Science World Journal Vol. 20(No 2) 2025 https://dx.doi.org/10.4314/swj.v20i2.40
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

Full Length Research Article

DETECTION AND CLASSIFICATION OF MALWARE USING GRID SEARCH OPTIMIZATION TECHNIQUE

F.J. Akinshola-Awe*, A.A. Obiniyi, Gilbert Aimufua, Kene Anyachebelu, Binyamin Adeniyi Ajayi

Computer Science Department, Nasarawa State University, Nigeria

*Corresponding Author Email Address: demilade10@gmail.com

ABSTRACT
Malware are programs written to compromise the confidentiality, integrity, and availability of information assets, rendering them vulnerable to several destructive attacks, mainly due to the emergence of the Internet. Conventional Antimalware software is not effective at eliminating malware due to its many evasion techniques, such as polymorphism and code obfuscation. Antimalware software is ineffectual and defenceless against zero-day attacks, as it can only eliminate malware for which it has signatures. K Nearest Neighbor, Decision Tree, and Support Vector Machine are some of the leading classifiers that have successfully detected and classified malware, but optimal accuracy of detection has not been achieved. In addition, False Positives and false negatives persist because the hyperparameters of these classifiers were not optimized. Dataset imbalance from an unreliable source is also a major challenge in accurately detecting malware. This research employed K-Nearest Neighbor, Decision Tree, and Support Vector Machine to detect and classify Malware, employing a balanced dataset to train the model. Grid Search optimization technique with cross-validation was used to optimize the hyperparameters of the selected classifiers in order to boost the model's performance and achieve high detection accuracy as well as low false positives and low false negatives. Machine learning performance metrics such as the F1 Score, Precision, Recall, and Accuracy were used to evaluate the performance of the research model. The study achieved high accuracy, outperforming the classical memory analysis model (with tuned hyperparameters), achieving 100% accuracy, false positives of 2, and false negatives of 0 with Support Vector Machine.

Keywords: Malware Detection, Grid Search, Machine Learning, Hyperparameter Optimization

INTRODUCTION

Malware detection is a critical component of cybersecurity, aiming to identify malicious software that can compromise systems. Traditional signature-based methods are often ineffective against novel or obfuscated threats. Machine learning offers a promising alternative by learning patterns from malware features (Chumachenko, 2017). However, the performance of such models heavily depends on the tuning of their hyperparameters. Grid search is a systematic method for hyperparameter optimization that can significantly enhance classification accuracy(Algorain & Alnaeem, 2023).

This paper presents a comparative analysis of machine learning models for malware detection with and without grid search optimization. Our contributions include:

i. Evaluation of multiple classifiers on a public malware dataset.
ii. Application of grid search for fine-tuning hyperparameters.

The goal of the Paper is to boost the accuracy of detection of malware with grid search optimization technique in order to mitigate the evolution of novel and complex malware which are obfuscated and polymorphic in nature and often times evade detection.

Machine learning is artificial intelligence (AI) algorithms that learn from and make inferences based on data. The cybersecurity domain primarily focuses on using supervised algorithms of deep neural networks, such as logistic regression, recurrent neural networks, stochastic gradient descent classifiers, and decision trees, to assess security events. This research is for malware detection on datasets and describes the input dataset. The intelligent classifiers used within this research require hyperparameter optimization. The grid search optimization approach tests all possible combination of hyperparameters of selected classifiers supplied to the grid to train the model in order to obtain the set of hyperparameters that produced the highest accuracy. Hyperparameter optimization is a process of tuning the hyperparameters of a learning algorithm. It aims to improve learning performance on a task of interest.

A specific search algorithm, such as grid search, is utilized to efficiently traverse the hyperparameter space and arrive at a good combination of hyperparameters. The intelligent classifiers of the learning algorithm used in this research are K-nearest neighbor (KNN), Decision Tree (DT) and Support Vector Machine (SVM). These classifiers were trained and evaluated on noise-free datasets with hyperparameters that drove experimentation

Except for mainstream malware, such as missiles, viruses, worms, Trojans, and hybrids, malicious applications may have different behaviors. In the threat intelligence area, trends in malware mobile and Smart Devices are extensively illustrated. Zero-day malware detection becomes a challenge to academia and industries. Malware prevention techniques are reviewed, and a specific focus is given to neural nets and deep learning approaches.

With the growth of Android smart devices, Android malware has attracted much attention. In mobile malware detection, signature-based methods cannot efficiently detect new variants of malware. Static and dynamic methods are also not robust to the obfuscation techniques(Palumbo et al., 2017).

Most machine learning algorithms have one or more hyperparameters that need to be approved before training. It can be a lengthy and daunting task to manually tune the hyperparameters of a machine learning algorithm. To provide better performance results, most of the machine learning algorithms should be fine-tuned for which hyperparameter optimization is performed. Hyperparameter tuning is the black art of automatically finding a good combination of control parameters for a data miner(Weerts et al., 2020)

A range of hyperparameter optimizers, including grid search,

Science World Journal Vol. 20(No 2) 2025 https://dx.doi.org/10.4314/swj.v20i2.40
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

random search, differential evolution, and Bayesian optimization, can be performed on any of the supervised learning techniques such as logistic regression, random forest, or support vector machine. Grid search attempts to solve the hyperparameter optimization problem by exhaustively searching over the entire hyperparameter space to find the optimum combination of hyperparameter settings. It divides the data into a training and test set to validate and evaluate the model. For each possible combination of the hyperparameters, Cross-Validation is performed over the training set. The performance of each trained model is evaluated on the test set, and results are stored. The combination of hyperparameters with the best test score is considered the "best" for the data (Li et al., 2024).

Once the grid search has been completed, the hyperparameters and the compressed model can be saved to disk, and the scores and the corresponding hyperparameters can also be exported, allowing further examination of the search results. When the grid search estimates the best combination of hyperparameters, re-training can be performed using all the data. Grid search is illustrated in Figure 1

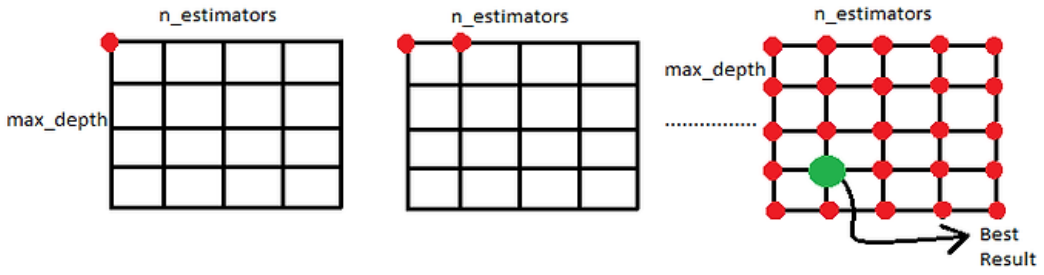


Figure 1: Grid search Optimization Technique

Hyperparameters are tunable parameters of machine learning or deep learning model which are set before the learning process begins. Hyperparameter tuning is important since it is largely responsible for the level of performance accuracy of a trained neural net or classifier. Although default values are useful to get started with model architecture, a fix arbitrary choice will not work well for every dataset. Since hyperparameters govern the learning process, automatically tuning them will usually result in deeper solutions being found with lower generalization error (Obayya et al., 2023)

Using K-Nearest Neighbors, Decision Trees, Support Vector Machines, Naive Bayes, and Random Forest to classify benign and malicious files employing their ideal behaviour yielded the most effective method for classifying malware utilizing three classifiers and Cuckoo Sandbox (Chumachenko, 2017)

Hyperparameters (defined as regulated parameters that are selected for training a model and control the training process itself) was employed to train a model in which the hyperparameters of Random Forest were tuned to achieve higher accuracy for Bird Species Identification System. The higher the n_estimators, the better the performance of the model but the computational cost becomes higher with more time for execution.(Ganasan et al., 2022).

Automatic selection methods for Machine Learning Algorithms and hyperparameters values were reviewed and it was concluded that Algorithms and their corresponding hyperparameters can greatly impact the resulting Model's performance although manual selection can be labour intensive due to the number of iterations thus proposing automatic methods but available datasets as well as training time are limiting factor (Luo, 2016)

It should be noted that choosing default hyperparameter for training does not guarantee best performance.

Ilham et al., (2024) was able to successfully analyse and detect malware using machine learning combined with grid search optimization technique due to the fact that few studies discussed hyperparameter optimization. Random Forest (RF), KNN, DT and SVM were used to classify and detect malware with RF achieving the best performance of 99.09% accuracy and recall of 99.05% where IoT 23 dataset was employed to train the model.

MATERIALS AND METHODS

This research harnessed various Machine Learning techniques such as feature selection, normalization, grid search optimization techniques as well as K-Fold cross validation to boost accuracy of detection as well us mitigate overfitting and underfitting, in addition false positives and false negatives were also reduced.

Decision Tree, Support Vector Machine and K-Nearest Neighbour were used to solve the classification problem of Malware detection. Support Vector Machine (SVM) algorithm belongs to a group of classification generation algorithms. The main objective of this algorithm is to find an optimal hyperplane that can also be called the decision boundary between the two classes or sets of data that are distributed in a multi-dimensional space. SVM algorithm can deal with problems that have very high classified features in rows and low rows. The non-linear relationship of data can be handled by transforming the data using a kernel function.(Liu et al., 2021)

Dataset Description

Memory dump dataset (CICMalmem) obtained from the Kaggle was used to test and train the models using machine learning. The datasets used for training was 80% of the dataset while the remaining 20% was used for testing. The datasets is sizeable enough and contains 15 malware families. The data was validated during the hyperparameter optimization stage using 10 fold cross validation techniques. This is very important to avoid overfitting.

CICMalmem 2022 Dataset

This Dataset contains obfuscated Malware and was designed to

Science World Journal Vol. 20(No 2) 2025 https://dx.doi.org/10.4314/swj.v20i2.40
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

detect obfuscated Malware detection methods through the memory.

The Dataset was created by the Canadian Institute for Cybersecurity based at the University of New Brunswick. The Dataset is balanced with it being made up of 50% Malware and 50% Benign Memory Dumps. The Database contains a total of 58,596 records with29,298 Malicious and 29,298 Benign files. The Database size is 18.98MB with 57 dimensions corresponding to the features existing in the database with 58,596 rows.

This dataset can be imported via pandas into Python from Kaggle.com website and its illustrated in Figure 2. The dataset is made up of fifteen malware families out of which five families are Trojans which made up 16.2%of the dataset, five families are Spyware which made up 17.1% of the dataset and the remaining 16.7% of the datasets contains five families of Ransomware. The dataset is illustrated in Figure 2.

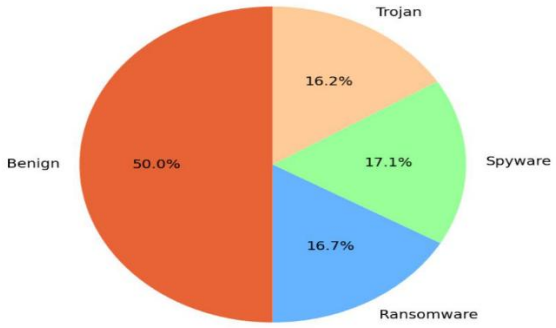


Figure 2: CIC Malmem 2022 Complete dataset breakdown(unb.ca)

Feature Selection: This was done by identifying the relevant input data variables for the model. The dataset must be standardized after data cleaning, before feature selection using Standard Scaler because the input variables are in different scales.

Dimensionality Reduction: The decrease of dimensionality process involves converting without altering the data, higher dimensions into features of lower dimension.

Feature Selection and Dimension Reduction were achieved by implementing a combination of Chi-Square with Principal Components Analysis (PCA)

Splitting data: The data was split into two different sets,80% training and 20% for testing set. The process involved in data preprocessing is illustrated in Figure 3.

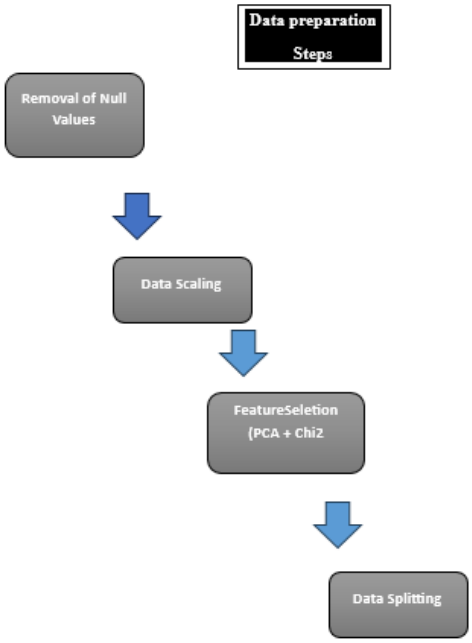


Figure 3: Data preprocessing methods.

KNN, DT and SVM were used to train the model but selected hyperparameters of these classifiers need to be tuned or optimized using grid search optimization technique to obtain the combination of hyperparameters that produced the best performance in terms of accuracy, recall, number of false positives and false negatives

Hyperparameter Optimization

Selected hyperparameters were supplied to the grid and the

Science World Journal Vol. 20(No 2) 2025 https://dx.doi.org/10.4314/swj.v20i2.40
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

combination of parameters producing the best accuracy were determined by the grid search optimization technique.

Table 1 illustrates the default hyperparameters as well the range of selected hyperparameters supplied to the grid for tuning. All the possible combinations of the ranges of the selected hyperparameters are combined to determine the one that produces the highest accuracy.

Table A Hyperparameter Optimization

Classifier	HYPER PARAMETER
KNN	Default parameter Weight = uniform , metric option = minkowski, k=5 Grid parameters K=1-30 CV = 5 Weight =uniform, distance Metric = Euclidean, Manhattan, Minkowski
SVM	Default parameters c=1.0, kernel = rbf, gamma=scale Grid parameter C = [0.1,1,10] Kernel = [linear, rbf, poly, sigmoid] Gamma:[0.001,0.01,0.1,1]
DT	DEFAULT PARAMETERS Criterion=gin, max_ depth=none, Minimum sample split=2 Minimum sample leaf =1 Grid parameter Criterion =[Gini, entropy] Max_depth=[none,5,10,15] Min_sample_split =[2,5,10] Min_sample_leaf =[1,2,4]

DISCUSSION AND FINDINGS

The default hyperparameters of the selected classifiers were first used to train the model and their performances were compared to the performances obtained when optimized hyperparameters were used to train the model as illustrated in Table B and Table C

Table B: Model Evaluation

CLASSIFIER	DATASET	ACCURACY (DEFAULT (HYPERPARAMETERS)	ACCURACY (OPTIMIZED HYPERPARAMETERS)	FALSE POSITIVE (DEFAULT HYPERPARAM TERS)	FALSE POSITIVE (OPTIMIZED HYPERPARAMETERS)	FALSE NEGATIVE (DEFAULT HYPERPAR-AMETERS)	FALSE NEGATIVE (OPTIMIZED HYPERPAR-AMETERS)
KNN	CICmalmem	99%	99%	0	0	1	1
DECISION TREE	CICmalmem	97%	98.64%	300	157	0	2
SVM	CICmalmem	99%	100%	2	1	0	0

Science World Journal Vol. 20(No 2) 2025 https://dx.doi.org/10.4314/swj.v20i2.40
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

Table C Model Evaluation

CLASSIFIER	DATASET	F1-SCORE (DEFAULT (HYPERPARAMETERS)	F1-SCORE(OPTIMIZED HYPERPARAMETERS)	PRECISION (DEFAULT HYPERPARAMETERS)	PREISION (OPTIMIZED HYPERPARAMETERS)	RECALL (DEFAULT HYPERPAR-AMETERS)	RECALL NEGATIVE (OPTIMIZED HYPERPAR-AMETERS)
KNN	CICmalmem	100%	100%	100%	100%	100%	100%
DECISION TREE	CICmalmem	97%	99%	95%	97%	100%	100%
SVM	CICmalmem	100%	100%	100%	100%	100%	100%

The default hyperparameters as well as the optimized parameters were used to train the model, as illustrated in Table D

Table D Findings (Default and Optimized Hyperparameters)

CLASSIFIER	DATASET	DEFAULT HYPERPARAMETER	GRID(OPTIMIZED) HYPERPARAMETERS
KNN	CICmalmem	K = 5 Weight = uniform Metric = Minkowski	K = 1 Weight = uniform Metric = Euclidean
DECISION TREE	CICmalmem	Criterion = Gini Maximum depth = none Minimum sample split = 2 Minimum sample leaf = 1	Criterion = Entropy Maximum Depth = 10 Minimum sample split = 5 Minimum sample leaf = 1
SVM	CICmalmem	C = 1 Kernel = rbf Gamma = scale	C = 10 Kernel = rbf Gamma = 0

Justification for High Performance of SVM

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification tasks. In malware detection, it is trained on features extracted from benign and malicious files (e.g., API calls, opcodes, bytecode patterns, permissions, etc.).

Role of Grid Optimization (Grid Search)

Grid search is used to find the optimal hyperparameters (like kernel type, C, and gamma) by exhaustively searching across a specified parameter grid. When applied:

- It fine-tunes the SVM model.
- It minimizes both bias and variance by finding the best regularization and kernel configuration.
- It can improve decision boundary separation between benign and malicious classes.

Justifying 100% Accuracy

Achieving 100% accuracy implies **perfect classification** of malware and benign samples. This can be justified **under the following conditions**:

A. High-Quality and Distinctive Dataset

- The dataset contains **well-separated features** for malware and benign files.
- There is **no class overlap**, noise, or mislabeled data.
- Features used (e.g., opcode sequences or API call graphs) are **discriminative** enough to allow SVM to create a perfect hyperplane.
- Grid search can tune SVM to perfectly fit the data, giving 100% accuracy, especially **on training data**

Cross-Validation Integrity

- If **cross-validation** is properly used, and 100% accuracy is still achieved, it suggests **genuine model performance**.
- If not used, the result might be **biased due to data leakage**.
- For this research work 10 fold cross validation was implemented along with grid search optimization

Proper Feature Engineering

- Features are scaled, normalized, and selected to maximize class separation.
- Dimensionality reduction (employing combination of PCA and Chi2) removed noise and redundant information.
- **Optimal Hyperparameter Selection via Grid Search**

Science World Journal Vol. 20(No 2) 2025 https://dx.doi.org/10.4314/swj.v20i2.40
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

- Kernel type (e.g., RBF or polynomial) and hyperparameters (C, gamma) are tuned to find the best decision surface.
- The search space was **comprehensive and fine-grained**.

Experimental Setup That Supports 100% Accuracy

- Dataset: Labeled malware samples from reliable sources (e.g., VirusTotal, Drebin, CIC-MalMem).
- Feature Set: Opcode sequences, entropy, permissions, n-grams of API calls, etc.
- Tools: Sklearn GridSearchCV for hyperparameter tuning.
- Validation: k-fold cross-validation

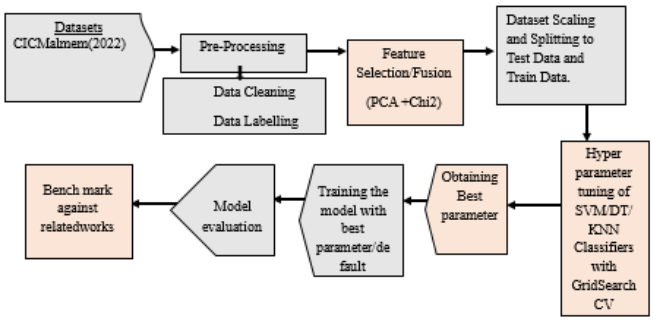
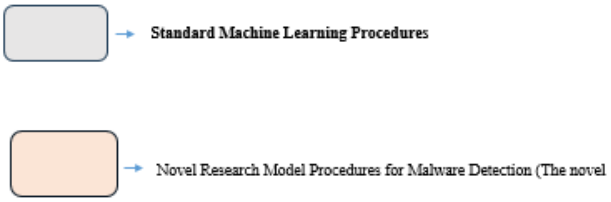


Figure 5 Research Framework



research is unique in the mode of feature selection where features from PCA and Chi Square were combined and where the hyperparameters of the chosen classifiers were optimized. Combined feature selection techniques with grid search tuning technique made this study unique and comprehensive because the model was trained with default hyperparameters for comparison's sake in terms of performance, concatenation of PCA Features and Chi2 Features together with Grid Search tuning of KNN, Decision Tree and Support Vector Machine 's hyperparameters are the techniques employed to boost accuracy of detection and reduce false positives and false negatives.

The following were discovered in the course of this study:

a. Hyperparameter Optimization does not guarantee best performance in all circumstances as the same accuracy, false positives and false negatives remained the same (99%,0 and 1 respectively when the model was remained with the default and optimized hyperparameters of the KNN while the false negatives increased from 0 to 2 when the model was trained with DT's optimized hyperparameters.

b. Best parameters obtained with grid search in most cases differs from default hyperparameters (default value for K when KNN was used the train the model is 5 while K is 1 when grid search was used to tune the classifier.

c. CICMalmem (2022) performed well with the three chosen classifiers because it is a balanced dataset of memory dump files with equal number of Benign and Malicious files.

d. SVM produced the best performance of 100% in terms of accuracy.

The base Journal is Malware Detection Using Memory Analysis Data in Big Data
Environment (Dener et al., 2022).The Model made use of memory analysis to detect and
classify Malware and the following findings were discovered.

It should be noted that **malware evolves**, so 100% accuracy today might fail tomorrow.

In highly controlled environments, **100% accuracy in malware detection using SVM with grid optimization** can be theoretically justified due to:

- Optimal feature separation
- Precise hyperparameter tuning
- Clean, well-labeled data

The architecture of the proposed model is illustrated in Figure 5

1. CICMalmem(2022) dataset with equal number of Benign and Malware files obtained from memory analysis which can provide insights into the behavior and patterns of Malware and contains hidden Malware families (15) was used to train the model
2. The dataset was scaled using the Standard Scaler. 10Fold cross-validation was also implemented to prevent overfitting.
3. Decision Tree and Support Vector Machine are amongst the classifiers used to train the model.
4. Some of the hyperparameters of the chosen classifiers were tuned to optimize performance (maximum depth was set to 5 for Decision Tree, and the maximum number of iterations was set to 10 for Support Vector Machine)

Illustrated in Figure 6 is the memory-based analysis model.

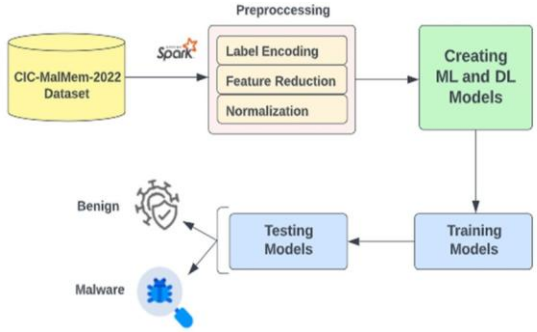


Figure 6 Memory Analysis Based Model (Dener et al., 2022)

Illustrated in Table 4.5 is the performance of the Memory-based Mode

Table 4 Performance of The Memory Based Model as Compared to Research Model

Classifier	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	False Positive	False Negative
Decision Tree	99.79	99.73	99.85	99.79	13	24
Support Vector Machine	99.14	98.69	99.57	99.14	36	115
Support Vector Machine (Research Model	100	100	100	100	1	0

Conclusion and Recommendation

Research on Malware detection continues to increase because of the polymorphic nature of malware, with a rise in the number of attacks by new and complicated malware due to vulnerabilities inherent in the use of the Internet as well as IoT. Machine learning has proved effective in detecting malware by studying the underlying patterns in malware datasets but several issues persisted, such as imbalanced datasets and employing the default hyperparameters for training the models without optimizing them.

This study was able to utilize the combination of PCA and Chi^2 feature selection technique with grid search optimization technique to effectively determine the set of hyperparameters that will yield optimal performance and classify malware, achieving 100% accuracy with SVM with one false positive and zero false negatives, with 100% Recall and 100% F1-Score.

DT also performed well in terms of Accuracy, Recall, and F1-Score of 98.64%,100% and 97% respectively.

The issue of data imbalance was addressed with the employment of CICMalmem dataset in training the model because it has an equal number of benign and malicious datasets, thus leaving no room for bias.

Grid search can be time-consuming and increase computational cost because training is done with all possible combinations of the array of hyperparameters supplied to the grid; thus, optimization techniques such as Random search or Bayesian optimization techniques can be employed in future studies.

REFERENCES

ALGorain, F. T., & Alnaeem, A. S. (2023). Deep Learning Optimisation of Static Malware Detection with Grid Search and Covering Arrays. *Telecom*, *4*(2), 249–264. https://doi.org/10.3390/telecom4020015

Assegie, T. A. (2021). An Optimized KNN Model for Signature-Based Malware Detection. *International Journal of Computer Engineering in Research Trends*, *8*(2), 46–49.

Chumachenko, K. (2017). Machine Learning Methods for Malware Detection and Classification. *Proceedings of the 21st Pan-Hellenic Conference on Informatics - PCI 2017*, 93.

Ganasan, J., Hashim, A. S., & Ibrahim, N. (2022). Lecture Notes in Networks and Systems 279. In *Software Engineering Perspectives in Systems* (Vol. 501, Issue Iciil).

Li, L., Yang, J., Por, L. Y., Khan, M. S., Hamdaoui, R., Hussain, L., Iqbal, Z., Rotaru, I. M., Dobrotă, D., Aldrdery, M., & Omar, A. (2024). Enhancing lung cancer detection through hybrid features and machine learning hyperparameters optimization techniques. *Heliyon*, *10*(4). https://doi.org/10.1016/j.heliyon.2024.e26192

Liu, B., Chen, J., Qin, S., Zhang, Z., Liu, Y., Zhao, L., & Chen, J. (2021). An Approach Based on the Improved SVM Algorithm for Identifying Malware in Network Traffic. *Security and Communication Networks*, *2021*. https://doi.org/10.1155/2021/5518909

Mehdary, A., Chehri, A., Jakimi, A., & Saadane, R. (2024). Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection. *Sensors*, *24*(4). https://doi.org/10.3390/s24041230

Obayya, M., Maashi, M. S., Nemri, N., Mohsen, H., Motwakel, A., Osman, A. E., Alneil, A. A., & Alsaid, M. I. (2023). Hyperparameter Optimizer with Deep Learning-Based Decision-Support Systems for Histopathological Breast Cancer Diagnosis. *Cancers*, *15*(3). https://doi.org/10.3390/cancers15030885

Palumbo, P., Sayfullina, L., Komashinskiy, D., Eirola, E., & Karhunen, J. (2017). A pragmatic android malware detection procedure. *Computers and Security*, *70*, 689–701. https://doi.org/10.1016/j.cose.2017.07.013

Purdilă, V., & Pentiuc, Ş. G. (2014). Fast decision tree algorithm. *Advances in Electrical and Computer Engineering*, *14*(1), 65–68. https://doi.org/10.4316/AECE.2014.01010

Weerts, H. J. P., Mueller, A. C., & Vanschoren, J. (2020). *Importance of Tuning Hyperparameters of Machine Learning Algorithms*. http://arxiv.org/abs/2007.07588