prohibit security constraint threats that are coming from intruders. This hybrid model is offered in which the cloud server data uploaded is therefore encrypted using a blowfish strategy to enhance the aspect of security thus preserving data privacy. Yet, for high protection, keys used in the encryption process are handled therefore and encrypted by the ECC approach. This hybrid model not only guarantees integrity & confidentiality but also offers authenticity. The suggested model utilizes two kinds of cryptography approaches which are a symmetric approach (blowfish) and an asymmetric algorithm (ECC). Therefore, this model of hybrid model integrates two approaches to benefit the encryption process. The blowfish approach or symmetric model is thus employed to encrypt data that is kept in the cloud. Thus, the decryption process is a reverse one carried out in data outsourcing. The asymmetric model is ECC & thus employed in the management & encryption of encrypting keys. The integrity of homomorphic ECC and Blowfish scheme processes enhances security in mobile environments where ECC processes have strong security with small key sizes compared to traditional models such as RSA. The 128-bit key of ECC offers the same security level as the 1024-bit key of RSA. Similarly, Blowfish is familiar for its speed and efficiency in a small, less resource-constrained environment like Android with 64-bit blocks for data encryption. Combining these two schemes improves confidentiality and is valuable for mobile environments without exposing all data inside the service provider. With this option, hybrid cryptography is more efficient than using AES and other traditional methods. The proposed homomorphic ECC and Blowfish scheme has the ability to perform computation by improving security enhancement, where the information inside the mobile data remains protected even during processing compared with traditional security schemes.

The approach aims to protect data that are exchanged between server and client in SaaS. Hence, the process of encryption is thereby performed at data which is to be updated by the client beforehand this is transferred to a server. The reverse process is employed on a client before this is sent to the server. The reversing process is employed on downloading data, therefore client decrypts data downloaded from server which are from server that could be able to employed. Hence, the functioning of the system is thus mentioned below.

Data uploading. Data is encrypted from plain to cipher text before uploading by means of the blowfish approach. After that, encryption keys are thus encrypted by the ECC algorithm. Finally, both encrypted files & generated secret keys were sent to a cloud server in the form of cipher text.

Data downloading. The reverse process was carried out on downloading data. At first, an encryption key is thus decrypted using the ECC approach. Then, the generated key is employed to decrypt data using the blowfish approach. Thereby plain text is effectively recovered. In this way, unauthorized users could not employ files as it is in a secure manner, hence they are not competent to access them without using decryption. The multi-layer encryption uses blowfish as the initial layer & ECC in the next layer which is blowfish on input text after which the encryption outcome attained is delivered to the next layer at which the encrypted blowfish keys are encrypted through ECC. The final encryption output is achieved. For the understanding purpose, a few variables &v some functions list is given by $Fb, E(b,k), P(F), Ek$, & Pk which are employed in the suggested approach as given below:

Alone the encryption algorithm offered as a decryption approach is nothing but the reverse process of this is as follows:

$Step\ 1$: $Input\ text$

$Step\ 2$: $Begin$

$Step\ 3$: $For\ b = 1\ to\ N$

$Step\ 4$: $K = BLexp()$

$Step\ 5$: $E = BLenc(D, K)$

$Step\ 6$: $ECC\ (k)$

$Step\ 7$: $S(k, E)$

$Step\ 8$: End

Algorithm 3. Hybrid Homomorphic ECC & Blowfish Approach.

In this, D is the input data file, the encrypted file is E, and N signifies the number of blocks in D, $E(b,k)$ signifies the encryption function that encodes $textblocks(b)$ laterally with $kkey$ using projected $IABE - PPKGC$ system. The P(F) function might allow (Fe) encrypted files to be sent them in a cloud server. The Ek function therefore encodes blowfish key with the use of ECC system. Pk signifies the function that permits encrypted key ($K1$) which are produced by means of Ek in a cloud server.

The mathematical equation of the proposed Hybrid Cryptography is mentioned below, Elliptic curve cryptography has key generation, key exchange, and symmetric key derivation as follows.

(i) Private key-PK, Public Key-PuK, where private key select integer in the range of (1, n-1) where n is the order in base point ECP-Elliptic Curve Point, then PuK is computed in Eq. (5)

$$PuK = PK.ECP \tag{5}$$

(ii) Now key exchange is processed with a shared secret key as SK, then SK is computed in Eq. (6)

$$SK = PK_1PuK_2 = PK_1(PK_2ECP) = PK_2(PK_1ECP) = PK_2PuK_1 \tag{6}$$

(iii) Now the symmetric key derivation is derived using secret key SK and Symmetric Key as SyK using a Key creation Function as KCF expressed in Eq. (7)

$$SyK = KCF(SK) \tag{7}$$

(iv) Blowfish has divided into two parts to make smaller keys for both encryption and decryption to solve computational overhead where P is for Plaintext and C for Ciphertext. The encryption and decryption process started with block size as b and i for data that is too processed is expressed in Eq. (8)

$$C_i = Blowfish_{SyK}(P_i) \tag{8}$$

$$P_i = Blowfish_{SyK} - 1(C_i) \tag{9}$$

(v) The final output is to be processed by getting the concatenation of all decrypted blocks to recover data from the Android ransomware encrypted data as output expressed as P Plaintext in Eq. (10)

$$P = P_1||P_2||..||P_n \tag{10}$$

The main focus of this research is to enhance the deep learning model with the SSO algorithm to optimize the significant pattern of the Android malware, whether it is vulnerable or normal data, with improved classification accuracy and powerful feature extraction using saliency. AlexNet for better generalization over traditional deep learning models Finally, hybrid cryptography uses high security on Android devices by using smaller key sizes and blowfish integration, making it a fast cryptographic model to achieve integrity and minimize computational overhead in a cloud environment where all information is stored[34].

Experimental results

The analysis of performance on the proposed system is carried and the evaluated outcomes attained are compared with traditional models to compare the efficacy of the proposed design with existing approaches[26–28]. The analysis is carried out for both detection mechanisms and security constraints[29].

Confusion matrix of the proposed model

It is a table that defines the evaluated concert of the classification algorithm to detect the actual and predicted classes to define the accuracy of the model it has True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The representation of the Confusion Matrix is shown in Fig. 2.

Evaluation results

The precision is computed by Eq. (11):

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

The accuracy is computed by Eq. (12):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Recall is Computed by Eq. (13):

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

F1-score is Computed by Eq. (14):

$$F1 - score = \frac{(2 * TP)}{(2 * TP + FN + FP)} \tag{14}$$

Comparative analysis of deep learning model to detect ransomware

The comparative analysis of the proposed and existing model[26] is made and the outcomes attained are provided here. Table 2 is the comparison made for the train/test split of the suggested strategy in terms of accuracy, precision, recall, and F1-score. Also, the graphical representation of this is shown in Fig. 3.

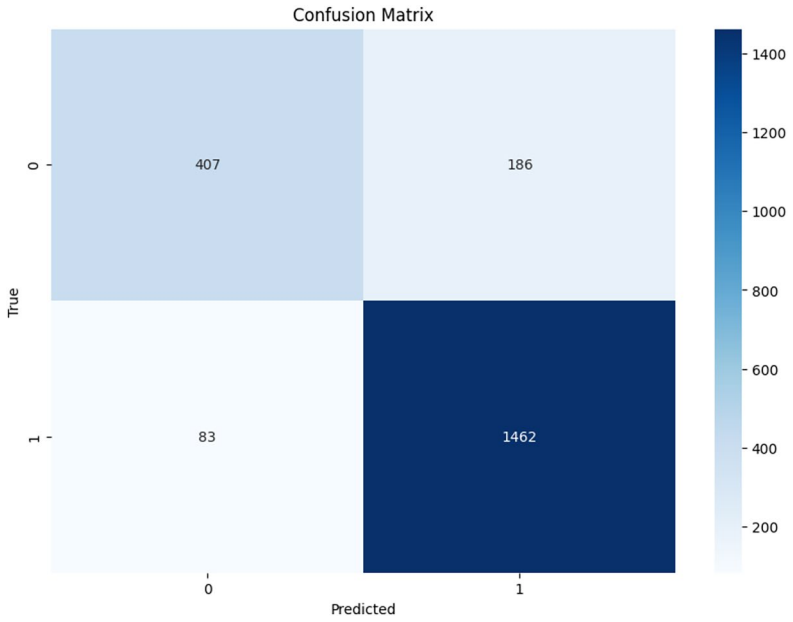

Fig. 2. Confusion Matrix of the proposed model.

Train	Accuracy	Precision	Recall	F1-score
80	99.89	98.93	94.62	96.63
70	99.96	98.21	95.78	97.89
60	99.95	98.14	95.44	97.90

Table 2. Comparison of the training model of the proposed scheme.

Table 3 denotes the comparative estimation of prediction time and training time for the training/testing split of the suggested scheme. Also, the graphical representation of this is shown in Fig. 4.

In Table 3 the analysis proves that the proposed model is effective in all metrics on comparing existing models. The graphic illustration of this is shown in Fig. 5. The proposed model comparative analysis with existing models is listed in Table 4.[36,37].

Table 5 shows the proposed model comparisons made for proposed and various existing schemes in terms of training time. The analysis proves that the proposed model is effective in comparing existing models[38,39].

Figure 6 is the proposed model comparisons made for the proposed and various existing schemes in terms of training time. The analysis proves that the proposed model is effective in comparing existing models [40].

Thus, from the analysis, it was obvious that the detection mechanism of the proposed scheme is more effective than others compared to existing models[41].

Comparative analysis of security storage from cloud data

The hybrid cryptographic model proposed is analyzed and outcomes are compared with traditional models[27,28] to validate the proposed system effectiveness.

Table 6 is the proposed model security constraint comparisons made for the proposed and various existing schemes in terms of key generation time, key pairing time, and accuracy of the cryptography approach. The analysis proves that the projected (SSO-AlexNet-based Hybrid cryptography) model is effective in comparing existing models[27,30,42].

In Table 6, a performance comparison of key generation time, key pairing time & accuracy value is made. From the outcomes, it was clearly shown that the proposed method offers lower time for key generation and key pairing with a high rate of accuracy on comparing traditional methods.

Thus, the proposed method is highly secure in comparing existing methods[31,43].

Figure 7 is the performance comparison of accuracy value is made. From the outcomes, it was clearly shown that the proposed method offers lower time for key generation and key pairing with a high rate of accuracy on comparing traditional methods. Thus, the proposed method is highly secure in comparing existing methods[44].

Table 7 shows the comparisons made for block generation analysis and computational time. The proposed shows better outcomes than other existing models[27,45].

From Table 8, it demonstrates that the time engaged is 1.5 days and the entire computational charge is assessed as 246 bytes. The computation charge is extreme and the time engaged is reduced associated with the prevailing studies[27,32].

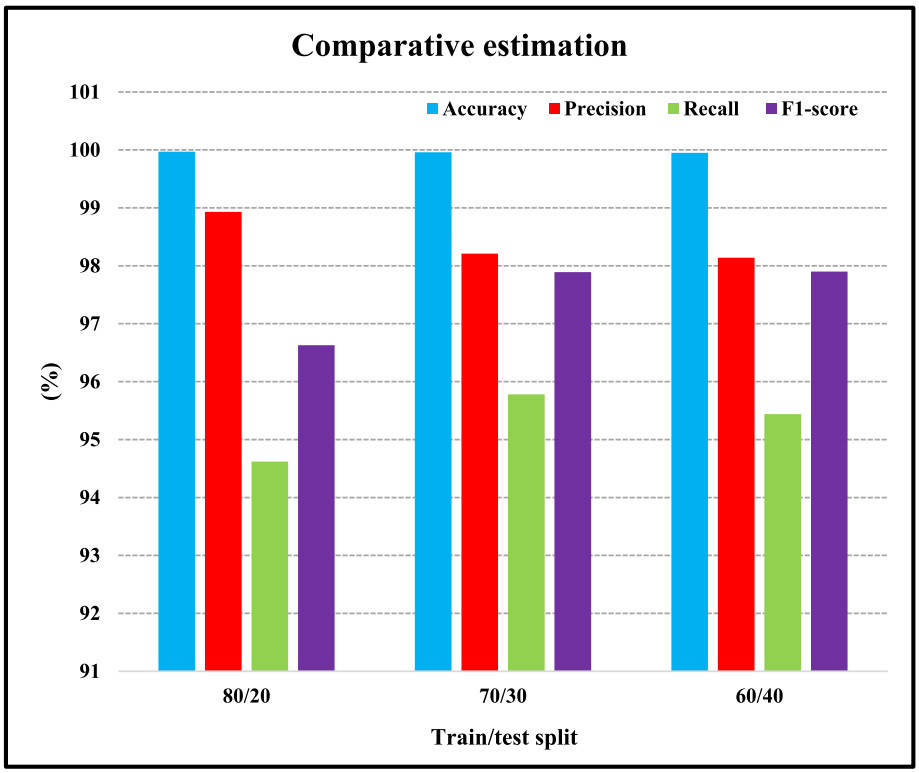

Fig. 3. Comparison of train/test split of the proposed model.

Train/test split	Prediction time	Training time
80/20	4.96	0.74
70/30	4.8	1.32
60/40	3.71	1.44

Table 3. Analysis of training time and prediction time for the proposed model.

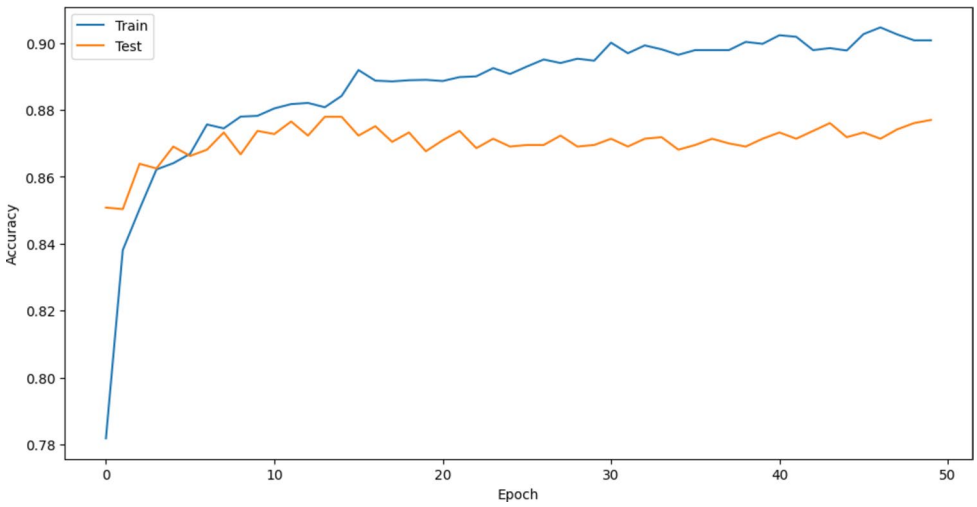

Fig. 4. Comparative analysis of training time and prediction time for the proposed model.

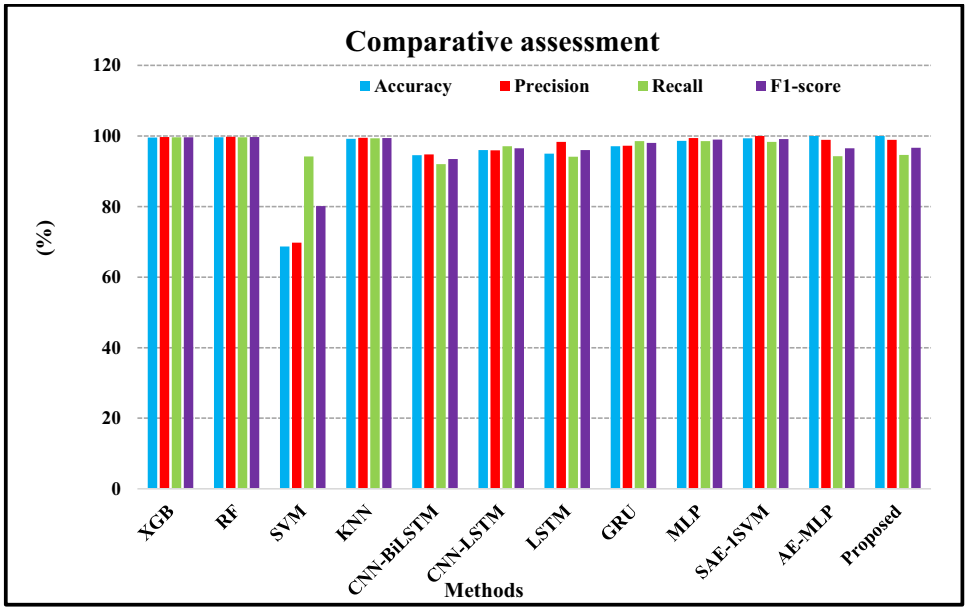

Fig. 5. Proposed model comparative analysis with existing models.

Approach	Accuracy	Precision	Recall	F1-score
XGB	99.54	99.68	99.62	99.65
RF	99.59	99.75	99.63	99.69
SVM	68.70	69.73	94.18	80.13
KNN	99.17	99.45	99.31	99.38
CNN-BiLSTM	94.52	94.74	92.04	93.44
AE-MLP	99.98	98.92	94.24	96.52
Proposed	99.97	98.93	94.62	96.63

Table 4. Proposed model comparative analysis with existing models.

Table 9 shows the comparisons made for encryption time for proposed and various traditional models in terms of time, size of secure key, security, and accuracy which reveals that the proposed method is better in offering lower encryption time with a high level of security[27,33,49,50].

Figure 8 defines the final IDE output.

Table 10 signifies the overall performance accuracy made for various existing techniques and proposed models. The overall accuracy of this proposed design is enhanced than others. Figure 9 is the representation of this in graphical form[51,52].

Henceforth, from the above evaluations of the proposed approach, the challenges observed were the computational complexity of the hybrid cryptography approach for mobile devices as a resource-constrained environment, but attaining confidentiality in the data requires process optimization to achieve efficiency without compromising security. Secondly, the challenges we face are in the representation of training data, where bias happens during the generalization of data. This challenge is overcome with SSO optimization with a data analysis function to monitor imbalanced data, and enhancement of detection accuracy is improved with the AlexNet DL model. From the analysis, it was obvious that the proposed model is better and offers an enhanced outcome than other existing models.

Approach	Training time (sec)
XGB	1.67
RF	7.42
SVM	8.91
KNN	7.97
AE-MLP	4.96
Proposed	4.05

Table 5. Proposed model comparative analysis with existing models.

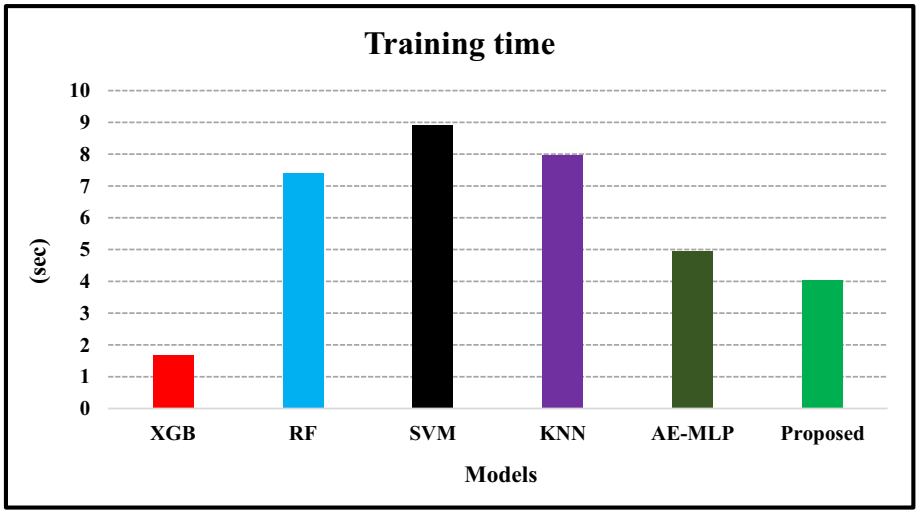

Fig. 6. Proposed model comparative analysis with existing models.

Methods	Time of key generation (sec)	Time of key pairing (ms)	Accuracy (%)
DSA	0.9817531	0.8761831	76
EC	0.77874323	0.83443677	89.23
Diffie Hellman	0.67874323	0.79459034	92.23
RSA	0.97874323	0.87874323	89.13
Proposed	0.49781462	0.60435719	99.83

Table 6. Comparison of proposed with existing methods.

Conclusion and future works

In this study, the proposed deep learning model to detect the data that is targeted by the Android malware, which processes the active structures such as permissions, opcodes, APIs, and system calls for the detection operation. To extract all the information related to the Android ransomware, we are using a DL-based model called adaptive deep security. An AlexNet classifier was employed to detect and classify data as malicious or not. Subsequently, for secured storage of data in the mobile cloud, the hybrid homomorphic ECC and Blowfish cryptographic scheme was employed, which includes key computation and key generation processes. The cryptographic scheme includes encryption and decryption of data, after which the app response was found to attain a decrypted result upon user request, where opcodes are analyzed with low computational overhead with the SSO optimization algorithm. The result of this experiment demonstrates that the detection accuracy is in the range of 99.89%, which outperforms the normal traditional Android malware detection model.

In the future, the same model is to be tested further with more new datasets related to Android malware and opcodes that are analyzed with different sequences to analyze the family of ransomware, and the type of

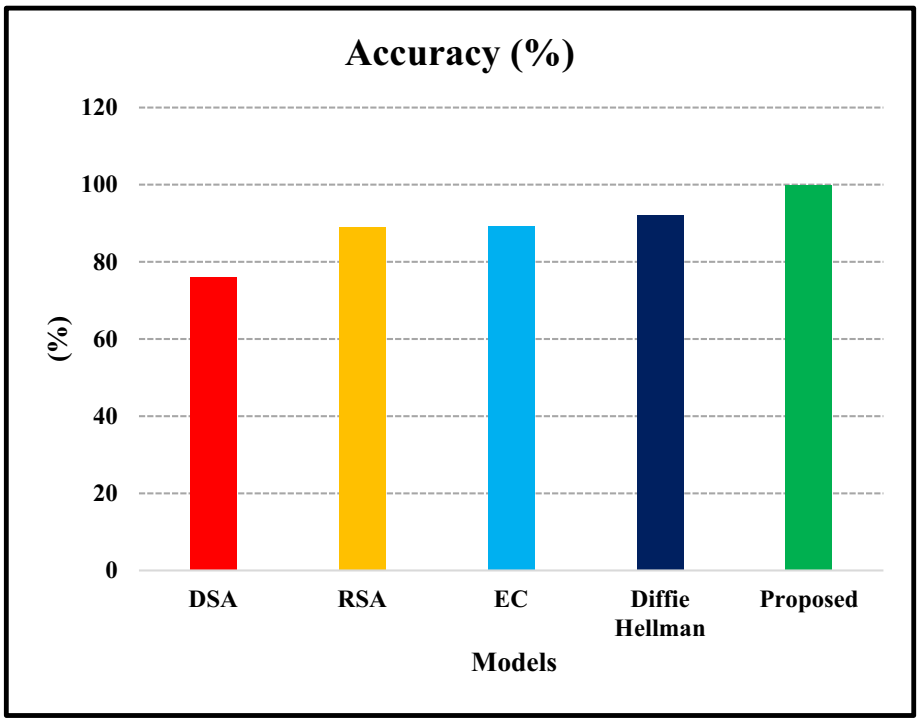

Fig. 7. Comparative analysis of accuracy security mechanism.

Methods	Number of blocks	Computation time (sec)
Wang et al. (2011)	750	0.92344
Liu et al. (2013)	900	0.85234
Guo et al. (2019)	1400	0.66654
Proposed	1764	0.42761

Table 7. Comparative estimation of Computational time and block generation analysis.

Methods	Cost of computation (bytes)	Time (in days)
Wang et al. (2011)[46]	128	3
Liu et al. (2013)[47]	128	2.8
Guo et al. (2019)[48]	168	2.5
Proposed	246	1.5

Table 8. Time delay comparison.

Methods	Time (Ms)	Secure key size (bytes)	Security	Accuracy
AES-128 CBC	10.3435	128	7.1	5.02
DES	27.023423	192	7.8	5.3
DES-CBC	19.12312	256	8.1	7.1
AES-256 CBC	13.124	256	7.6	6.3
DES-ECB	16.312	256	8	6.8
Proposed	4.27610	256	9.4	9.8

Table 9. Comparative estimation of Encryption time.

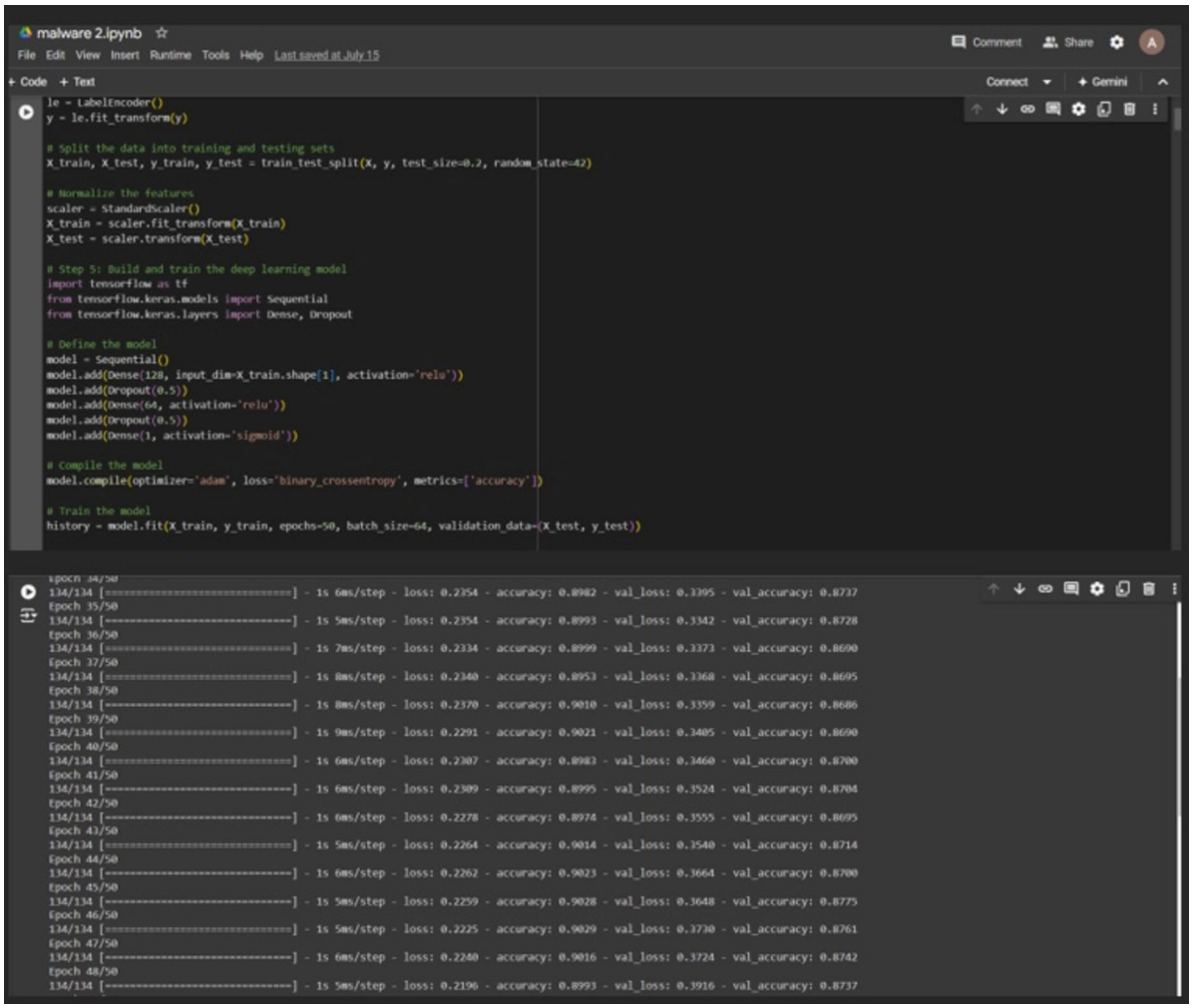

Fig. 8. IDE Output of the proposed model.

References	Entity check	Techniques	Overall accuracy (%)
Wu[40]	All permissions	GNN, Bi-LSTM	94.76
Hossain[41]	Network Traffic	PSO	81.58
Yadav et al.[42]	Image File	CNN	95.7
Ahmed[43]	All permissions	TF-IDF	98.0
Wu[44]	All permissions	MLP	93.90
Lachtar[45]	Opcode	CNN	96.0
Gao et al.[49]	API	GCN	98.99
Sharma[50]	All permissions	VT & PCA	98.08
Surendran et al.[51]	System Calls	RF, NB	96.0
Zhang et al.[52]	All permissions	TF-IDF	91.43
Proposed	All permissions	Deep learning model	99.89

Table 10. Overall performance accuracy.

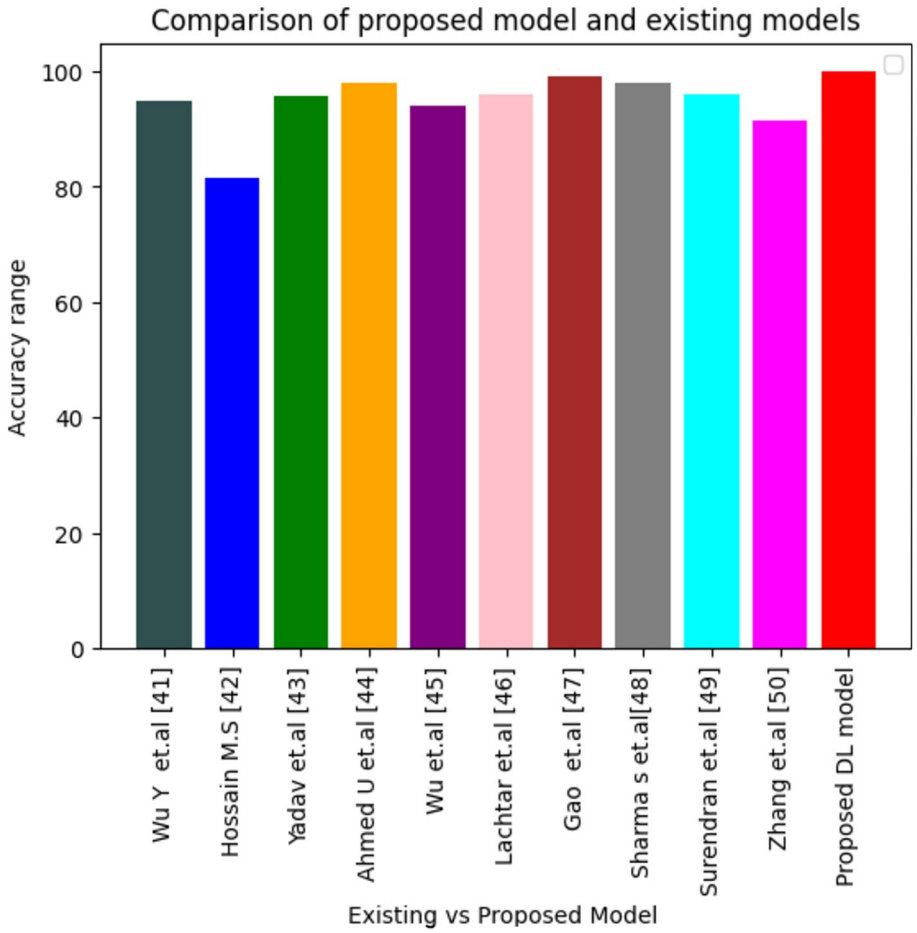

Fig. 9. Overall performance accuracy.

cryptography algorithm is to be analyzed in the near future by adding an additional tuning layer to the proposed model and getting superior results by comparing it with the hybrid cryptography algorithm fused with the deep learning model. This work might be extended by adding re-encryption and secured transmission of data with quality-dependent coding in different domains such as healthcare and finance, which are implications for future research in mobile environments.

Data availability

The datasets used and analyzed during the current study are available from the corresponding author on request.

Received: 10 March 2024; Accepted: 19 August 2024
Published online: 27 September 2024

References

1. Liu, K. *et al.* A review of android malware detection approaches based on machine learning. *IEEE Access* **8**, 124579–124607 (2020).
2. Jyothi, K. K. *et al.* A novel optimized neural network model for cyber attack detection using enhanced whale optimization algorithm. *Sci. Rep.* **14**(1), 5590 (2024).
3. Almomani, I. *et al.* Android ransomware detection based on a hybrid evolutionary approach in the context of highly imbalanced data. *IEEE Access* **9**, 57674–57691 (2021).
4. Ogwara, N. O., Krassie, P., & Yang, M. L. B. MOBDroid: An intelligent malware detection system for improved data security in mobile cloud computing environments. In *2020 30th International Telecommunication Networks and Applications Conference (ITNAC)* (IEEE, 2020).
5. Ezhilarasi, T. P. *et al.* A secure data sharing using IDSS CP-ABE in cloud storage. In *Advances in Industrial Automation and Smart Manufacturing: Select Proceedings of ICAIASM 2019* (Springer, 2021).
6. Shabbir, M. *et al.* Enhancing security of health information using modular encryption standard in mobile cloud computing. *IEEE Access* **9**, 8820–8834 (2021).
7. Qi, S. *et al.* Secure data deduplication with dynamic access control for mobile cloud storage. *IEEE Trans. Mob. Comput.* **23**(4), 2566–2582 (2023).
8. Wang, Y. *et al.* Efficient and secure content-based image retrieval with deep neural networks in the mobile cloud computing. *Comput. Secur.* **128**, 103163 (2023).
9. Benil, T. & Jasper, J. J. C. N. Cloud based security on outsourcing using blockchain in E-health systems. *Comput. Netw.* **178**, 107344 (2020).

10. Velmurugadass, P. *et al.* Enhancing Blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm. *Mater. Today Proc.* **37**, 2653–2659 (2021).
11. Thirumalai, C., Mohan, S. & Srivastava, G. An efficient public key secure scheme for cloud and IoT security. *Comput. Commun.* **150**, 634–643 (2020).
12. Masud, M. *et al.* A robust and lightweight secure access scheme for cloud based E-healthcare services. *Peer-to-peer Netw. Appl.* **14**(5), 3043–3057 (2021).
13. Shen, J. *et al.* A privacy-preserving and untraceable group data sharing scheme in cloud computing. *IEEE Trans. Dependable Secur. Comput.* **19**(4), 2198–2210 (2021).
14. Hedaia, O. A. *et al.* Bio-CAPTCHA voice-based authentication technique for better security and usability in cloud computing. *Int. J. Serv. Sci. Manag. Eng. Technol. (IJSSMET)* **11**(2), 59–79 (2020).
15. Kavin, B. P. *et al.* A modified digital signature algorithm to improve the biomedical image integrity in cloud environment. In *Advances in Computational Techniques for Biomedical Image Analysis* 253–271 (Academic Press, 2020).
16. Movassagh, A. A. *et al.* Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model. *J. Ambient Intell. Humaniz. Comput.* **14**, 1–9 (2023).
17. Orantes, J., Sandra, D. & Eleazar, A. A. A survey on information security in cloud computing. *Comput. y Sist.* **24**(2), 819–833 (2020).
18. Ogiela, U. Cognitive cryptography for data security in cloud computing. *Concurr. Comput. Pract. Exp.* **32**(18), e5557 (2020).
19. Jabbar, A. A. & Bhaya, W. S. Security of private cloud using machine learning and cryptography. *Bull. Electr. Eng. Inform.* **12**(1), 561–569 (2023).
20. Mohd, A. A. *et al.* Design of mutual authentication method for deep learning based hybrid cryptography to secure data in cloud computing. *Int. J. Saf. Secur. Eng.* **13**(5), 893 (2023).
21. Attou, H. *et al.* Cloud-based intrusion detection approach using machine learning techniques. *Big Data Min. Anal.* **6**(3), 311–320 (2023).
22. Ahmad, F. B. *et al.* Securing cloud data: A machine learning based data categorization approach for cloud computing. (2022).
23. Alzubi, O. A. *et al.* Optimized machine learning-based intrusion detection system for fog and edge computing environment. *Electronics* **11**(19), 3007 (2022).
24. Singh, A. *et al.* Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data. *Electronics* **12**(18), 3899 (2023).
25. Alzubi, O. A. *et al.* Quantum Mayfly optimization with encoder-decoder driven LSTM networks for malware detection and classification model. *Mob. Netw. Appl.* **28**(2), 795–807 (2023).
26. Adeniyi, O. *et al.* Securing mobile edge computing using hybrid deep learning method. *Computers* **13**(1), 25 (2024).
27. Hahn, C. *et al.* Enabling fast public auditing and data dynamics in cloud services. *IEEE Trans. Serv. Comput.* **15**(4), 2047–2059 (2020).
28. Shah, P. & Prajapati, P. Provable data possession using additive homomorphic encryption. *J. King Saud Univ. Comput. Inf. Sci.* **34**(6), 3448–3453 (2022).
29. Singh, A. *et al.* Transfer fuzzy learning enabled streebog cryptographic substitution permutation based zero trust security in IIOT. *Alex. Eng. J.* **81**, 449–459 (2023).
30. Anitha, T. *et al.* A novel methodology for malicious traffic detection in smart devices using BI-LSTM–CNN-dependent deep learning methodology. *Neural Comput. Appl.* **35**(27), 20319–20338 (2023).
31. Dhanaraj, R. K. *et al.* Black hole and sink hole attack detection in wireless body area networks. *Comput. Mater. Contin.* **68**(2), 1949–1965 (2021).
32. Aanjankumar, S. & Poonkuntran, S. Peer-2-Peer Botnet manage SDT security algorithm. In *2016 IEEE international conference on computational intelligence and computing research (ICCIC)*. (IEEE, 2016).
33. Alzubi, O. A. *et al.* An efficient malware detection approach with feature weighting based on Harris Hawks optimization. *Clust. Comput.* **25**, 1–19 (2022).
34. Dataset Available online- https://github.com/harrypro02/Android-Malware-Permission-Based-Dataset.
35. Alzubi, J. A. *et al.* A blended deep learning intrusion detection framework for consumable edge-centric iomt industry. *IEEE Trans. Consum. Electron.* **70**, 2049 (2024).
36. Alzubi, O. A. *et al.* An optimal pruning algorithm of classifier ensembles: dynamic programming approach. *Neural Comput. Appl.* **32**, 16091–16107 (2020).
37. Dataset Available online- https://www.unb.ca/datasets/maldroid-2020.html.
38. Available online : https://github.com/Mahesh68i90/EGG-test-AANJAN.
39. Alzubi, O. A. *et al.* Cryptosystem design based on Hermitian curves for IoT security. *J. Supercomput.* **76**(11), 8566–8589 (2020).
40. Wu, Y. *et al.* DeepCatra: Learning flow-and graph-based behaviours for Android malware detection. *IET Inf. Secur.* **17**(1), 118–130 (2023).
41. Hossain, M. S. *et al.* Android ransomware detection from traffic analysis using metaheuristic feature selection. *IEEE Access* **10**, 128754–128763 (2022).
42. Yadav, P. *et al.* EfficientNet convolutional neural networks-based Android malware detection. *Comput. Secur.* **115**, 102622 (2022).
43. Ahmed, U., Lin, J.-W. & Srivastava, G. Mitigating adversarial evasion attacks of ransomware using ensemble learning. *Comput. Electr. Eng.* **100**, 107903 (2022).
44. Wu, B. *et al.* Why an android app is classified as malware: Toward malware classification interpretation. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **30**(2), 1–29 (2021).
45. Lachtar, N. *et al.* Ransomshield: A visualization approach to defending mobile systems against ransomware. *ACM Trans. Priv. Secur.* **26**(3), 1–30 (2023).
46. Wang, X., Wang, X., Zhao, J. & Zhang, Z. Chaotic encryption algorithm based on alternant of streamcipher and block cipher. *Nonlinear Dynamics* **63**, 587–597 (2011).
47. Liu, T. *et al.* A dynamic secret-based encryptionscheme for smart grid wireless communication. *IEEE Transactions on Smart Grid*, **5**(3), 1175–1182 (2013).
48. Guo, S., Hu, X., Guo, S., Qiu, X., & Qi, F. Blockchain meets edge computing: A distributed and trustedauthentication system. *IEEE Transactions on Industrial Informatics*, **16**(3), 1972–1983 (2019).
49. Gao, H., Cheng, S. & Zhang, W. GDroid: Android malware detection and classification with graph convolutional network. *Comput. Secur.* **106**, 102264 (2021).
50. Sharma, S., Krishna, C. R. & Kumar, R. RansomDroid: Forensic analysis and detection of Android Ransomware using unsupervised machine learning technique. *Forensic Sci. Int. Digit. Investig.* **37**, 301168 (2021).
51. Surendran, R., Thomas, T. & Emmanuel, S. Gsdroid: Graph signal based compact feature representation for android malware detection. *Expert Syst. Appl.* **159**, 113581 (2020).
52. Zhang, H. *et al.* Classification of ransomware families with machine learning based onN-gram of opcodes. *Futur. Gener. Comput. Syst.* **90**, 211–221 (2019).

Author contributions
All the authors have contributed to this article equally.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to R.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.