IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

IJARSCT

Impact Factor: 7.301

Volume 3, Issue 11, May 2023

Botnet Detection using SVM Algorithm

Yash Birdavade[1], Ram Todkar[2], Aashish Walke[3] ,Sambhaji Shinde[4], Dr. Neelam A Kumar[5]

Students, Department of Computer Engineering[1,2,3,4]

Professor, Department of Computer Engineering[5]

Shree Ramchandra College of Engineering Pune, Maharashtra, India

Abstract*: The proliferation of mobile devices and their increasing connectivity have led to the emergence of mobile botnets, posing a significant threat to the security and privacy of users. Mobile botnets are networks of compromised mobile devices controlled by malicious actors for various illicit activities, including spamming, distributed denial of service (DDoS) attacks, and information theft. Traditional security measures are often insufficient to detect and mitigate mobile botnet attacks due to their dynamic and stealthy nature. This project proposes a novel approach for mobile botnet detection using machine learning techniques. The objective is to develop a robust and accurate system that can identify and classify mobile devices participating in botnets based on their behavioural patterns and network activities.*

Keywords: Machine Learning, SVM Algorithm, Mobile Botnet

I. INTRODUCTION

With the exponential growth of mobile devices and the increasing dependence on them for various tasks, mobile security has become a critical concern. One of the major threats to mobile security is mobile botnets, which are networks of compromised devices controlled by malicious actors. These botnets can be used for a wide range of malicious activities, including distributed denial-of-service attacks, spam dissemination, data theft, and unauthorized access to sensitive information. Detecting and mitigating mobile botnets is a challenging task due to the evolving nature of these threats. Traditional security mechanisms often fall short in identifying and neutralizing sophisticated botnet attacks. However, advancements in machine learning techniques provide promising solutions for mobile botnet detection. This project aims to explore the application of machine learning algorithms in detecting mobile botnets and enhancing mobile security. By leveraging the power of machine learning, we can develop intelligent systems capable of analysing vast amounts of data and identifying botnet activities based on behavioural patterns, network traffic analysis, and other relevant features. [1]

1.1 Objective

- To analyse and understand the characteristics and behaviour of mobile botnets.
- To identify and collect relevant datasets containing labelled instances of botnet-infected mobile devices.
- To explore and select appropriate machine learning algorithms for mobile botnet detection

II. LITERATURE REVIEW

M. Eslahi C.Kang-yu et al.in To Study different algorithms of Machine Learning to Detect Mobile Botnets.Botnets pose a significant threat to Internet security, and their constant sophistication and resilience have led to a new trend of shifting from desktop to mobile environments. Detecting mobile botnets is crucial to mitigate their impact. Among various strategies, identifying patterns in their anomalous behaviour has shown promising and generalized results. In this research, we provide a host-based and anomaly-based method for mobile botnet detection. Our suggested approach makes use of machine learning methods to spot unusual patterns in statistical characteristics taken from system calls. We evaluated the performance of our high accuracy. These Android bots can be enlisted in larger botnets, enabling various types of attacks such as DDoS attacks, spam generation, phishing, click fraud, and data theft This research provides a deep learning method that uses Support Vector Machine (SVM) for Android botnet detection to counteract this expanding danger. Our approach utilizes 342 static features extracted through automated reverse engineering of

DOI: 10.48175/IJARSCT-10630

apps to classify new or previously unseen apps as either 'botnet' or 'normal'. These features are directly fed into the SVM model without requiring additional pre-processing or feature selection [1]

Falguni Ghatkar et al in ,” To Study Different Types of Supervised Learning Algorithm” paper states that A Concepts and principles underlying machine learning, including supervised, unsupervised, and reinforcement learning. It then delved into an extensive discussion of popular machine learning. algorithms such as linear regression, decision trees, support vector machines, random forests Among Others. Each algorithm was critically evaluated based on its strengths, weaknesses, and suitability for different types of dataand problem domains.

Zubail Abdullah, Madihah Mohd Saudi et al.in ABC: Android botnet classification Using feature selection and classification algorithms According to the advanced wireless technology in nowadays, most people mainly use their mobile phones as an essential tool. Mobile phone risks including viruses, botnets, and other malware are also growing at the same time. However, most users have the limited of knowledge about mobile threats [3]

Dr. K. Muthu Manickam Dr. T. Sengolrajan et al.in Smartphone Based Botnet Detection using Behavioral Analysis In this Paper, we proposed a study on bot network detection both in wired and wireless network with their detection approaches [4]

Suleiman Y. Yerima and Mohammed K. Alzaylaee et al.in Mobile Botnet Detection: A Deep Learning Approach Using Convolutional Neural Networks In this paper, we proposed a deep learning model based on 1D CNN for the detection of Android botnets. We evaluated the model through extensive experiments with 1,929 botnet apps [5]

ArashMahboubi, SeyitCamtepe Stochastic et al.in Modeling of IoT Botnet Spread: A Short Survey on Mobile Malware Spread Modeling Many cyber criminals are financially motivated to develop new forms of malware. Botnet-based malware has seen a tremendous growth as a result of the widespread adoption of IoT devices. The Mirai botnet malware, mainly consisting of embedded systems and IoT devices, has invaded network services and overwhelmed several high-profile organizations with massive DDoS attacks [6]

Rodrigo S. Miani, Bruno B. Zarpelao et al.in Detecting Mobile Botnets Through Machine Learning And System Calls Analysis An efficient approach to detect bot malware is needed to neutralize the threat that botnets impose. This approach must have high detection rates, maintaining a low false positive rate while also minimizing the time required to identify this malware [7]

III. SYSTEM ARCHITECTURE

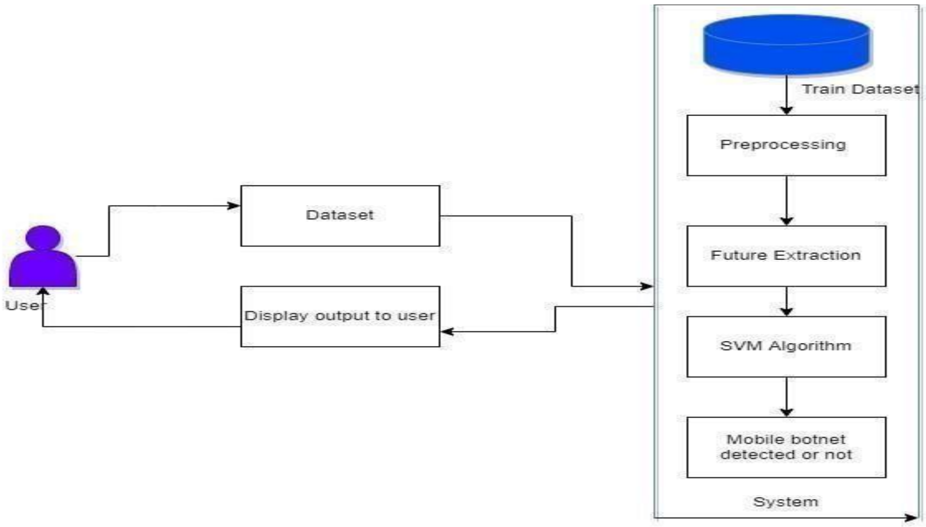


Fig no. 1: Mobile Botnet System Architecture

3.1 DATASET

In mobile botnet detection, a dataset refers to a collection of mobile device data that is used to train, validate, and test machine learning models for identifying botnet-infected devices. The dataset plays a crucial role in developing an effective botnet detection system as it provides the necessary information to analyse and learn from.

DOI: 10.48175/IJARSCT-10630

3.2 PREPROCESSING

Pre-processing in mobile botnet detection refers to the steps taken to prepare the dataset before it is used for training or testing machine learning models. Pre-processing plays a critical role in cleaning and transforming the dataset to ensure its quality and suitability for effective botnet detection.

3.3 FUTURE EXTRACTION

Feature extraction in mobile botnet detection refers to the process of identifying and selecting relevant characteristics or attributes from raw data collected from mobile devices. The goal is to transform the raw data into a set of informative features that can effectively represent the underlying patterns and behaviours associated with botnet activities. Feature extraction plays a crucial role in developing accurate and efficient botnet detection models

3.4 SVM ALGORITHAM

In mobile botnet detection, the Support Vector Machine (SVM) algorithm is a commonly used machine learning technique. SVM is a supervised learning algorithm that can be used for classification tasks, making it well-suited for identifying botnet-infected devices from mobile device data.

3.5 MOBILE BOTNET DETECTED OR NOT

If parameter value is equal to the 1 (one) then mobile botnet are detected and parameter value is equal to the 0 (zero) then mobile botnets are not detected.

IV. ALGORITHM

4.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is a popular algorithm used in various machine learning tasks, including mobile botnet detection. SVMs are effective in distinguishing between different classes by finding an optimal hyperplane that maximally separates the data points. Here are the general steps involved in using SVM for mobile botnet detection:

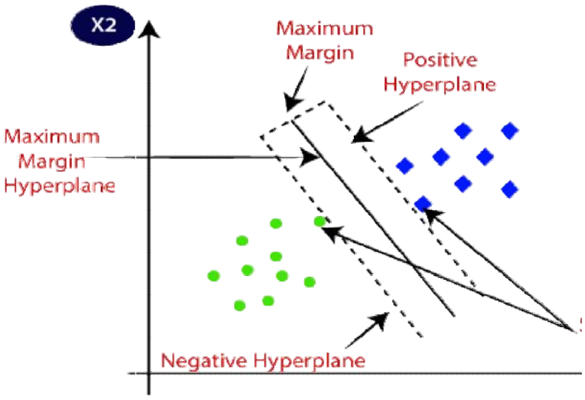


Fig No. 2: Support Vector Machine (SVM)

Step1. Data Collection: Gather a dataset consisting of features that are relevant to mobile botnet detection. These features could include network traffic patterns, system call traces, application behaviour, or any other relevant information.

Step2. Data Pre-processing: Clean the dataset by removing any irrelevant or redundant features and handle missing data if necessary. It's also important to balance the dataset to ensure an equal representation of both normal and botnet instances

Step 3. Feature Selection/Extraction: Select or extract the most informative features from the dataset. This step aims to reduce the dimensionality of the data and improve the model's performance.

DOI: 10.48175/IJARSCT-10630

Step 4. Training and Testing Data Split: Divide the dataset into two parts: a training set and a testing set. The training set is used to train the SVM model, while the testing set is used to evaluate its performance.

Step 5. Feature Scaling: Normalize or scale the features to ensure that they are on a similar scale. This step is crucial because SVMs are sensitive to the scale of the input features.

Step 6. SVM Model Training: Train the SVM model using the training data. In SVM, the algorithm finds an optimal hyperplane that maximally separates the instances belonging to different classes. The choice of the SVM kernel (e.g., linear, polynomial, radial basis function) depends on the data characteristics and problem domain.

Step 7. Model Evaluation: Evaluate the trained SVM model using the testing set. A few common evaluation metrics are F1-score, recall, accuracy, and precision. These metrics help assess the performance of the model in detecting both normal and botnet instances.

Step 8. Hyperparameter Tuning: SVM has various hyperparameters that can impact its performance, such as the regularization parameter (C) and the kernel parameters. Perform hyperparameter tuning using techniques like grid search or random search to find the best combination of hyperparameters for your specific problem.

Step 9. Model Deployment: Once the SVM model is trained and optimized, it can be deployed for real-time mobile botnet detection. The model can take input from live mobile devices, analyse the relevant features, and classify instances as normal or belonging to a botnet.

V. PERFORMANCE MATRIX

Precision: A measure of precision counts how many correctly positive forecasts were made. As a result, accuracy for the minority class is determined by precision. It is determined by dividing the total number of correctly anticipated positive examples by the ratio of correctly predicted positive examples.

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

Recall

Recall is a metric that quantifies the number of correct positive predictions made from all positive predictions that could have been made. Unlike precision that only comments on the correct positive prognoses Recall shows which positive predictions were missed out of all positive predictions.

$$Recall = \frac{TP}{P}$$

F-Measure

Precision and recall can be combined into one metric using F-metric, which covers both characteristics. Precision and memory don't give the complete tale on their own. We can have great precision but poor recall, or vice versa, great precision but poor recall. A way to communicate both concerns with a single score is offered by the F-measure. For a binary or multiclass classification task, precision and recall can be measured; the two scores can then be combined to calculate the F-Measure.

$$F\ Score = \frac{2TP}{2TP + FP + FN}$$

Accuracy

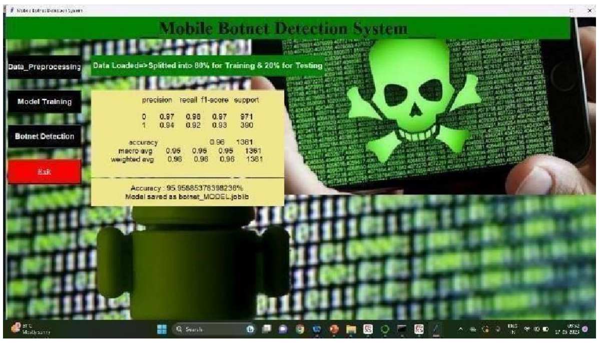
The number of accurate forecasts divided by the total number of predictions generated by the model is how accuracy is calculated.

$$accuracy = \frac{(TP + TN)}{TP + FP + TN + FN}$$

DOI: 10.48175/IJARSCT-10630

VI. RESULT

In the context of mobile botnet detection, a performance matrix refers to the set of metrics used to evaluate the effectiveness and accuracy of a botnet detection system. These metrics help assess the performance of the detection algorithm or methodology in identifying and classifying mobile devices that are infected or participating in botnet activities.

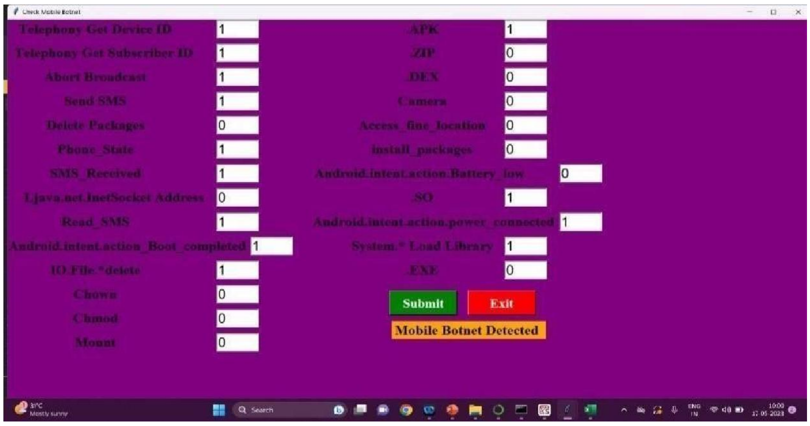

Mobile Botnet Detection System
Mobile Botnet Detection System
Data_Preprocessing
Model Training
Botnet Detection
Exit

Data Loaded=>Splitted into 80% for Training & 20% for Testing

```
              precision    recall  f1-score   support

           0       0.97      0.98      0.97       971
           1       0.94      0.92      0.93       390

    accuracy                           0.96      1361
   macro avg       0.95      0.95      0.95      1361
weighted avg       0.96      0.96      0.96      1361

Accuracy : 95.95885378398236%
Model saved as botnet_MODEL.joblib
```

Fig no. 3: Performance Matrix of SVM

If parameter value is equal to the 1 (one) then mobile botnet are detected

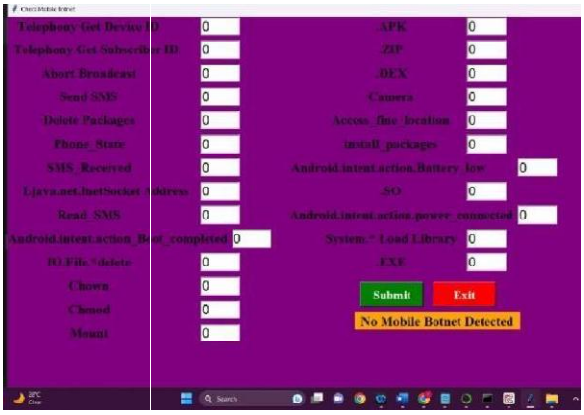


Telephony Get Device ID: 1
Telephony Get Subscriber ID: 1
Abort Broadcast: 1
Send SMS: 1
Delete Packages: 0
Phone_State: 1
SMS_Received: 1
Ljava.net.InetSocket Address: 0
Read_SMS: 1
Android.intent.action_Boot_completed: 1
IO.File.*delete: 1
Chown: 0
Chmod: 0
Mount: 0
.APK: 1
.ZIP: 0
.DEX: 0
Camera: 0
Access_fine_location: 0
install_packages: 0
Android.intent.action.Battery_low: 0
.SO: 1
Android.intent.action.power_connected: 1
System.* Load Library: 1
.EXE: 0

Mobile Botnet Detected

Fig no. 4: Output of Mobile Botnet Detected

If parameter value is equal to the 0 (zero) then mobile botnet are Not detected



Telephony Get Device ID: 0
Telephony Get Subscriber ID: 0
Abort Broadcast: 0
Send SMS: 0
Delete Packages: 0
Phone_State: 0
SMS_Received: 0
Ljava.net.InetSocket Address: 0
Read_SMS: 0
Android.intent.action_Boot_completed: 0
IO.File.*delete: 0
Chown: 0
Chmod: 0
Mount: 0
.APK: 0
.ZIP: 0
.DEX: 0
Camera: 0
Access_fine_location: 0
install_packages: 0
Android.intent.action.Battery_low: 0
.SO: 0
Android.intent.action.power_connected: 0
System.* Load Library: 0
.EXE: 0

No Mobile Botnet Detected

Fig no. 5: Output of Mobile Botnet Not Detected

VII. CONCLUSION

Mobile botnet detection is a crucial area of research and development aimed at identifying and mitigating the threats posed by malicious botnets on mobile devices. Detecting and combating mobile botnets is vital for protecting users' privacy, preventing financial fraud, and ensuring the stability and security of mobile networks. [2] The conclusion of mobile botnet detection using Support Vector Machines (SVM) would depend on the specific implementation and evaluation of the SVM-based detection system. However, I can provide you with a general conclusion based on the potential advantages and limitations of using SVM for mobile botnet detection. matrix is a useful tool for evaluating the performance of a detection system. It provides a tabular representation of the predicted outcomes compared to the actual outcomes. The confusion matrix is typically constructed using the results of testing a detection system against a set of labelled data, where each data point is classified as either belonging to a botnet (positive class) or not belonging to a botnet (negative class)

REFERENCES

[1]. M. EslahiC.Kang-yu et al.in To Study different algorithms of Machine Learning to Detect Mobile Botnets.

[2]. Falguni Ghatkar, Sakshi kharche, Priyanka Doifode, Jagruti Khairnar, Prof. Neelam Kumar,” To Study Different

[3]. Types of Supervised Learning Algorithm” May 2023, International Journal of Advanced Research in Science, Communication and Technology.

[4]. Zubail Abdullah, Madihah Mohd Saudi et al.in ABC: Android botnet classification Using feature selection and classification algorithms.

[5]. Dr. K. Muthu Manickam Dr. T. Sengolrajan Smartphone Based Botnet Detection using Behavioral Analysis.

[6]. Suleiman Y. Yerima and Mohammed K. Alzaylaee Mobile Botnet Detection: A Deep Learning Approach Using Convolutional Neural Networks.

[7]. ArashMahboubi, SeyitCamtepe Stochastic Modeling of IoT Botnet Spread: A Short Survey on Mobile Malware.

[8]. S. Y. Yerima and S. Khan “Longitudinal PerfomanceAnlaysis of Machine Learning based Android Malware Detectors” 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), IEEE.

[9]. Kadir, A.F.A., Stakhanova, N., Ghorbani, A.A., 2015. Android botnets: What urls are telling us, in: International Conference on Network and System Security, Springer. pp. 78–91.

[10]. ISCX Android botnet dataset. Available from https://www.unb.ca/cic/ datasets/androidbotnet.html. [Accessed 03/03/2020]

[11]. M. Eslahi, R. Salleh, and N. B. Anuar, ”Bots and botnets: An overview of characteristics, detection and challenges,” in Proceedings of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2012, pp. 349-354.

[12]. J. Dae-il, C. Kang-yu, K. Minsoo, J. Hyun-chul, and N. Bong-Nam, ”Evasion technique and detection of malicious botnet,” in Proceedings of the Internet Technology and Secured Transactions (ICITST), 2010 International Conference for, 2010, pp. 1-5.

[13]. Z. Yajin and J. Xuxian, ”Dissecting Android Malware: Characterization and Evolution,” in Proceedings of the Symposium on Security and Privacy (SP), 2012, pp. 95-109.

[14]. M. Eslahi, M. Rohmad, H. Nilsaz, M. Var Naseri, N. Tahir, and H.Hashim, ”Periodicity classification of

[15]. HTTP traffic to detect HTTP Botnets,” in Proceedings of the Computer Applications Industrial Electronics (ISCAIE), 2015

[16]. T. Strazzere and T. Wyatt, ”Geinimi trojan technical teardown,” Lookout Mobile Security, 2011.

[17]. Gu, R. Perdisci, J. Zhang, and W. Lee, ”BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection,”in Proceedings of the 17th conference on Security symposium, San Jose, CA, 2008, pp. 139-154.

[18]. M. Eslahi, R. Salleh, and N. B. Anuar, ”MoBots: A new generation of botnets on mobile devices and networks,” in Proceedings of the IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), 2012, pp. 262- 266.