www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 13 Issue 07 July 2024, Page No. 26274-26283
ISSN: 2319-7242 DOI: 10.18535/ijecs/v13i07.4846

A Model for Detection of Malwares on Edge Devices

Nwagwu, C .B., Taylor O. E. & Nwiabu, .N.D.

Department of Computer Science,
Rivers State University,Port Harcourt, Nigeria.

Abstract-
Malware detection is a significant challenge in today's digital landscape. As new forms of malware are continuously being developed, traditional detection techniques often fall short due to their inability to detect these new strains. This paper introduces meaningful features that effectively capture various types of malware, including viruses, worms, Trojans and Ransomware on Edge devices. The paper used a model that implemented Random forest classifier for feature selection and a support vector machine (SVM) model for Malware detection. Object-Oriented Analysis and Design (OOAD) methodology was used to as the design methodology, which involved identifying and modeling the different components of the system and their interactions. The system was developed using Python programming language, with an emphasis on model deployment via Python Flask for web-based testing and execution. The experimental results demonstrate the effectiveness of the proposed systems when compared with other existing system. The result gotten from proposed system is better than that of the existing system by achieving a detection accuracy of 99.98% which is better than existing techniques. This dissertation presents a promising direction for improving malware detection using support vector machine (SVM) model and highlights the potential for collaborative learning approaches to overcome the challenges of traditional centralized approaches. This result simulates edge device that performs malware detection. It measures the latency for each detection and prints whether the latency is high or low. After the simulation, it plots a graph to visualize the latency over multiple requests. Which shows that the proposed model had low latency between 0.25secs to 0.15 secs on multiple requests.

Keywords- Malwares, Support Vector Machine, Random Forest Classifier, Python Flask.

1. **Introduction**

Malware, also known as malicious code, is a significant concern in cybersecurity. While the technical aspects of malware are often emphasized, it is crucial to acknowledge the human element involved in its creation and deployment (King *et al.*, 2018). The evolution of malware has resulted in sophisticated forms that present challenges for detection and mitigation, with ransomware being a notable type that encrypts data and demands ransom for decryption (Pedreira *et al.*, 2021). Research indicates that malware can infiltrate various components of communication networks, highlighting the importance of robust security measures such as antivirus software for detecting and removing malicious software (Eichelberg *et al.*, 2020). The healthcare sector has also been impacted by malware attacks, leading to unauthorized access to sensitive patient data (Moore & Frye, 2019).

Malware is the short form of malicious software or application which is not limited to computer system rather extend to the internet and related fields. Viruses and the Rise of the Internet grow gradually and significantly back, with only four hosts on the internet back in 1969, statistic shows the total number reached approximately 1.01 billion in 2019. In order to prevent these attacks and catastrophes, scientists around the world attempt to design security tools and antivirus packages that are mainly used to prevent, detect, avoid, and remove viruses, Trojans, worms, etc, whereas firewalls are used to monitor incoming and outgoing connections. (Patil & Joshi, 2012). Malware Detection is done using an anti-malware software. The anti-malware is a program that is designed to fight against malware. It protects the computer and ensures that it is malware free by scanning it regularly. (Dovom *et al.*, 2019) Users may be able to detect malware if they observe unusual activity such as a sudden loss of disk space, unusually slow speeds, repeated crashes or freezes, or an increase in unwanted internet activity and pop-up advertisements.

Edge devices play an important role in data accumulation, pre-processing, and real-time decision making. Edge devices include smartphones, PCs, tablets, and small servers that also act as intermediaries between Internet-of-things (IoT) devices and remote servers (i.e., cloud) where big data analytics are performed.

An edge device is the network component responsible for connecting your local area network to an external and wide area network, where you can collect data from everywhere. The edge device is responsible for providing the local information to an external network. If you have different protocols, it also translates this information, making the connection between both network boundaries. (Fabricio, 2019).

Types of malwares include: computer viruses, worms, Trojan horses, ransomware and spyware. These malicious programs steal, encrypt and delete sensitive data; alter or hijack core_computing functions and monitor end users' computer activity.

Viruses are malicious software that can replicate themselves and infect files on your computer. Trojans are malicious software that masquerades as something legitimate, like a helpful PDF reader, but actually do something harmful. Worms are software that spreads across networks and computers. Spyware collects information about you and your computer's activity without your knowledge. Ransomware can lock your computer or your files until you pay a ransom. Malware can do many different things, but you can protect yourself by keeping your devices clean and being careful about what you download. (Iyas *et al.* 2022). These integrated internet of things devices could use different security measures, leading to a lack of standardization in the network. Each connected device could use different security protocols, with their security bugs and limitations, exposing the system to different kinds of hacking. The number of internet of things devices is increasing exponentially and is generating an unprecedented amount of data. The expected number of IoT devices by 2025 is between 25 billion and 50 billion. (Zhang *et al*., 2020). The IoT devices are heterogeneous as they may be built on different platforms and have different specifications. The hardware, such as a simple sensor to monitor the heart rate. (Khan & Algarni, 2020). Furthermore, concerns have been raised about the potential misuse of synthetic biology at the intersection of biotechnology and cybersecurity, including the creation of bio-malware that could compromise systems through DNA-encoded malicious code (Elgabry et al., 2020).

2. Review of Related Works

Kitsune *et al.* (2022) Proposed an anomaly detection system based on recurrent neural network model Accuracy: 98%. Testing time is not calculated. DS1-D1, DS1-P Presented an IoT attack detection scheme Accuracy: 98.8%. Old dataset used. Self-generated DL-based federated technique is presented for multiparty computation with the security of IoT devices Accuracy: 56%. Algorithm is unable to achieve promising detection accuracy.

Ma *et al*. (2016). Eexplores detecting malicious websites from the lexical and host-based features of their URLs. They develop a real-time system for gathering URL features, and pair it with a real-time feed of labeled URLs from a large webmail provider. From these features and labels, they are able to train an online classifier for detecting malicious websites with 99% of precision over a balanced dataset.

Baptista *et al*. (2019). Introduces a novel approach for malware detection by utilizing binary visualization and self-organizing incremental neural networks. A demonstration was conducted on detecting malicious payloads in various file types including Portal Document File .pdf and Microsoft Document Files .doc files where the experimental results indicate a detection accuracy of 91.7% and 94.1% for ransomware respectively. According to the authors, the proposed technique performed well with an incremental detection rate, allowing for efficient real-time identification of unknown malware

Sharma et al., (2017). Proposes an approach based on opcode occurrence to detect malware using machine learning techniques. The researchers also use a dataset from the Kaggle Microsoft malware classification challenge dataset and evaluate five classifiers including LMT, REPTree, Random Forest, NBT, J48Graft. A demonstration indicates that the proposed approach capable of detecting the malware with almost 100% accuracy.

Blum *et al*., (2017). Explore the possibility of utilizing a confidence-weighted classification combined with content-based phishing URL detection, to produce a dynamic and extensible system for the detection of present and emergent types of phishing domains. Their system is capable of detecting emerging threats as they appear, and subsequently can provide increased protection against zero-hour threats, unlike traditional blacklisting techniques (which function reactively).

Hsu *et al.,* (2020). Implemented the privacy-preserving federated learning system based on support vector machine (SVM) and secure multi-party computation techniques. It also demonstrates the feasibility using the Android malware dataset by National Institute of Information and Communication Technology (NICT), Japan. The presented experiments evaluate the performance of the trained classifier by the proposed PPFL system. The evaluation also compares the performance of the classifier of PPFL and that of centralized training system for the use cases of i) different data set and ii) different features on distinct mobile device. The results show that the performance of the PPFL classifier outperforms that of centralized training system. Moreover, the privacy of app information (i.e., API and permission information) and trained local models is guaranteed.

Touceda *et al*., (2015). Pproposed an attribute based authentication for permission in to peer-to-peer network; the proposed system is aimed at allocating privileges without third party. In (Yang, *et al* 2016) proposed a system that allow mobile device users to move from one geographical location to the other with intractability, authentication is based on handover, clients identity and location are kept with the aid of the elliptic curve algorithm cryptography.

Niharika *et al.,* (2021). Presents a detailed analysis of the static, dynamic, and hybrid methods with an evaluation of malware detection techniques. The author also facilitates the detection process by combining machine learning and data mining techniques.

Another solution was proposed by Kim *et al*., (2018). The solution is a real time ransomware detection technique without the inclusion of a trust party, the proposed method increase access control policy to the task process of an OS on user's device, which will have disallowed unauthorized application from initiating or editing in the device, it is white list based, applications are opened or edited if and only if they belong to the white list. The layers are: Strong Trap layer takes care of the early detection of ransomware, machine learning ensure a zero-day intrusion, and finally a File Backup layer for maintaining user files, with these layers Ransom Wall attain a detection rate of 98.25% with a near zero false positive using the Gradient Tree Boosting algorithm but the model has not been evaluated on a large scale real setup which is a limitation of the work. A lot of ransomware solutions are designed using machine learning approach, they include Cheng *et al*., (2007). The paper "Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks" discusses the detection method based on modified Random Forest algorithm in combination with Information Gain for better feature representation. It should be noticed, that the data set consists purely of portable executable files, for which feature extraction is generally easier. The result achieved is the accuracy of 97% and 0.03 false positive rate. (Singhal and Raul (2015).

Taylor *et al*., (2021) Presents a Deep Learning Based Approach for malware detection. The system used a malware dataset which was made up of 19612 files of both malware files and benign files which comprises of both malware files and benign files which comprises of 72 columns. The Deep Learning Model was trained using a Deep Forward Neural Network Algorithm with an input data of 18929, a dense layer of three, two, input layers and one output layer, batch size of 32 and an epoch value of 50. After training the system obtained an accuracy of about 99.94%.

Onyedeke *et al*., (2020). In a signature-based method, developers use a database containing signatures of viruses, scan the file, and evaluate information with that database for detecting malware in the database. If the information matches with the database's data that means the file contains viruses. The primary advantage of this method is effective for the known malware, however, it has limitations in detecting unknown malware.

Syam & Vankata, (2017). Propose a detection way where a virtual analyst was developed by using Artificial Intelligence to defend threats and take appropriate measurements. The researchers categorize supervised and unsupervised data, and later converted unsupervised data to supervised data with the help of analyst feedback and then auto-update the algorithm.

3. Methodology

The system architecture is shown in Figure 1 Proposed system, Support Vector Machine will be used to show properly how they work with the goal of improving accuracy and to reduce high detection latency in the detecting malware on edge devices. The model will be trained on a large dataset consisting of both malware samples and legitimate files.Once the model is trained, it will be deployed using the Python Flask framework to create a web interface. The Flask framework will be responsible for handling user requests and displaying the classification results.

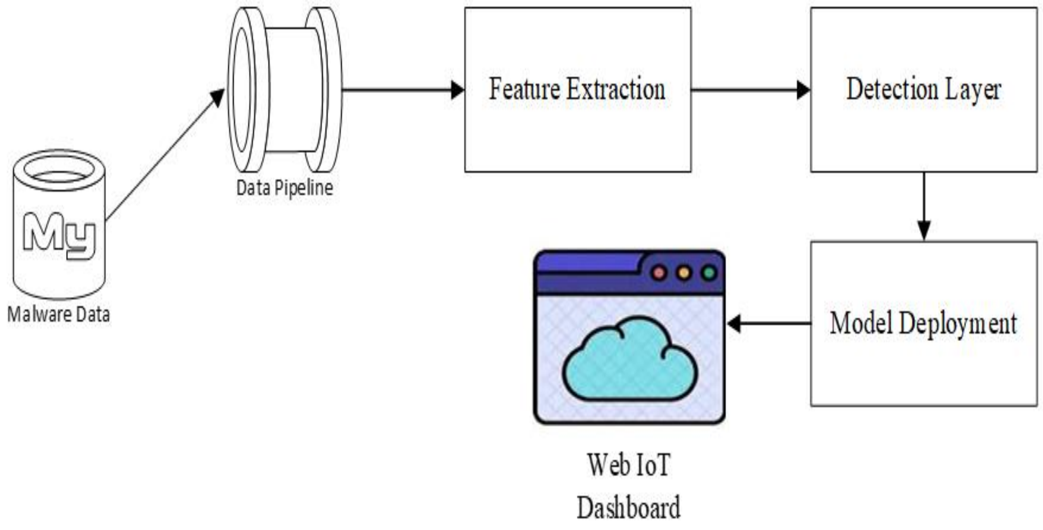


Figure 1: Architecture of the Proposed System

Algorithm 3.1 SVM Algorithm

```
Input: k, m, $C$, y, features, and termination condition
Output: classification accuracy, optimal feature subset, and optimal value for SVM parameters
Begin  $C$ solution archive$_C$  $\gamma$ solution archive$_\gamma$  features solution archive$_{feature}$  call SVM algorithm to
evaluate the initialize solution in solution archive  while classification accuracy $\neq$ 100% or number of
iteration
$\neq$ 10 do   for $n = 1$ to $m_{ants}$ do    call ACO$_{MV\text{-tune SVM parameter}}$
   call ACO$_{MV\text{-feature subset selection}}$  call SVM algorithm to evaluate the newly built solution
     end
    solution archive = first Rank ($S_{old}$ U $S_1, \dots S_{Nants}$)
    update solution archives
   end
      End
```

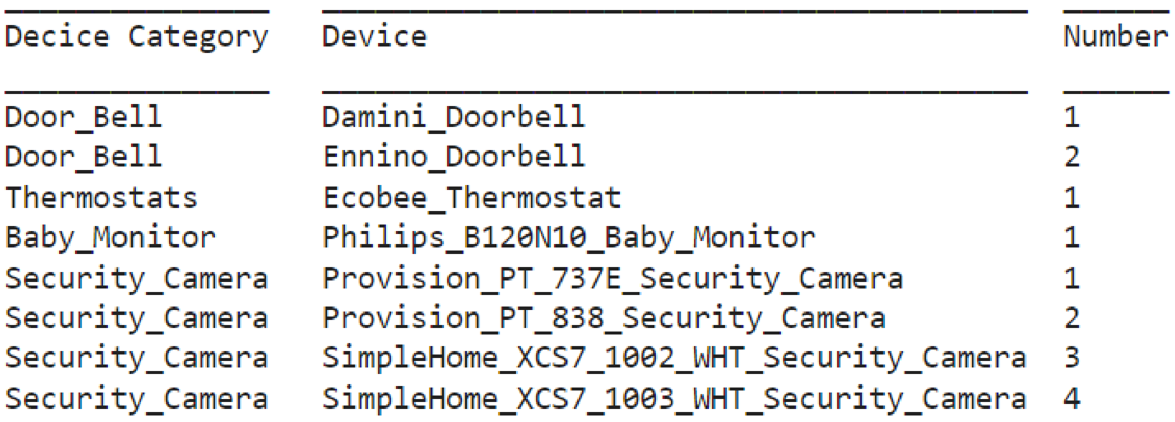
Algorithm 3.2: Random Forest for Feature Selection

1. **Input:**
   a) Training dataset with features X and corresponding labels y.
   b) Number of trees in the Random Forest ensemble (n_estimators).
   c) Number of features to select (k).
2. **Initialize an empty list to store feature importance scores.**
3. **For i from 1 to n_estimators:**
   a) Randomly sample with replacement from the training dataset to create a bootstrap sample.
   b) Train a decision tree on the bootstrap sample.
   c) For each feature in the dataset:
   d) Evaluate the importance of the feature based on how much it decreases impurity or increases accuracy in the decision tree.
   e) Add the importance score of the feature to the list.
4. **Calculate the average importance score for each feature** by taking the mean of importance scores across all trees.
5. **Select the top k features** based on their average importance scores.
6. **Output:**
   a) List of the selected features.

4. Results

The implementation of detecting malware on edge devices, the process begins with data analysis to comprehensively understand the characteristics and patterns of malware. This involves collecting and preprocessing IoT dataset containing both normal and malicious activities from edge devices. The devices affected by malware can be seen in Table1, Figure 1, Figure 2, shows the countplot of the malware attacks on edge devices.

Table 1: Devices Affected by Malware



```
_______________   ________________________________________   ______
Decice Category   Device                                     Number
_______________   ________________________________________   ______
Door_Bell         Damini_Doorbell                            1
Door_Bell         Ennino_Doorbell                            2
Thermostats       Ecobee_Thermostat                          1
Baby_Monitor      Philips_B120N10_Baby_Monitor               1
Security_Camera   Provision_PT_737E_Security_Camera          1
Security_Camera   Provision_PT_838_Security_Camera           2
Security_Camera   SimpleHome_XCS7_1002_WHT_Security_Camera   3
Security_Camera   SimpleHome_XCS7_1003_WHT_Security_Camera   4
```

Some of the important features can be seen in Table 2 shows the ranking of the dataset features. The feature importance values for the first 15 features are provided, indicating their contribution to the model's predictive performance. The most influential feature is "MI_dir_L0.1_weight" with an importance value of 0.100042, followed closely by "MI_dir_L1_weight" at 0.093057, and "H_L0.1_weight" at 0.092163. These features play a significant role in the model's decision-making process, suggesting that variations in these attributes have a pronounced impact on the output.

Table 2: Important Features

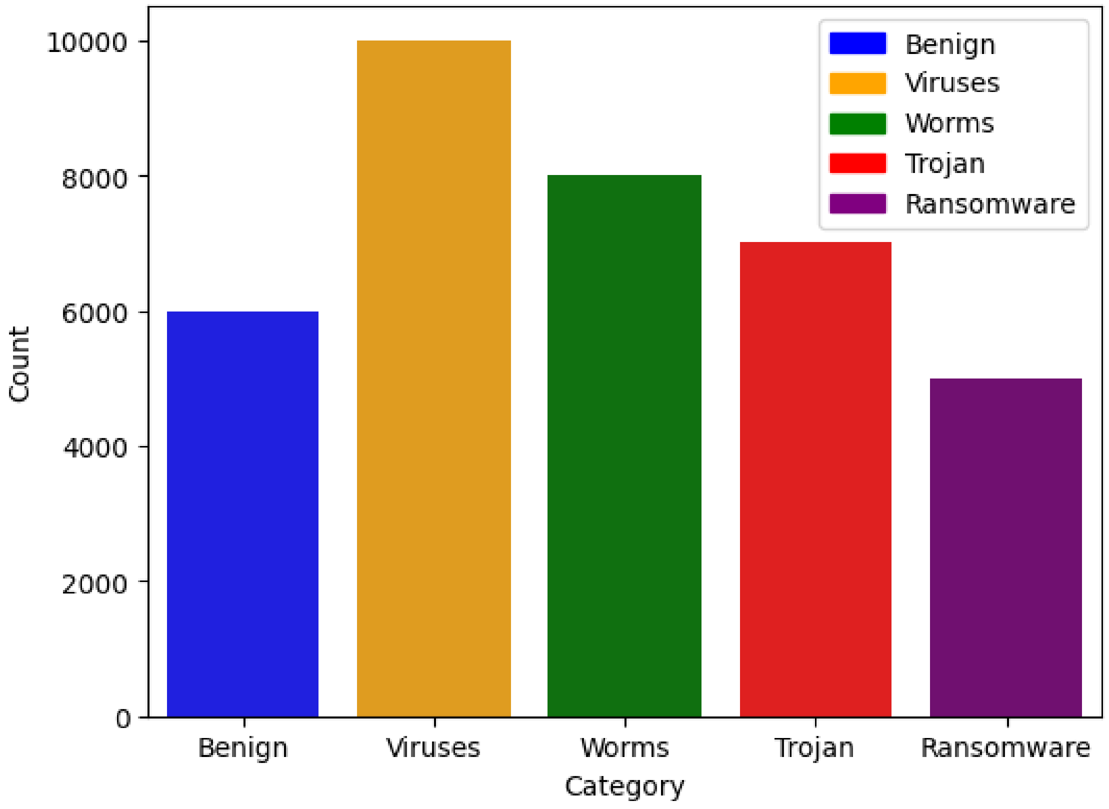
	Feature	Important_Features
9	MI_dir_L0.1_weight	0.100042
6	MI_dir_L1_weight	0.093057
24	H_L0.1_weight	0.092163
12	MI_dir_L0.01_weight	0.073647
27	H_L0.01_weight	0.050845
21	H_L1_weight	0.050375
18	H_L3_weight	0.050309
37	HH_L3_weight	0.050102
13	MI_dir_L0.01_mean	0.050032
28	H_L0.01_mean	0.030861
3	MI_dir_L3_weight	0.030434
0	MI_dir_L5_weight	0.030399
15	H_L5_weight	0.029964
77	HH_jit_L0.01_weight	0.029914

The proposed federated Support Vector Machine (SVM) model evaluated against existing systems in terms of model accuracy. The comparative analysis is presented in Table 3.

Table 3: Acuracy of the Federated SVM Model For Two Round

Accuracy:	99.81\%
TPR:	99.51\%
TNR:	99.82\%
F1-Score:	98.09\%
Accuracy:	99.31\%
TPR:	98.38\%
TNR:	99.36\%
F1-Score:	93.41\%

Figure 2: Countplot of Malware Attacks on Edge Devices



The line plot in Figure 3. Shows the accuracy of the support vector machine across the federated rounds. The accuracy of the federated Support Vector Machine (SVM) model has been consistently high across three rounds of training, with an impressive accuracy rate of 99.93%.

Fig 3: Accuracy of the federated SVM Model Across 3 Rounds (Training).

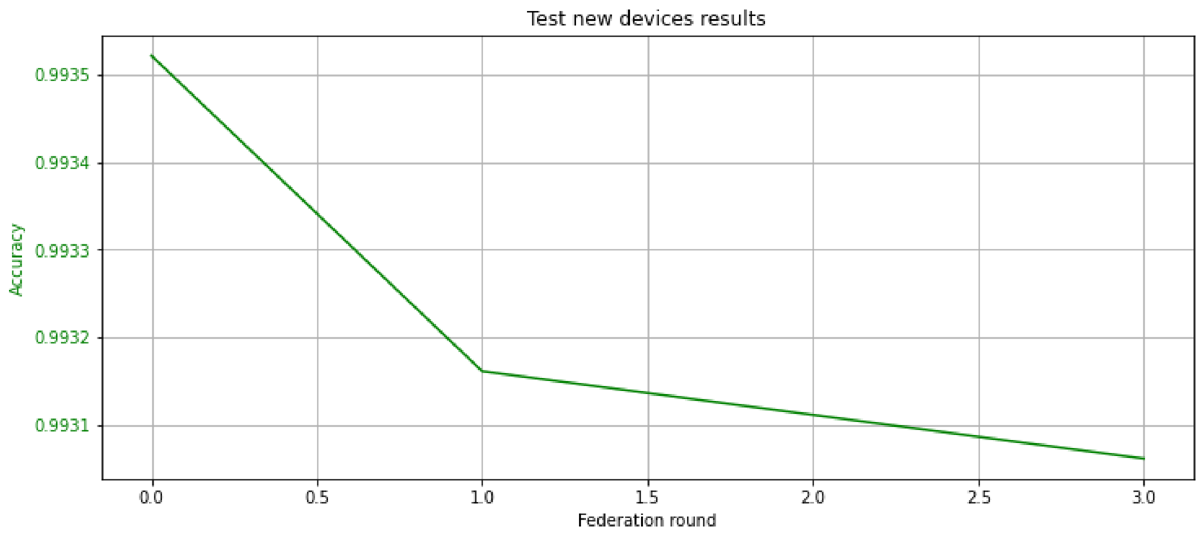


Figure 4. Shows the error rate of the SVM model. The error rate of a federated Support Vector Machine (SVM) model is assessed on both known and unknown devices, employing various training strategies including Naive, Centralized, Mini Epoch, and Multiple Epoch approaches.

Figure 4: Error Rate of the Federated SVM Model on Known and Unknown Devices

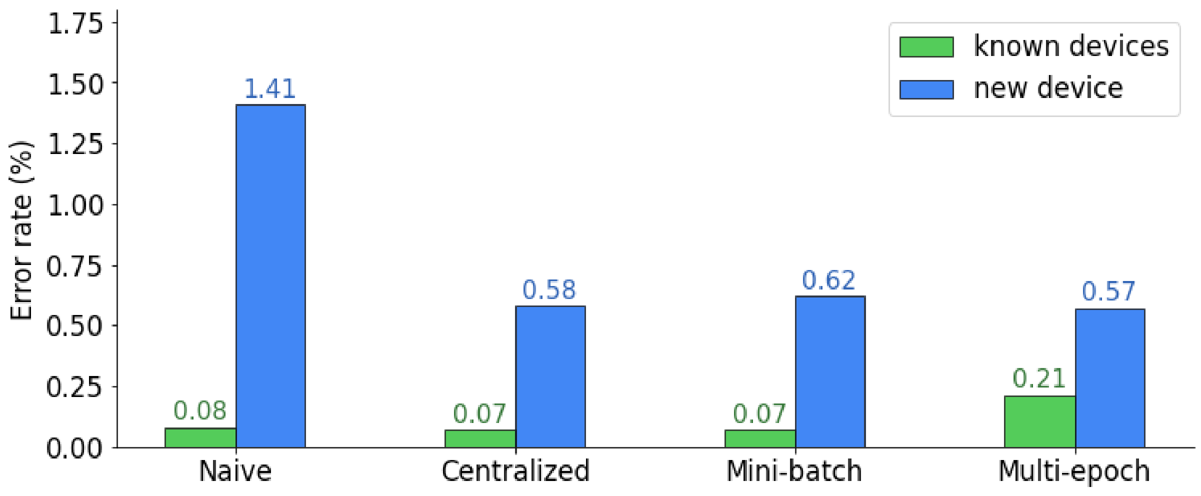
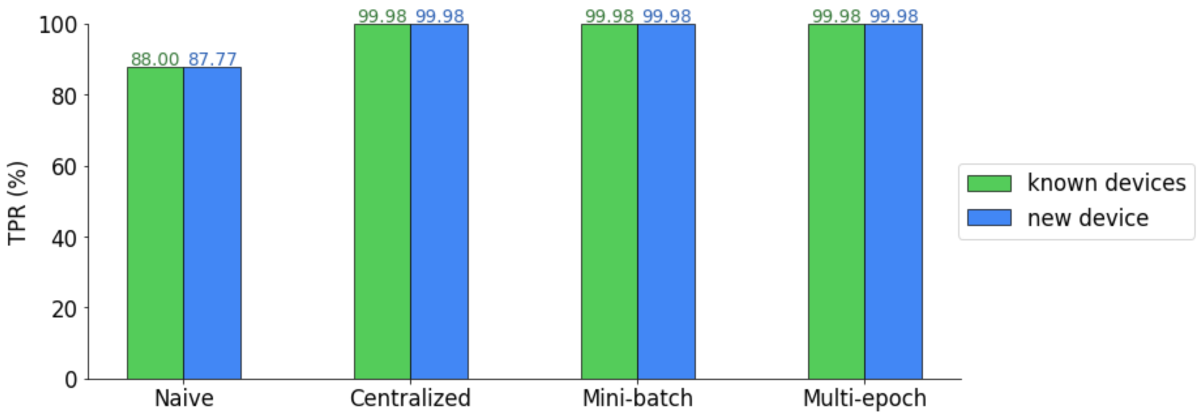


Figure 5. Shows the true positive rate of the SVM model. The true positive rate (TPR) of a federated support vector machine (SVM) model is a measure of its ability to correctly identify positive instances, specifically in the context of known and unknown devices.

Figure 5: True Positive Rate of the Federated SVM Model on Known and Unknown Devices

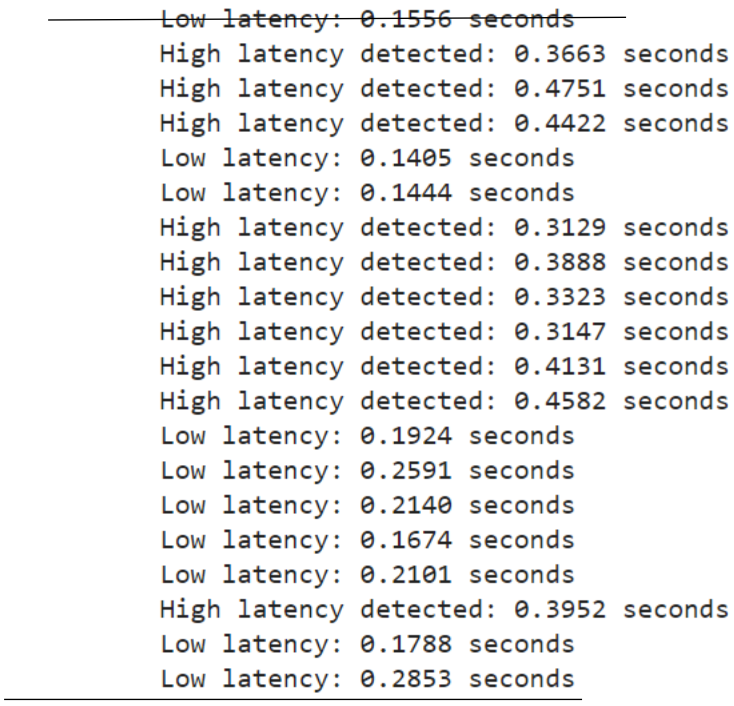


The proposed federated Support Vector Machine (SVM) model evaluated against existing systems in terms of model accuracy. The comparative analysis is presented in Table 3.

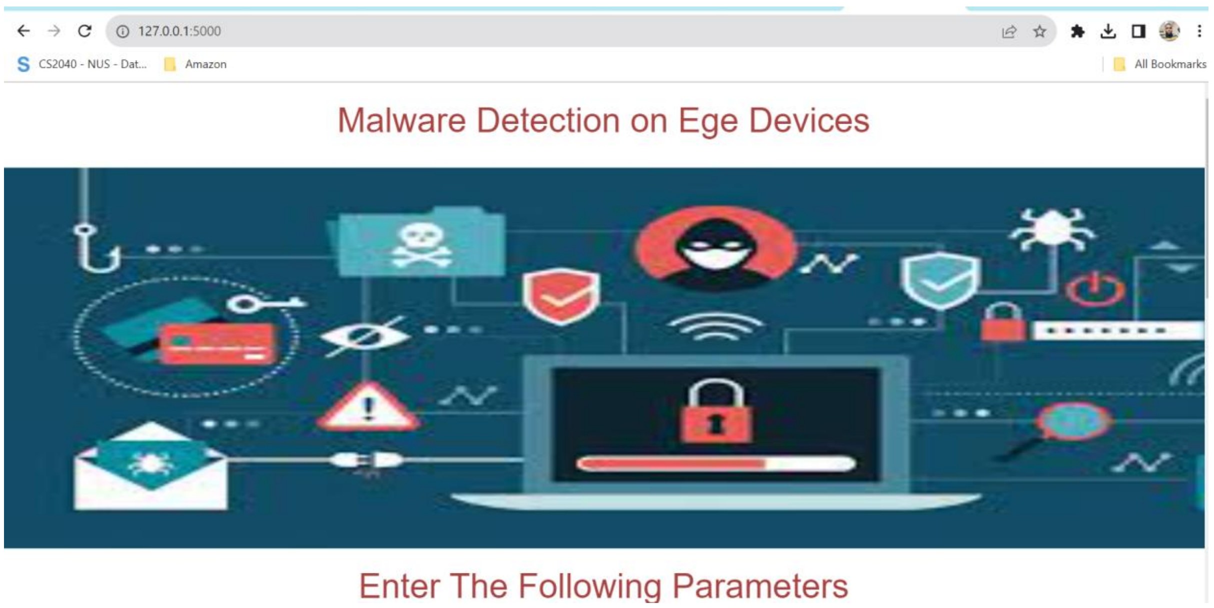
Table 4: Comparison with Other Existing System

Authors	Technique	Model Accuracy (%)
Kitsune *et al.* (2022)	Static signature-based method	94
Hsu *et al.* (2020)	Privacy-preserving Federated Learning	90
The proposed system	Federated SVM	99.98

Table 5: Comparison with the Existing and Proposed Systems in Terms of Latency



```
~~Low latency: 0.1556 seconds~~
High latency detected: 0.3663 seconds
High latency detected: 0.4751 seconds
High latency detected: 0.4422 seconds
Low latency: 0.1405 seconds
Low latency: 0.1444 seconds
High latency detected: 0.3129 seconds
High latency detected: 0.3888 seconds
High latency detected: 0.3323 seconds
High latency detected: 0.3147 seconds
High latency detected: 0.4131 seconds
High latency detected: 0.4582 seconds
Low latency: 0.1924 seconds
Low latency: 0.2591 seconds
Low latency: 0.2140 seconds
Low latency: 0.1674 seconds
Low latency: 0.2101 seconds
High latency detected: 0.3952 seconds
Low latency: 0.1788 seconds
Low latency: 0.2853 seconds
```



Malware Detection on Ege Devices

Enter The Following Parameters

Figure 6: Interface of Malware Detection on Edge Devices

5. Discussion

Table1. Shows devices that are affected by malware. From the table a "device category" refers to the classification or type of IoT device affected, such as security cameras, doorbells, thermostats, and baby monitor. A "device" represents a specific instance of the identified IoT device within its respective category. The "number" in the dataset indicates the quantity or count of instances for each device category, providing insights into the prevalence and distribution of Nbot malware infections across different types of IoT devices. Table 2 shows the ranking of the dataset features. The feature importance values for the first 15 features are provided, indicating theircontribution to the model's predictive performance. The most influential feature is "MI_dir_L0.1_weight" with an importance value of 0.100042, followed closely by "MI_dir_L1_weight" at 0.093057, and "H_L0.1_weight" at 0.092163. These features play a significant role in the model's decision-making process, suggesting that variations in these attributes have a pronounced impact on the output. Table 3 shows the accuracy of the federated SVM model for two rounds. The countplot in Figure 2 visualizes the frequency distribution of malware attacks on edge devices, with different malware types represented on the x-axis and their respective occurrence counts on the y-axis. In this specific scenario, the plot reveals that Ransomware malware has the lowest occurrence, with 5000 instances, benign with 6000 while viruses' malware exhibits the highest frequency at 10000 occurrences. Following Worms, Trojan malware is depicted with a count of 6200. This graphical representation provides a clear overview of the relative prevalence of these malware types on edge devices, with Worms being the most prominent threat, followed by Trojan, Benign and Ransomware malware being the least common. The line plot in Figure 3 shows the accuracy of the support vector machine across the federated rounds. The accuracy of the federated Support Vector Machine (SVM) model has been consistently high across three rounds of training, with an impressive accuracy rate of 99.93%. Figure 4 shows the error rate of the SVM model. The error rate of a federated Support Vector Machine (SVM) model is assessed on both known and unknown devices, employing various training strategies including Naive, Centralized, Mini Epoch, and Multiple Epoch approaches. In the Naive approach, each device independently trains its SVM model without collaboration, potentially leading to inconsistencies and suboptimal performance. Figure 5 shows the true positive rate of the SVM model. The true positive rate (TPR) of a federated support vector machine (SVM) model is a measure of its ability to correctly identify positive instances, specifically in the context of known and unknown devices. When employing a naive federated approach, the TPR on known devices is expected to be suboptimal due to limited collaboration and information sharing among devices. Table 4 shows the comparison the proposed system and other existing system. The technique used and the model accuracy. Table 5 shows the comparison with the existing system and proposed system in terms of latency. Figure 6 shows the interface of malwares detection on edge devices.

6. Conclusion

This paper successfully achieved its primary goal of developing a model for the detection of malwares on edge devices. Through the systematic pursuit of defined objectives, the adoption of SVM augments the overall accuracy and precision of malware detection, contributing to a more robust defense mechanism against evolving cyber threats. SVM, known for its ability to handle high-dimensional data and complex decision boundaries, brings a layer of sophistication to the model's architecture. The efficacy of the Random Forest Classifier in discerning intricate patterns contributes to the model's accuracy and reliability in detecting diverse types of malware. The system was developed using Python programming language, with an emphasis on model deployment via Python Flask for web-based testing and execution. By fulfilling the outlined objectives, this paper has enhanced accuracy on edge devices by effectively handling diverse datasets and overcoming resource constraints on high detection latency against evolving malware threats in edge devices. It is recommended to explore the integration of deep learning approaches. Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) have demonstrated success in capturing complex patterns and sequential behaviors, which are prevalent characteristics of sophisticated malware.

7. References

1. Baptista, I. S., Shiaeles, & N. Kolokotronis, (2019). "A novel malware detection system based on machine learning and binary visualization, 1–6.

2. Cheng, J., Wong, S. H., Yang, H., & Lu, S. (2007). Smartsiren: virus detection and alert for smartphones. In: *Proceedings of the 5th international conference on Mobile systems, applications and services.* 258–271
3. Dovom, E. M., Azmoodeh, A., Dehghantanha, A., Newton, D. E., Parizi, R. M., & Karimipour, H. (2019). Fuzzy pattern tree for edge malware detection and categorization in IoT. J. Syst. Archit., 97, 1–7.
4. Eichelberg, M., Kleber, K., & Kämmerer, M. (2020). Cybersecurity in pacs and medical imaging: an overview. Journal of Digital Imaging, 33(6), 1527-1542. https://doi.org/10.1007/s10278-020-00393-3
5. Fabricio, M. (2019). Data Science. IEEE Intelligent System, 34.
6. Elgabry, M., Nesbeth, D., & Johnson, S. (2020). A systematic review protocol for crime trends facilitated by synthetic biology. Systematic Reviews, 9(1). https://doi.org/10.1186/s13643-020-1284-1
7. Hsu, R. H., Wang, Y. C., Fan, C. I., Sun, B., Ban, T., Takahashi, T., ... & Kao, S. W. (2020). A privacy-preserving federated learning system for android malware detection based on edge computing. In *2020 15th Asia Joint Conference on Information Security (AsiaJCIS)* (128-136).
8. Iyas, Zhong, Yu, Guoming (2022). "Illuminating Malware Byte Codes with Images for Malware Classification". *IEEE Access* 438-451
9. Khan, M., & Algarni, F. (2020). A Healthcare Monitoring System for the Diagnosis of Heart Disease in the IoMT Cloud Environment Using MSSO-ANFIS. *IEEE Access* 122259–122269.
10. Kim, H. L., Nguyen, M. H., Tran, T. D., & Tran, N. D. IMIDS: (2022). An Intelligent Intrusion Detection System against Cyber Threats in IoT. *Electronics 11*, 524.
11. King, Z., Henshel, D., Flora, L., Cains, M., Hoffman, B., & Sample, C. (2018). Characterizing and measuring maliciousness for cybersecurity risk assessment. Frontiers in Psychology, 9. https://doi.org/10.3389/fpsyg.2018.00039
12. Moore, W. and Frye, S. (2019). Review of hipaa, part 2: limitations, rights, violations, and role for the imaging technologist. Journal of Nuclear Medicine Technology, 48(1), 17-23. https://doi.org/10.2967/jnmt.119.227827
13. Niharika Sharma, A., & Sahay, S. K. (2021). An effective approach for classification of advanced malware with high accuracy. arXiv:1606.06897.
14. Onyedeke, O. C., Taoufik, E., Okoronkwo, M., Ihedioha, C., & Ugwuishiwu, H. (2020). "Signature based network intrusion detection system using feature selection on android," *International Journal of Advanced Computer Science and Applications.*
15. Pedreira, V., Barros, D., & Pinto, P. (2021). A review of attacks, vulnerabilities, and defenses in industry 4.0 with new challenges on data sovereignty ahead. Sensors, 21(15), 5189. https://doi.org/10.3390/s21155189
16. Patil, D. B., & Joshi, M. (2012). "A study of past, present computer virus performance of selected security tools," Southern Economist
17. Sharma, R., Challa, & Sahay, S. (2017). Detection of Advanced Malware by Machine Learning Techniques: Proceedings of SoCTA, 333–342.
18. Singhal & Raul (2015). "Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks"
19. Syam & Vankata (2017). "Detection way where a virtual analyst was developed by using Artificial Intelligence".
20. Taylor Onate, Ezekiel & Sako (2021). A Deep Learning Based Approach for Malware Detection and Classification. *ISSN*-2347-4890.
21. Zhang, J., Li, G., Marshall, A., Hu, A., & Hanzo, L. (2020). A New Frontier for IoT Security Emerging From Three Decades of Key Generation Relying on Wireless Channels. *IEEE Access* 138406–138446.