Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 7
July 2025

Malware Attack Detection on IoT Network using Machine Learning Techniques

C.Sangami, B. Swapna

[1]Research Scholar, Department of Computer Science Engineering, Dr. M.G.R. Educational and Research Institute, Chennai, India.

[2]Associate Professor, Department of Electronics and Communication Engineering,

Dr. M.G.R. Educational and Research Institute, Chennai, India.

Abstract

The use of Internet of Things (IoT) devices is growing rapidly in parallel with the expansion of the internet. As the data capacity of IoT devices grows, they are becoming more susceptible to malware assaults. Consequently, detecting malware on IoT devices has become a crucial concern. A technique that is both effective and dependable, while also saving time, is necessary for identifying complex malware. In recent years, researchers have put forth many techniques for identifying malware. Nevertheless, achieving precise detection still poses a significant hurdle. This research introduces a Hyperparameter Fine-tuned SVM (HFSVM) machine learning method for identifying malware on IoT devices. The Randomized Search Cross-Validation (CV) approach is used to fine-tune the model parameters and optimize performance. After conducting a thorough comparison with the most advanced techniques available, we have determined that our suggested approach surpasses the performance of the current methods.

Keywords: IoT, Malware Detection, Hyperparameter Optimization, SVM, machine learning.

Introduction

The Internet of Things (IoT) refers to a network of devices equipped with sensors that have limited resources and are capable of wired or wireless connection with cloud services. In recent years, there has been a significant surge in the use of Internet of Things (IoT) devices across several sectors, including industry, healthcare, automation, education, smart homes, and smart cities. By 2020, it is anticipated that the number of linked IoT devices will exceed 50 billion, according to current forecasts [1]. Attackers are increasingly targeting IoT devices with malware due to their susceptibility to infection, which is higher compared to traditional PCs. The prevalence of outdated devices without security upgrades, insufficient attention given to security throughout the development cycle, and insecure login credentials are among the causes for this [2].

Malware is a significant issue in contemporary cybersecurity, and its identification has been a historical priority for both academic and commercial research and development. Efficient and effective detection is crucial, especially in fields like forensics or threat hunting, where extensive information storage may need thorough scanning for malware while minimising both false-positives and false-negatives [5]. Malware detection systems use various machine learning methods to detect and identify malicious software. Traditional malware detectors rely on signatures, permissions, or dynamic taint analysis to identify harmful files [6].

The results of our study demonstrate that SVM [7] may effectively classify sparse and high-dimensional data. Recent research indicates that a simpler optimisation method, such as Random Search (RS), may be enough for optimising SVM hyperparameters. Support Vector Machines (SVMs) possess a limited number of hyper-parameters that are linked to the selected kernel functions. Therefore, this optimisation presents a challenge due to its low dimensionality, which may be effectively tackled using simple strategies. These select hyper-parameters are often interdependent, resulting in the presence of an optimum region rather than a single global optimal solution [8].

Hyper-parameter optimisation, also known as the model selection issue, involves trying different combinations of hyper-parameters to create various classification or regression models. The goal is to pick the best model out of these options. Various options exist in the literature for determining an optimal combination of techniques and associated hyper-parameters. These approaches often include searching the hyper-parameter space and experimenting with various combinations [9].

2. Related Works

The Internet of Things (IoT) refers to the interconnectedness of physical and logical items via networks, allowing them to communicate and integrate with the current Internet. With the constant evolution and increasing sophistication of IoT threats, the need for robust security measures has become more crucial than ever. Malware has the ability to take advantage of weaknesses in compromised IoT systems, or it may impose special restrictions on certain IoT applications. Hence, the essential security concern that must be resolved for the IoT network is the mitigation of malware [4].

The authors of this study are Finn Gustafsson and colleagues [12]. Optimising hyperparameters is undeniably crucial for achieving a higher score. The paper concluded that it is beneficial to provide a diverse set of hyperparameters for a layered cross-validation technique. An outlier score was generated in the random forest model as a result of an incorrect range placement and/or an inappropriate amount of values for the hyperparameters. Both random forest and XGBoost exhibited a broad range of values for the hyperparameter n_estimators, indicating that the specified values were suitable.

RaniyahWazirali et al. [15] devised a novel approach using supervised and semisupervised machine learning to identify intrusions. The suggested technique focuses on enhancing the performance of the IDS (Intrusion Detection System) by optimising the k-nearest neighbours algorithm via the use of hyperparameters and cross-validation. Principal component analysis (PCA) was used to determine the most crucial region of the data. Hyperparameter tuning is the process of selecting the most optimal settings. The suggested technique aims to enhance accuracy by optimising the combination of variables such as the number of neighbours, data normalisation, distance function, and distance weight.

SharjeelRiaz and his colleagues (Riaz et al., 2018) used several machine learning and deep learning classifiers. To provide protection against very dangerous and advanced malwares, they have suggested a combination of a deep learning method called Convolutional Neural Network-Convolutional Neural Network (CNN-CNN). The suggested approach has a very favourable detection accuracy of around 99%. In addition to the full examination, they have also conducted experiments with other hybrid classifiers and other machine learning techniques. The temporal complexity of the proposed CNN-CNN is very favourable when compared to other approaches.

Rania El-Sayed and her colleagues (El-Sayed et al., 21) introduced seven machine learning algorithms for classifying malware based on images. These algorithms include Two-Layer CNN, Four-Layer CNN, VGG16, as well as various deep learning CNN classifiers. Additionally, they employed Logistic Regression, Support Vector Machine, and K-Nearest Neighbours as Ordinary Classifiers with feature selection. The algorithms are evaluated based on their complexity level and output performance. Experimental findings indicate that VGG16 produces the greatest performance with a moderate degree of complexity, while the SVM classifier obtains the second-highest accuracy with a lighter model.

3. Proposed Method

The objective of the suggested technique is to create an efficient model, known as the Hyperparameter Fine-tuned SVM (HFSVM), for identifying malware in IoT network data, as seen in Figure 1. The process starts with data acquisition, whereby an IoT network malware dataset is gathered, maybe from platforms such as Kaggle. This dataset has attributes pertaining to network traffic and labels that indicate whether the behaviour is regular or malicious. Afterwards, data preparation is performed, which includes data

cleaning to address missing values and outliers, data transformation via label encoding for categorical variables, and data scaling using RobustScaler to guarantee uniformity of features. After preprocessing, the dataset is divided into training and testing sets to enable the training and assessment of the model, respectively. The HFSVM model is constructed by combining the Support Vector Machine (SVM) method with hyperparameters that are carefully adjusted using Randomised Search CV in order to maximise performance. During the process of model training, the Hierarchical Feature Selection Support Vector Machine (HFSVM) acquires knowledge about the connection between certain features and the target variable, which indicates whether network traffic is classified as malicious or benign. Afterwards, the HFSVM model that has been trained is examined and confirmed using the testing set, and its effectiveness is assessed using metrics such as accuracy, precision, recall, and F1-score. The objective of this complete technique is to create a strong HFSVM model that can effectively identify malware in IoT network traffic, thereby improving cybersecurity in IoT settings.

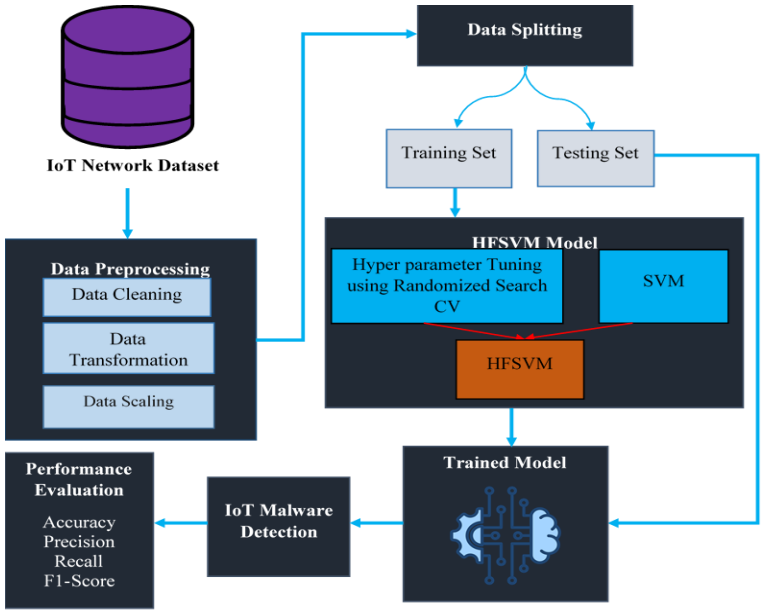


Figure1. Proposed Architecture

(i) Dataset

The tests used seventy data sets sourced from the UCI library. The dataset is created by using two sets of executables, one consisting of malicious executables and the other consisting of benign executables. Benign files are system files extracted from the system32 or programme files directory of the Windows operating system. The user downloaded malicious executables that include many types of malware, including viruses, adware, worms, Trojan horses, and others. These malwares carry out many dangerous actions like back-door downloaders, system attacks, false alerts, fake warnings, adware, and information theft [17]. Figure 2 displays the properties of the dataset together with their corresponding values, which are used for preprocessing.

(ii) Preprocessing

In the field of machine learning, having too many features may make the process of training a model more difficult and can lead to a decrease in performance, particularly when a large number of features do not have a substantial impact on the predictions. Overfitting is the term used to describe the situation when a model demonstrates good performance on the training data but performs badly on fresh, unseen data. In order to reduce overfitting, it is essential to determine the most significant characteristics and evaluate their individual impact on the model's predicted accuracy [13].

In order to address the issue of imbalanced data, we used data cleaning and data normalisation techniques during the preparation stage. Data

cleaning is the process of preparing data for analysis by eliminating inaccurate or unnecessary data, rectifying inconsistencies, and addressing missing data. Normalisation is a data preparation method used to standardise the characteristics in a dataset to a uniform scale, hence enhancing the performance and precision of machine learning algorithms. Figure 4 displays the results of the null value checking process for data cleaning, both before and after the cleaning process.

index	ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	conn_state	history	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes	label
0	1525879831.015811	8363	5	2335	8311	3	5	3	3	180	0	0	1
1	1525879831.025055	4745	5	2873	8230	3	5	3	1	60	0	0	1
2	1525879831.045045	899	5	1045	567	3	5	3	1	60	0	0	1
3	1525879832.01624	3760	5	3377	1431	3	5	3	3	180	0	0	1
4	1525879832.024985	5412	5	1443	9515	3	5	3	1	60	0	0	1
5	1525879832.044975	6103	5	2164	975	3	5	3	1	60	0	0	1
6	1525879833.016171	11280	5	198	2163	4682	5	3	3	180	0	0	0
7	1525879833.044906	2097	5	275	2038	2010	5	3	1	60	0	0	0
8	1525879834.024847	3825	5	3106	3576	3	5	3	1	60	0	0	1
9	1525879834.045086	12283	5	1387	9643	768	5	3	1	60	0	0	1

Figure 2. Data Transformation – String into numerical using Label Encoding

Figure 2 shows the Data Transformation of String into numerical using Label Encoding. Figure6 shows the Data Scaling using Robust Scaler process

(iii) Train and Test

The models, which include conventional methods such as decision trees and support vector machines as well as sophisticated neural networks, underwent thorough training and testing in a controlled setting. The extensive testing yielded useful insights into the strengths and shortcomings of each model when dealing with unbalanced data sets. Following examination, the gathered characteristics are consolidated into a dataset that is used for training and evaluating machine learning (ML) models [13]. Figure 7 displays the division of data into training and testing sets. The system utilises the HFSVM classifier, a machine learning approach that consists of two phases: training and testing. We partitioned the dataset, using a portion for training the classifier and reserving another portion for validation as test data. HFSVM provides a benchmark with a measured target value for both benign and malicious instances.

(iv) Proposed HFSVM Model

The categorisation is conducted on the traffic at the level of the IoT access gateway, rather than at the level of individual devices. This approach is quicker and requires less memory space, since it operates on aggregated data. There are two categories of traffic at the gateway level: benign and malicious. Benign traffic refers to gateway traffic that does not include any scanning packets induced by malware, whereas malicious traffic refers to gateway traffic that contains scanning packets induced by malware from one of the three kinds of malware [2]. Generating benign traffic is a very simple task since it just requires the regular functioning of devices that are not affected. Nevertheless, harmful traffic comprises both benign communication and scanning/infection packets created by malware.

The Support Vector Machine (SVM) is a kind of supervised learning model that was first introduced in 1995. It is often used for classifying datasets into binary or multiple categories. The operational principle involves treating classification issues as quadratic optimisation problems. Therefore, this method achieves the desired outcome with a reduced number of transactions, resulting in a significant performance advantage over other algorithms. Due to this attribute, it yields favourable outcomes in both small-scale and large-scale datasets [14].

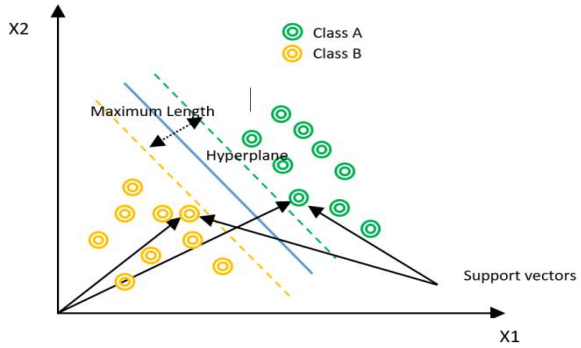
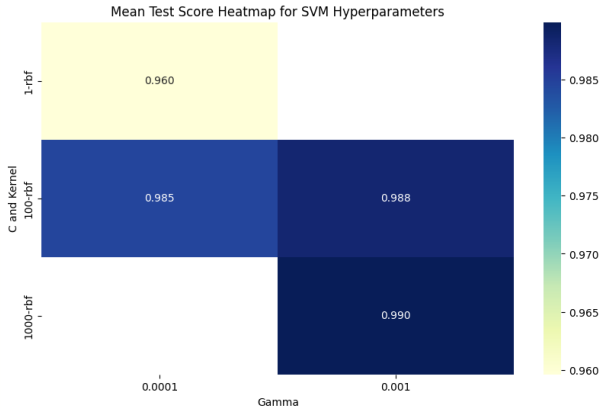


Figure 3. The optimal hyperplane that separates the positive and negative values

Support vector machines (SVM) are a kind of supervised machine learning techniques that construct a binary classification model for intricate and extremely non-linear situations. The Support Vector Machine (SVM) uses data samples to construct a hyperplane, known as the decision surface, and aims to maximise the margin around it (refer to Figure 3). During the training phase, each data sample (xi) is allocated to its corresponding class (yi), which represents the predicted value. Hence, the training set is denoted by pairs (xi ,yi), where i ranges from 1 to l, x belongs to the set of real numbers (Rn), and y belongs to the set {1, -1}. The data sample comprises the features, which are the data parameters used to characterise the activity of the data sample vector. At the conclusion of the SVM training phase, a collection of support vectors is obtained. These support vectors form the optimum hyperplane, and the corresponding weights wi are assigned to each input feature. These weights are used to forecast the value of y. The key distinction of SVM from other neural networks is in its use of margin maximisation optimisation to minimise the number of non-zero weights to a minimal amount. These relate only to the significant characteristics that provide valuable information for determining the hyperplane [3].

The HFSVM method synergistically integrates the capabilities of the SVM classifier with fine-tuned hyperparameters to augment its efficacy in identifying malware in IoT network data. First, a basic SVM classifier is created, which forms the basis for the next fine-tuning procedure. The process of optimising hyperparameters is performed using a Randomised Search Cross-Validation (RandomizedSearchCV) technique. This involves randomly sampling a preset set of hyperparameters from specified distributions [8]. This search includes tuning parameters, such as the regularisation parameter `C` and the kernel coefficient `gamma`, which may be adjusted for both linear and radial basis function (RBF) kernels. The RandomizedSearchCV method systematically explores different combinations of hyperparameters while using cross-validation to evaluate the performance of the model. Random search offers several significant benefits in comparison to grid search [10] [11]. The technique discovers the best-performing variation of the SVM model by assessing multiple parameter configurations and choosing the combination that maximises the supplied scoring measure, which in this instance is accuracy. The HFSVM model incorporates the optimised hyperparameters, which are included into the SVM framework to create a classifier specifically designed to accurately differentiate between normal and malicious IoT network activities. The HFSVM method aims to enhance the accuracy and resilience of malware detection in IoT settings via an iterative refinement process, ultimately strengthening their security.



12

Figure 4.HFSVM Model Building & Training

Figure 4 displays a heatmap that represents the average test scores achieved for various combinations of hyperparameters when adjusting the Support Vector Machine (SVM) model. Every individual cell in the heatmap corresponds to the average test score obtained from a particular combination of the regularisation parameter (C), kernel type, and kernel coefficient (gamma).

The x-axis indicates the values of the kernel coefficient (gamma), while the y-axis displays the combinations of the regularisation parameter (C) and kernel type. The colour intensity of each cell represents the mean test score, with deeper hues signifying better scores.

This visualisation offers significant insights into the influence of various combinations of hyperparameters on the performance of the SVM model. It assists in determining the most favourable parameter configurations that result in the greatest average test scores, hence aiding in the selection of hyperparameters to achieve optimum performance in detecting malware in IoT network traffic.

4. Results

To discover IoT network malware, the IoT Malware Detection dataset delivers extensive network traffic data. Network connections' source and destination IPs, ports, communication protocols, duration, and data exchange are recorded in each record. Additional fields monitor connection status, including success, failure, and missing or retransmitted bytes. The dataset differentiates local and external traffic and identifies connections as benign or malicious. These qualities aid in IoT malware detection model development.

index	label	Predicted Result
5945	1	1
17192	1	1
13996	0	0
51	1	1
8873	1	1
9622	1	1
16082	0	0
2021	1	1
10331	0	0
13841	0	0
14961	1	1
11891	1	1
2164	1	1
4723	1	1
3922	1	1
157	1	1
13066	0	0
9016	1	1
12153	1	1

Figure 5. Predicted Result for Testing Set

Figure 5 displays the DataFrame that presents the real findings along with their matching predictions given by the Hyperparameter Fine-tuned Support Vector Machine (HFSVM) model. This DataFrame presents a concise summary of the model's performance by comparing the real labels with the

predictions generated by the HFSVM model. Each row in the dataset represents an individual instance, with the "Actual" column providing the real label of the occurrence, and the "Predicted" column showing the equivalent prediction made by the model. This presentation enables a direct evaluation of the model's precision and efficiency in categorising instances of normal and malicious behaviour in the IoT network data.

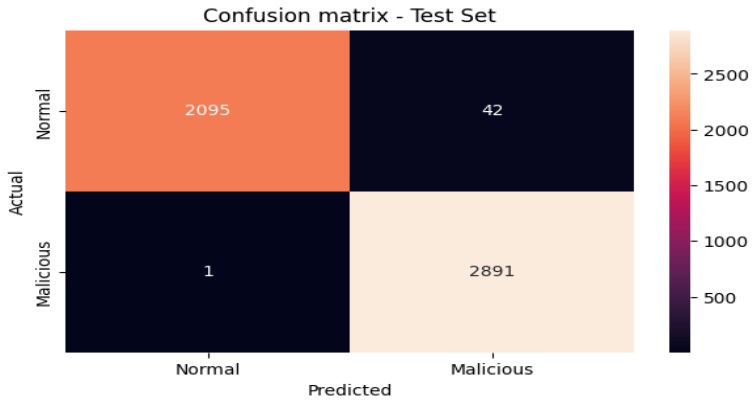


Figure 6. Confusion Matrix of the proposed method

Table 1. Confusion matrix of proposed and existing

TP	**FN**	**FP**	**TN**
1838	299	266	2626
1864	273	106	2786
1954	183	66	2826
2011	126	52	2840
2095	42	1	2891

The performance assessment of the HFSVM model demonstrates its efficacy in identifying malware in IoT network data. Figure 6 depicts the examination of the confusion matrix, which demonstrates the model's capacity to categorise events as either normal or malevolent activities. More precisely, the model successfully detects 2891 occurrences of harmful behaviour, demonstrating its resilience in identifying security risks. Furthermore, the algorithm accurately categorises 2095 instances of regular network activity, demonstrating its capacity to properly differentiate benign traffic. Nevertheless, there were 42 instances in which regular behaviour was mistakenly identified as harmful, whereas just 1 case of malevolent conduct was inaccurately categorised as normal. Although there are some misclassifications, the HFSVM model nonetheless demonstrates impressive overall performance, characterised by a high accuracy rate and a reasonably low frequency of false positives.

Table 2. Comparison of proposed and existing methods for precision, recall, fscore, specificity, accuracy

Methodology	Precision	Recall	Fscore	Specificity	Accuracy
LR	87.36	86.01	86.68	90.80	88.77
DT	94.62	87.23	90.77	96.33	92.46
KNN	96.73	91.44	94.01	97.72	95.05

SVM	97.48	94.10	95.76	98.20	96.46
HFSVM	99.95	98.03	98.98	99.97	99.14

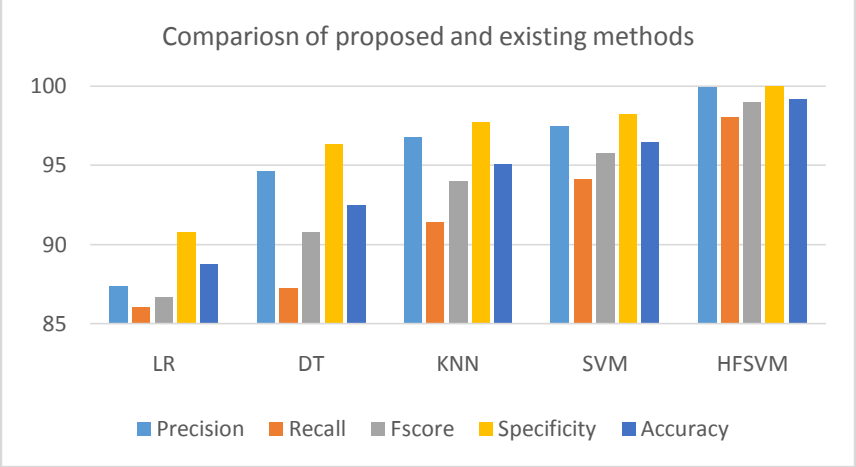


Figure 7. Accuracy, precision, recall and f1-score

The HFSVM model shown exceptional efficacy in identifying malware in IoT network data. Figure 7 displays the assessment metrics that demonstrate the effectiveness of the HFSVM algorithm in effectively distinguishing between normal and harmful activities.

Accuracy: The HFSVM model attained a precision of 0.9914, indicating that it accurately categorised 99.14% of cases in the testing set.

Precision: The HFSVM model demonstrated a precision of 0.9900, indicating a high degree of accuracy in correctly detecting real positive instances while minimising false positive detections.

The HFSVM model achieved a recall score of 0.9900, demonstrating its high capability to correctly identify 99.00% of positive occurrences and effectively reducing the number of incorrect negative classifications.

The HFSVM model earned a high F1-score of 0.9900, indicating a strong balance between accuracy and recall. This demonstrates the model's robust ability in detecting malware.

The outstanding results highlight the efficacy of the HFSVM algorithm in precisely detecting and reducing harmful behaviour in IoT network settings. The HFSVM model's strong accuracy, precision, recall, and F1-score metrics combined prove its effectiveness in enhancing cybersecurity measures and protecting IoT systems from possible attacks.

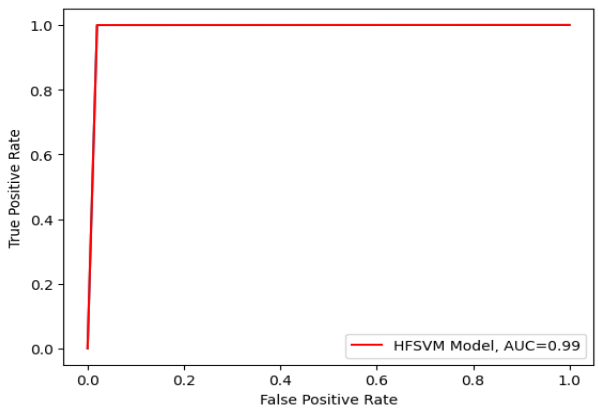


Figure 8.ROC Curve

Figure 8 displays the assessment of the HFSVM model's Receiver Operating Characteristic Area Under Curve (ROC AUC) score. This score provides more clarity on the model's ability to differentiate between normal and malicious activities in IoT network data.

The HFSVM model demonstrated a commendable ROC AUC score of 0.99, indicating its strong capability to distinguish between benign and malicious network traffic examples. A ROC AUC value nearing 1 indicates exceptional ability to distinguish between genuine positives and false positives, with very little overlap between the two rates. The high ROC AUC value further validates the effectiveness of the HFSVM algorithm in identifying and reducing cybersecurity risks in IoT systems. The HFSVM model utilises optimised hyperparameters and powerful machine learning algorithms to achieve a high degree of accuracy and reliability in detecting malicious activity. This significantly improves the security of IoT infrastructures.

5. Conclusion

Many Internet of Things (IoT) devices are connected to the Internet without receiving any security upgrades. Security is not prioritised highly in the creation of IoT devices. With the rapid expansion of IoT devices, it is crucial to protect these devices in the network from susceptible assaults, such as malware. This research introduces a Hyperparameter Fine-tuned SVM (HFSVM) machine learning method for identifying malware on IoT devices. After conducting a thorough comparison with the most advanced techniques available, we have determined that our suggested approach surpasses the performance of the current methods.

Future research might explore the detection of subtle malware via the analysis of resource information and the improvement of system accuracy. Due to the increasing prevalence of various forms of mobile malware and the emergence of new varieties, it is necessary to plan for additional research on a method that can effectively identify future malware.

References

1. Bendiab, G., Shiaeles, S., Alruban, A., &Kolokotronis, N. (2020, June). IoT malware network traffic classification using visual representation and deep learning. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)* (pp. 444-449). IEEE.
2. Kumar, A., & Lim, T. J. (2019, April). EDIMA: Early detection of IoT malware network activity using machine learning techniques. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (pp. 289-294). IEEE.
3. Ioannou, C., &Vassiliou, V. (2021). Network attack classification in IoT using support vector machines. *Journal of sensor and actuator networks*, *10*(3), 58.
4. Jamal, A., Hayat, M. F., & Nasir, M. (2022). Malware detection and classification in IoT network using ANN. *Mehran University Research Journal Of Engineering & Technology*, *41*(1), 80-91.
5. ALGorain, F. T., & Clark, J. A. (2022). Bayesian hyper-parameter optimisation for malware detection. *Electronics*, *11*(10), 1640.
6. Alqahtani, E. J., Zagrouba, R., &Almuhaideb, A. (2019, June). A survey on android malware detection techniques using machine learning algorithms. In *2019 Sixth International Conference on Software Defined Systems (SDS)* (pp. 110-117). IEEE.
7. Han, H., Lim, S., Suh, K., Park, S., Cho, S. J., & Park, M. (2020, February). Enhanced android malware detection: Ansvm-based machine learning approach. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 75-81). IEEE.
8. Mantovani, R. G., Rossi, A. L., Vanschoren, J., Bischl, B., & De Carvalho, A. C. (2015, July). Effectiveness of random search in SVM hyper-parameter tuning. In *2015 international joint conference on neural networks (IJCNN)* (pp. 1-8). Ieee.
9. Tsamardinos, I., Rakhshani, A., &Lagani, V. (2015). Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization. *International Journal on Artificial Intelligence Tools*, *24*(05), 1540023.

10. Soper, D. S. (2021). Greed is good: Rapid hyperparameter optimization and model selection using greedy k-fold cross validation. *Electronics*, *10*(16), 1973.
11. Bergstra, J., &Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2).
12. GUSTAFSSON, F. Comparing Random Forest, XGBoost and Neural Networks With Hyperparameter Optimization by Nested Cross-Validation.
13. Tan, M., Zontul, M., &Bakır, H. (2024). Malware Detection using different Machine Learning Models using Random Search and Different Features Subsets.
14. YILMAZ, E. K., & BAKIR, H. (2023). Hyperparameter tunning and feature selection methods for malware detection. *PoliteknikDergisi*, 1-1.
15. Wazirali, R. (2020). An improved intrusion detection system based on KNN hyperparameter tuning and cross-validation. *Arabian Journal for Science and Engineering*, *45*(12), 10859-10873.
16. Ranveer, S., &Hiray, S. (2015). SVM based effective malware detection system. *Int. J. Comput. Sci. Inf. Technol*, *6*(4), 3361-3365.