ISSN: 0974-5823 Vol. 7 No. 5 May, 2022

International Journal of Mechanical Engineering

INTENSIFIED DEEP NEURAL NETWORK WITH CAT SWARM BEHAVIOUR OPTIMIZATION BASED ANDROID MALWARE ATTACK DETECTION

[1] Balamurugan.R, [2]Ravichandran.M

Department of Computer Science, Sri Ramakrishna Mission vidyalaya College of Arts and Science, Coimbatore, Tamilnadu.

Abstract

As the usages of mobile devices expands, malware attacks particularly on Android phones, is also rapidly increased. Hackers use a varieties of tactics to attack on cell phones including identity thefts, eavesdropping, and fraudulent marketing. Due to its open source design, the Android Operating System has been widely adopted by varieties of developers. Despite of the fact that mobile phone development has resulted in significant technological advancements that they have also become powerful platforms for cyberattacks and cyberwarfare against business infrastructures and individual users of these smart phones. Different types of cyberattacks occur but malicious application attacks in smart phones have gained the lead. There are numerous literatures are in existence which proves machine learning based approaches to be an efficient resource for detecting malware attacks among variety of defense mechanisms. But, the problem of class imbalance and the overfitting is not a primary focus of the existing works in android malware attack detection, as they emphasis on the existing patterns of malware behaviour.To overcome this issue, this paper concentrated on developing an Intensified Deep Neural network (IDNN), its objective is to detect the android malware attacks by understanding the depth pattern of both benign and malicious applications. This proposed model IDNN, adapted the intelligence of the cat swarm behavior to optimize classification task during training phase. The seek and trace mode of cat swarm algorithm searches for the best suited weight values and assigned to the hidden nodes of the dense layer to improve the accuracy of the android malware attack detection instead of using gradient descent-based weight assignment. The simulation result proved that IDNN, accomplishes high accuracy rate in the android malware attack detection while comparing to the other standard models

Keywords: Android, malware attacks, Deep Neural Network, Cat Swarm optimization, class imbalance, gradient descent

Introduction

The advancement in digital India project has increase the usages of mobile.The most popular Smartphone platforms which occupied 90% of Android mobile operating system used worldwide. The feasibility of Android, allows third parties to install their applications without any authorized control and it impacts by its open source code. Thus, it becomes target for malware activity even in presence of security approaches [1]. The vulnerability over Android becomes an important area of research work whose objective is to provide a strong security solution by developing statistical, dynamic, machine learning and mining approaches. The analytical technique investigates the data to discover insights which assist in malware activities prediction.

There are two kinds of analysis, namely the approach which collects the information about the software for analyzing without executing it is known as static analysis [2]. Another approach which examines cyber thread based on its behaviour during its progression is referred as dynamic analysis.

During the earliest stages, crime on internet has done through the malware attacks, the attackers install the malware to delete or install programs, sensitive information hacking, modifying content of various files [3]. The systems which comes under the control of hackers may impersonate as owner of the system and capture user's camera, retrieve videos, photos and monitors the action of users to infect their device. Thus, the mobile devices become the target for the malware developers, as it gives them more incentives, as the Android operating system has large market share it becomes the popular target for them. But it is very hard with the present malware attacks on mobile platforms. The mobile malware detection approaches are moderately immature and still researches are in process in handling the malware prediction [4]. The traditional malware detection methods are either signature or behavior based. The static malware analysis has used to fight with these problems but, they are often relied with human interactions which reduces the speed and investigation scalability. To automate the process of static analysis, source code transformation is essential to covert it as calculus for system communication with the statements and to verify the software behavior for averting ransomware attacks. The machine learning models are greatly used for automating the static malware investigation. With their intelligent approach, it learns the attack patterns and accordingly it updates their ability to detect the malware. Although Dynamic Android malware examination has addressed in many works, only very few works have done on static malware analysis-based machine learning.

Thus, this paper focuses on the Android static malware prediction, by acquiring depth knowledge of malware patterns. This research work proposed an intensified deep learning model with metaheuristic model of cat swarm optimization to fine tune its performance to achieve the highest accuracy in android malware prediction.

Related Work

Garg and Baliyan[5] in their study performed a thorough and so well taxonomy for assessing the state-of-the-art methods to Android security. The authors recognized essential features in terms of aims, analysis tools, code formats, tools and frameworks employed. They also analyzed about trends and patterns of various analysis approaches.

Chenet al., [6] designed a revolutionary compact static detection model knows as TinyDroid that used instruction reduction and computational methods to identify Android malware. A symbol-based simplification methodology is being used first to abstract the opcode series decompiled from Android Java virtual machine Executable files. Following that, N-gram is utilized to extract information from the simplified bytecode sequencing, and a learner is created to detect and describe malware. To improve the effectiveness and adaptability of the parameter estimation system, a compression method is employed to minimize characteristics and choose exemplars for the malware sample collection.

Pan et al.,[7] presented a comprehensive systematic overview, focusing on static evaluation techniques that can be employed to detect Android malware. Character trait, opcode-based, software graph-based, and symbolic performance methods were identified.Then, using the existing literature, it assessed the abilities of static analysis-based Botnet detection methods on such four approaches. The work implies that machine learning and mathematical analysis can be used to identify Android spyware.Yet, it is solely on static analysis approaches, machine learning methods have not been properly examined.

As reported in recent research of Abikoyeet al.,[8] developed a complete evaluation of machine learning paradigm for malware attack detection. First, the malware behavior is discovered using machine learning paradigm. The inferred knowledge is used to isolate or discover any such identical behaviour from unidentified attacks. Rashidi et al., [9] provided an Android malware detection system based on Support Vector Machine and Active Learning technologies in this study. This model extracts the actions of programs while they are running and map them into a feature set, after that timestamps are assigned to a specific feature in the set. The highlighted the importance of time-dependent behaviour tracking can enhance malware detection accuracy dramatically. This active learning model that uses an expected error reduction query approach to assimilate new revealing instances of Android malware and requalify the model to take on adaptive online learning.

Milosevic et al., [10] developed a two-machine learning-assisted static analysis approaches for mobile applications,first is based on permissions and the other on source code analysis using a bag of words illustration approach.

Shivi Garg and NiyatiBaliyan [11] designed both individual and parallel ensemble classifiers in their work. Separating destructive Android Package Kits (APKs) from the combined dataset is the first step towards categorization. Only harmful APKs are required for further processing and implementation in order to identify and classify malware families. The dataset acquired through static feature extraction and private identification is split into two parts in the model training and evaluation phase; one part is used to train the model, while the other is used to test the model.

Sabhadiya et al.,[12] devised a deep learning strategy for attacking devices and antivirus products to protect android system from malware. They construct this model to detect different types of android malwares by construction the deep learning model to detect whether or not an Android app is compromised with malware without having to install it.

Karbab et al., [13] in their study, presented MalDozer, an autonomous Android malware detection and family attribution platform based on deep learning sequence classification. MalDozer intelligently extracts and learns the dangerous and benign patterns from the actual samples to identify Android malware, starting with the raw sequence of the app's API method calls.

Methodology: Intensified Deep Neural Network by using the Cat Swarm Behavior Optimization for Android Malware Detection

The main objective of this research work is to detect android application as benign or malicious by developing a Cat Swarm Behaviour optimized Deep Neural Network. The main issue focused in this research work is to overcome the problem of class imbalance in the dataset and avoid overfitting while using deep neural network. The overall framework of the proposed model for android malware detection is depicted in the figure 1.

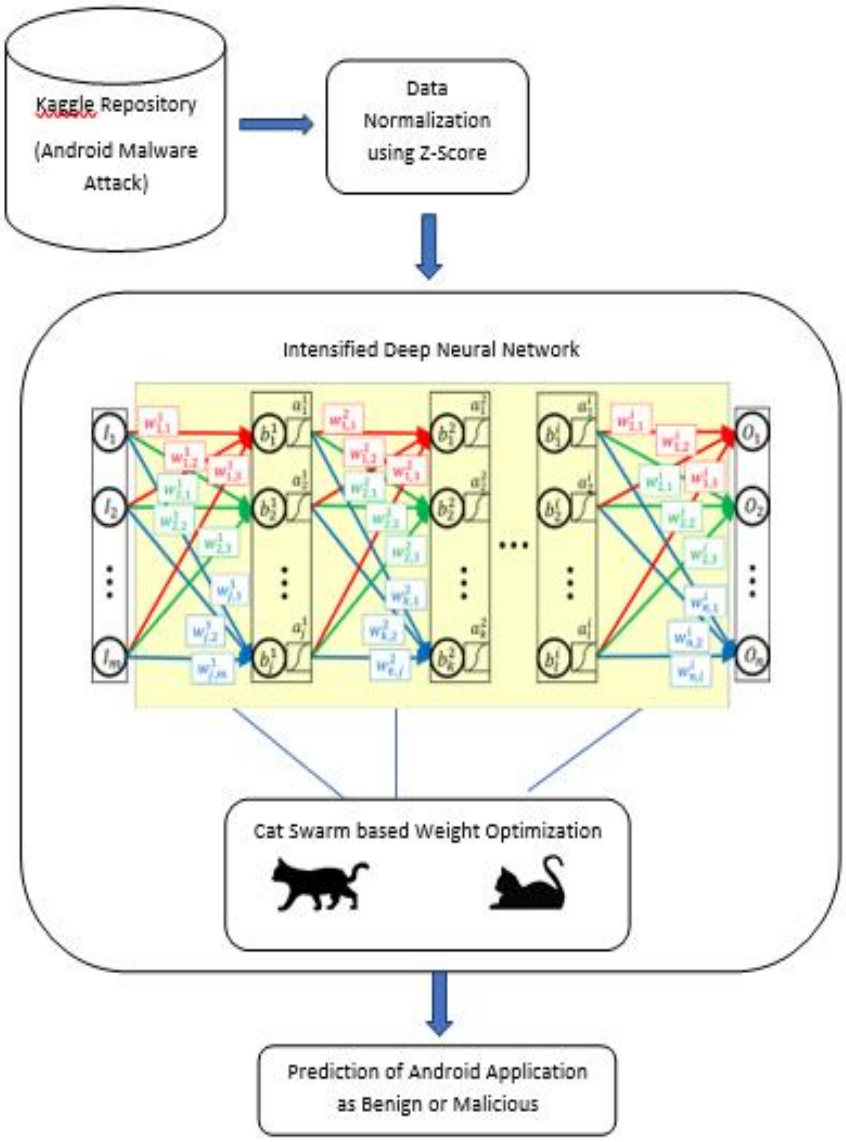

Figure 1 Overall Framework of the proposed Intensified Deep Neural Network using Cat swarm Optimization for Android Malware attack Detection

In this proposed work the dataset is collected from the Kaggle repository to detect the Android malware detection through smart phones which pose a significant threat to both individuals and business applications since they now hold all of our personal and financial informations it may consequence disclosure or destruction of privacy information. The android malware analysis dataset [14] comprised of the APK details installed in the android to detect its genuinity. The number of records in this dataset is 7845 with 17 attributes and a class variable. Initially, the value of raw dataset has different range of values, to fall them under a same range of values, this work used Z-Score normalization. The normalized dataset is fed as input to the deep neural network. Once the data has received by the DNN, it is passed to the dense hidden layers, the parameters weights and bias is used for activating the concern hidden nodes. The output layer performs the classification of concern input apk as benign or malicious. During the pre-training phase, DNN parameter weight values are optimized by applying the metaheuristic model known as cat swarm behavior optimization. Which uses its seeking and tracing mode to select the best weight values instead of random selection in standard DNN. Thus, it detects the Genuity of the android apk as benign or malicious.

The detailed description of each process involved in Android malware attack detection is in the following sections.

Dataset Normalization using Z-Score

To normalize the android malware attack dataset, this work used the Z-Score computation which is formulated in the below equation.

$$ZS = \frac{x_{i=1,,m,j=1..n} - \mu_{i=1..m,j=1...n}}{\sigma_{i=1...m,j=1...n}}$$

Where x is the attribute value of i^{th}record and j^{th} attribute, μ is the mean value and σ refers to population standard deviation.

Deep Neural Network

The deep neural network is a class of machine learning algorithm developed based on the convolutional neural network []. It has dense structure and it is constructed with greedy layer by layer method. DNN belongs to feedforward network type, its data passes from the input to the output layer deprived of looping back. It has the option of inbuilt feature extraction which is not presented in the ANN and MLP. The general structure of the DNN is shown in the figure 2.

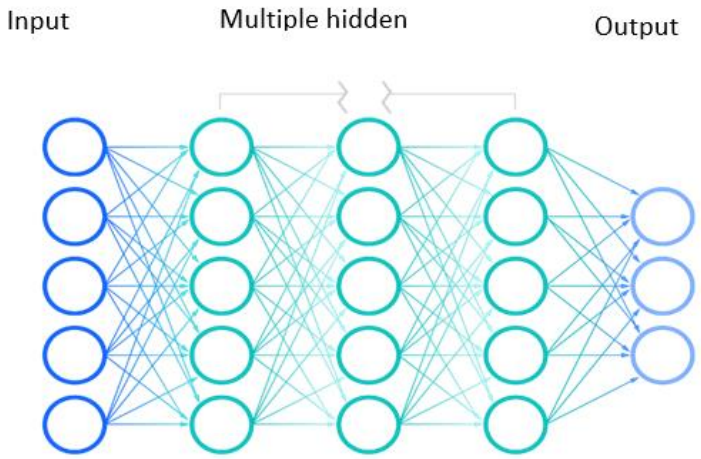

Figure 2: Simplified View of Deep Neural Network

The depth of a network signifies the non-linear transformations among separating layers whereas width of the hidden layer is the dimensionality of it. In a neural network, neurons are connected or linked to each other, and each connection in the neural network is associated with a weight, which has multiplied by input value to determine the importance of the relationship between the neurons as shown in the figure 3.

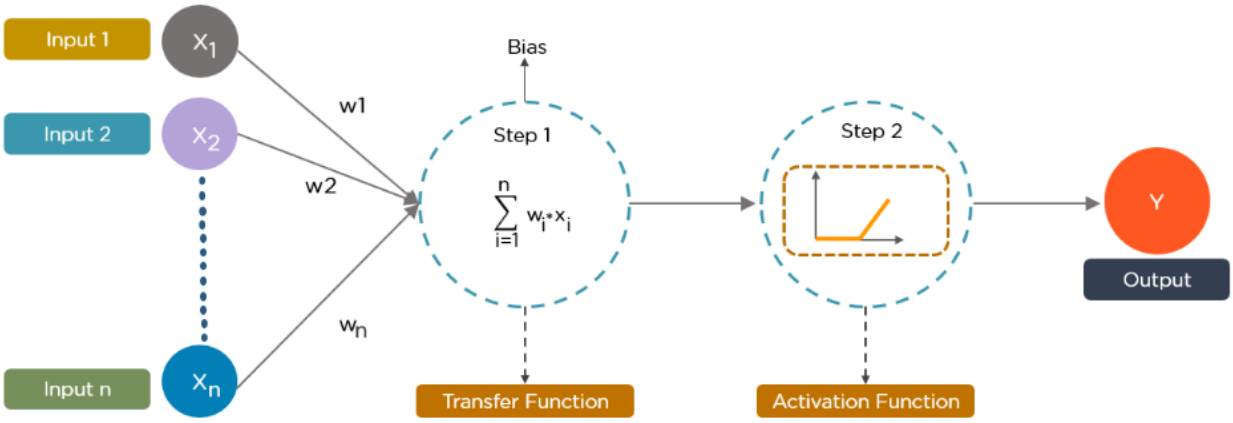

Figure 3 activation function process for single hidden layer in DNN

The output of neuron is defined by using activation function which has the modeling capability of non-linearity in the network. During the training phase neural network understands and learns about the parameter values namely weight and bias. These parameters are the genuine part of deep learning model which learns the parameters by applying forward propagation and back propagation in trail and error basis.

The input of the android application details is passed as the input during the training phase. The forward propagation occurs in the first phase in which all the nodes in the network apply their transformation function to the input data they receive from the previous layers neurons and pass its output to the preceding layer's neuron. After the input data is processed by all the neurons, the final layer is known as the output layer, it generates the label prediction as malicious or benign application.

The predicted output is compared with the expected output by applying the loss function to calculate the error rate. The difference between the predicted and expected value wants to be zero. If there is a difference, the model will be trained by adjusting the weights of the links of neurons in a repeated manner until it produces best prediction result.

The computed loss information is propagated by backward using backpropagation which starts from the output layer to all the nodes in the hidden layer which subsidize directly to the output. But, based on the relativeinfluence ofeach neuron to the original output, the neurons of the hidden layer only get a percentage of the total signal of the loss. Layer after layer, this procedure is repeated until all of the neurons in the network have received a loss signal expressing their relative contribution to the total loss.To adjust the weights, gradient descent method is used for achieving the global minimum during each iteration. It is repeated for all

the android malicious dataset that is passed to the network. Initially the parameters weights and biases are assigned with random values.

This proposed work improves the learning capability of the deep neural network by inducing the knowledge of the metaheuristic algorithm known as Cat swarm Optimization.

Cat Swarm Optimization (CSO) Algorithm

Based on the inspiration of the cat's behaviour, the cat swarm optimization algorithm has developed to discover the optimal solutions for many applications. In this proposed work the cat behaviouris has applied to adjust the weight values of the hidden nodes in Deep Neural Network. Due to the standard DNN use back propagation to update or change the weights. The strategy used for adjusting the weights are gradient descent. Based on the obtained error rate the values of weights changed by gradient descent. In current scenario metaheuristic models are used for improving the training process by substituting the Gradient Descent Strategy (GDS).In conventional DNN the gradient descent is used for modifying the weights during the training process. But in the proposed Intelligent Deep Neural Network, the cat swarm optimization is used for assigning the weight values.

$$Wt_{i+1} = Wt_i + GDS \Rightarrow Wt_i + CSO$$

While adapting cat swarm optimization it has the ability to achieve global optimum in searching of best values for assigning the parameters. The fitness value of each cat swarm is the value of the error function computed at the current location of the cat and the position vector of the cat relates to the weight matrix of the network.

Cat Swarm Algorithm imitates the seeking mode and tracing mode, which is the significant behaviour of the cats. During the seeking mode, the cats will be in rest but they are alert, while in tracing mode they search for local optima to solve the specific problem depicted in the figure 4.

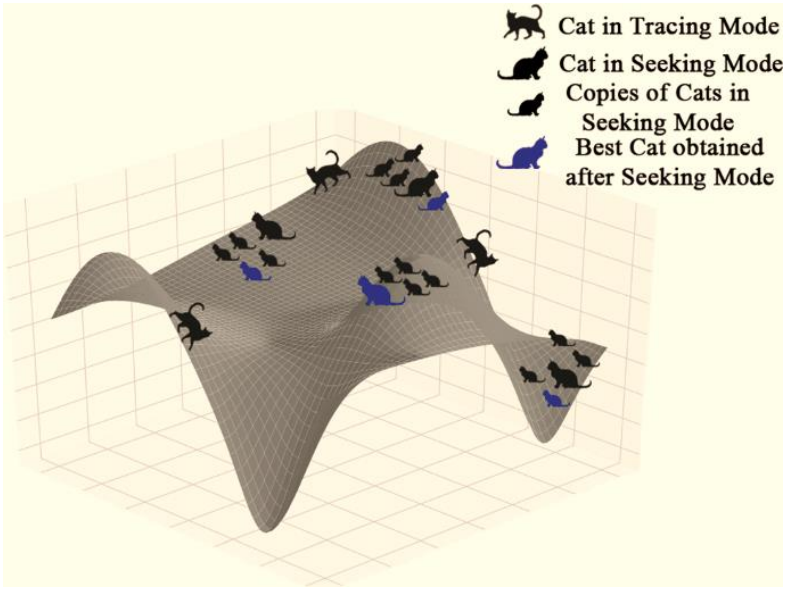

Figure 4 Cat Swarm Optimization

In this proposed work it searches for the best values to be assigned for the weight parameters of the deep neural networks to improve the accuracy rate, reduce the error rate and speedy training phase.

Seeking Mode of Cat

In this mode, it comprised of four essential factors they are seeking range of the chosen dimension (δ), seeking memory pool size (α), number of dimensions to change (D), minimal and maximum range of seeking, considering self-position (S), A mixture ratio (R) which is a faction of the cat population that has a small value to ensure that cats usually spend most of their time in resting and observing.

Algorithm

Seeking mode:

1. Select random fraction of population as seeking cats based on the mixture Ratio R
2. Make copies of cat based on (α)
3. Update the position of each copy using δ, let consider the current position also and either by add or subtract with (α) percentage to obtain the present position and replace the old ones.
4. Compute the fitness value of each copies
5. If all the fitness values are not exactly equal, then select the best ones by applying probability of each candidate using the equation

$$P_i = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}}, where\ 0 < i < j$$

6. Else set the selecting probability of each candidate to 1
7. Arbitrarily select the place to move from candidate place and replace the cat m[th]position with it
8. Repeat step 3until all the seeking cat gets involved

Tracing mode of Cat

In the optimization process, the tracing mode is used as an exploratory strategy. The cat can track the desired object with considerable energy during this period. The cat's fast chase can be mathematically represented by shifting its position. As a result, define i[th] cat's position and velocity in D-dimensional area.When a cat enters tracing mode, it moves in each dimension through its own velocities. The velocity of the cat is updated for each dimension

$$\boldsymbol{v_{m,d} = v_{m,d} + r_1 * e_1 * (y_{bst,d} - y_{m,d}) d = 1, 2, \ldots N}$$

The catd position is updated as

$$\boldsymbol{y_{m,d} = y_{m,d} + v_{m,d}}$$

Dataset Description

The android malware analysis dataset is collected from the Kaggle repository. It comprised of the APK details installed in the android to detect its genuinity. The number of records in this dataset is 7845 with 17 attributes and a class variable. The attributes are Name of the APK, tcp_packets, dist_port_tcp, external_ips, volume_bytes, udp_packets, tcp_urg_packet, source_app_packets, remote_app_packets,source_app_bytes, remote_app_bytes, duration, avg_local_pkt_rate, avg_remote_pkt_rate, source_app_packets, dns_query_times and type of attack.

Results and Discussions

In this section the performance of the proposed model Intensified Deep Neural Network based (IDNN) android malware attack detection is discussed. The newly developed IDNN is deployed using python software and the dataset is collected from Kaggle repository. The dataset has comprised of 7845 records with 17 features which are about the applications in android software and the type of the attack. The IDNN performance is compared with other four existing classification models Decision Tree (DT), Random Forest (RF) and Support Vector Machine(SVM). The evaluation of each model has done by using accuracy, precision and recall which are mathematically formulated as shown in the equations.

$$\text{Precision} = \frac{\text{Total No of correctly classifed malware attacks}}{\text{TotalNumberofappsdetectedasmalware}}$$

$$\text{Recall} = \frac{\text{Total No of correctly classifed malware attacks}}{\text{TotalNumberofmalwareapps}}$$

$$\text{Accuracy} = \frac{\text{Total No of correctly classified normal and malware apps}}{\text{TotalNumberofapps}}$$

Table : Performance comparison of Classification models to detect Android Malware Attack

Prediction Models	Accuracy	Precision	Recall
DT	77	78.6	79.6
RF	81.5	82.4	83.9
SVM	90	89.2	89.4
IDNN	98.1	96.8	98.3

The table 1 displays the experimental results of four different classification models based on their precision, recall an accuracy measures in android malware attack detection. It is observed from the result that the performance of the newly constructed Intensified Deep Neural Network which is the combination of metaheuristic model known as cat swarm optimization and Deep Neural Network understands the depth pattern of the android dataset to predict the malicious applications during the pre-training phase of the model. The other existing models DT, RF and SVM suffers by the class imbalance and overfitting problem in detection of malware attacks and produced less result compared to the proposed IDNN.

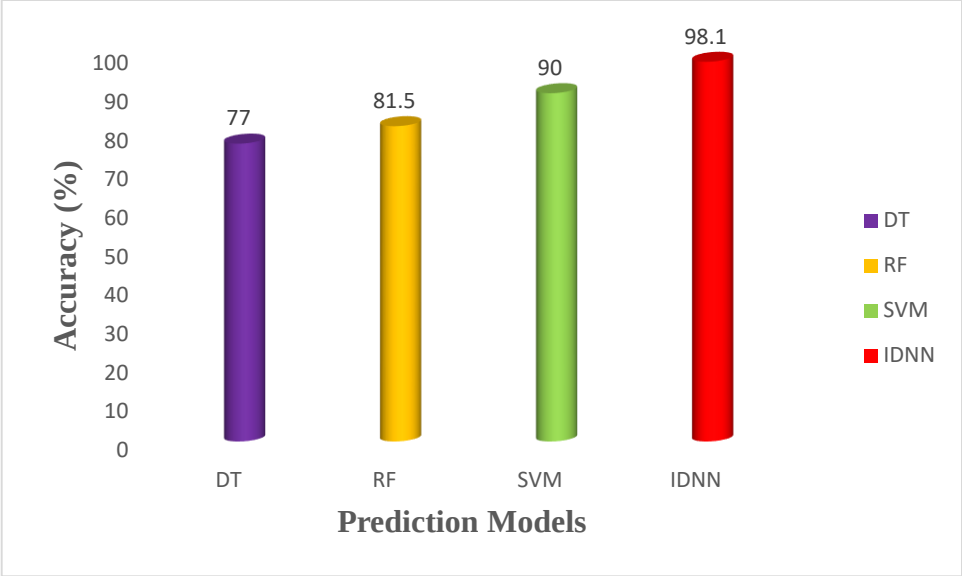

Figure 5: Analysis Based on Accuracy

The result from the figure 5 explores the performance of the four different android malware detection based on the accuracy. The proposed Intelligent Deep Neural Network (IDNN) produce the highest rate of accuracy compared to the other prediction models. The DNN is pretrained using metaheuristic algorithm, Cat Swarm Optimization which understand the pattern of the android malware dataset and speeds up the process by replacing the gradient descent method. The existing classification models Support Vector Machine, Decision Tree and Random Forest suffers from the problem of overfitting and class imbalance, thus these state of arts produce less accuracy rate compared to the proposed IDNN.

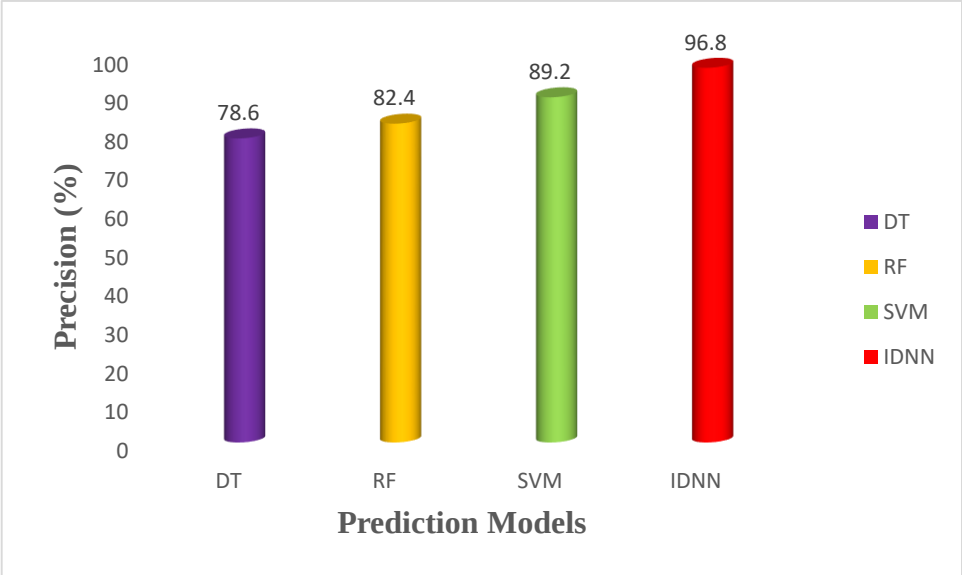

Figure: Analysis Based on Precision

The figure 6 displays precision-based performance comparison of four different prediction modelsto predict malware attacks. The performance of IDNN accomplish higher rate of precision because during pretraining process, the pattern of normal and attack instances is deeply investigated using deep neural network and utilization of cat behaviour for assigning weight values in hidden nodes greatly influence prediction of attacking applications very effectively.

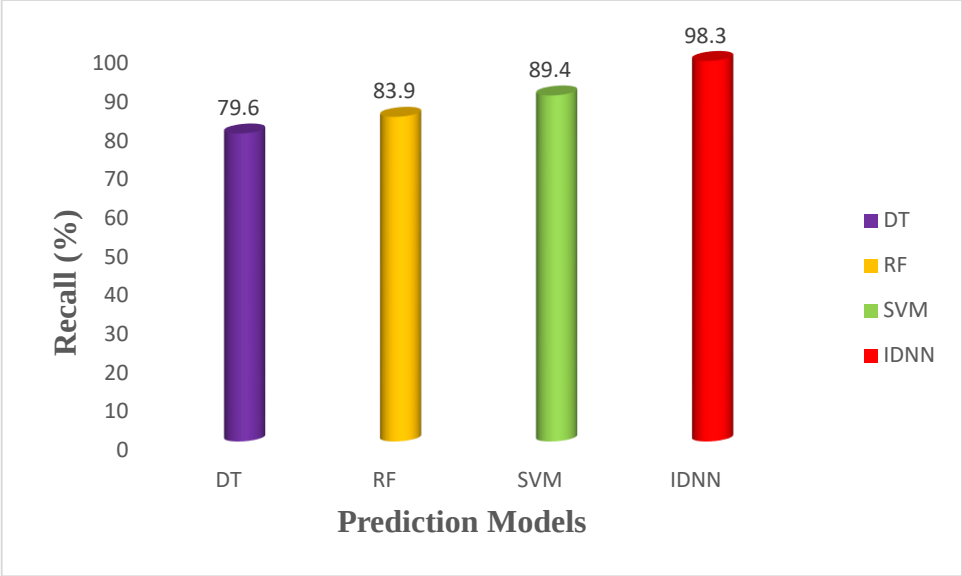

Figure: Analysis Based on Recall

The recall value-based performance analysis for malware attack detection obtained by four different models has illustrated in the figure 7. The behaviour of cat, especially its seeking and tracing mode is used to assign optimized weights to achieve the best recall value with depth knowledge of normal and malware apps characteristics.The dense layer of DNN plays important role to infer input patterns in-depth. The hidden nodes have activated based on the weights and bias assigned. The decision tree, random forest and support vector machine suffers by class imbalance by over fitting issues and produced less precession value compared to the proposed IDNN.

Conclusion

In this research paper a behavior based deep learning network has constructed to detect the android malware attack in smart phones. Though, there are different types of cyberattacks occur but malicious application attacks in smart phones have gained the lead. Thus, the proposed model understands the depth pattern of the benign applications and malicious applications by designing Intensified Deep Neural Network (IDNN) with the aid of Cat swarm behavioural based optimization has improved the detection of malware attacks more precisely then the conventional classification models. The seeking and tracking nature of the cat is utilized in this work to discover best weight values to be assigned to the hidden layers of the DNN. Because the standard deep neural network used the gradient descent to assign the values of the weights during the training phase, where it has done in a random manner and results in higher rate of misclassification. From the results obtained, it provides a consistent proof about newly constructed IDNN accomplishes higher efficiency and accuracy in detection of android malware attacks by handling the class imbalance and overfitting using intelligence of cat swarm behavior.

References

1. Enck W, Defending users against smartphone apps: Techniques and future directions, Intelligence and Lecture Notes in Bioinformatics, LNCS, 49–70, 2011.
2. Arp, D., Spreitzenbarth, M., Malte, H., Gascon, H., &Rieck, K. (2014). Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. In Symposium on Network and Distributed System Security (NDSS) (pp. 23–26).
3. Namrud, Z.; Kpodjedo, S.; Talhi, C. AndroVul: A repository for Android security vulnerabilities. In Proceedings of the 29thAnnual International Conference on Computer Science and Software Engineering, Toronto, ON, Canada, 4–6 November 2019;pp. 64–71
4. Zhuo, L.; Zhimin, G.; Cen, C. Research on Android intent security detection based on machine learning. In Proceedings of the2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 21–23, 2017, pp. 569–574
5. Garg S, Baliyan N. Data on Vulnerability Detection in Android, Data in Brief, Volume 22, Pages 1081-1087, 2019.
6. Tieming Chen, Qingyu Mao, Yimin Yang, MingqiLv, Jianming Zhu, TinyDroid: A Lightweight and Efficient Model for Android Malware Detection and Classification, Mobile Information Systems, vol. 2018, 9 pages, 2018.
7. Pan, Y, Ge X, Fang, C, Fan Y, A Systematic Literature Review of Android Malware Detection Using Static Analysis. IEEE Access 2020, 8, 116363–116379
8. Oluwakemi Christiana Abikoye, Benjamin AruwaGyunka Android Malware Detection through Machine Learning Techniques: A Review, International Journal of Online and Biomedical Engineering, Vol. 16, No. 2, 2020, pp 14-29
9. B. Rashidi, C. Fung and E. Bertino, "Android malicious application detection using support vector machine and active learning," 2017 13th International Conference on Network and Service Management (CNSM), 2017, pp. 1-9

10. Milosevic N, Dehghantanha, A., Choo K. K. R. (2017). Machine learning aided Android malware classification. Computers and Electrical Engineering, 61, 266–274.
11. Shivi Garg, NiyatiBaliyan, Android Malware Classification using Ensemble Classifiers, Cloud Security, 1st Edition, Pages13, 2021
12. S. Sabhadiya, J. Barad and J. Gheewala, "Android Malware Detection using Deep Learning," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp1254-1260.
13. E. B. Karbab, M. Debbabi, A. Derhab and D. Mouheb, "MalDozer: Automatic framework for android malware detection using deep learning", *Digital Investigation*, vol. 24, 2018.
14. Mcdonald, J.; Herron, N.; Glisson, W.; Benton, R. Machine Learning-Based Android Malware Detection Using ManifestPermissions. In Proceedings of the 54th Hawaii International Conference on System Sciences, Maui, HI, USA, 5–8 January 2021;p. 6976.
15. Li, J.; Sun, L.; Yan, Q.; Li, Z.; Srisa-An, W.; Ye, H. Significant permission identification for machine-learning-based androidmalware detection. IEEE Trans. Ind. Inform. 2018, 14, 3216–3225.
16. Zhuo, L.; Zhimin, G.; Cen, C. Research on Android intent security detection based on machine learning. In Proceedings of the2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 21–23 July 2017;pp. 569–574.