Science, Engineering and Technology Vol. 5, No. 1, pp. 203-212
www.setjournal.com *https://doi.org/10.54327/set2025/v5.i1.167*

Malware Detection Using a Random Forest Method Trained on a Balanced Synthetic Dataset

Neo Onica Matsobane[1], Sello Mokwena[1]

[1]*Department of Computer Sciences, Faculty of Science and Agriculture, University of Limpopo, Polokwane, South Africa*.

Abstract

The accuracy of malware detection is closely related to the available datasets, which are often small and imbalanced. To overcome these challenges, this study proposed a new method that creates synthetic malware data and increases the size and balance by generating several data sets with a flow-based model. Subsequently, a random forest classifier is fitted on this augmented dataset. This study aimed to analyze the generation of synthetic data based on flow-based models and the impact of synthetic data generation on the performance of a random forest for malware detection. A flow-based model was used to generate a balanced synthetic dataset based on the CICMalDroid2020 dataset. The generated data was used for feature selection and engineering to optimize the Random Forest model. The experimental results demonstrate the effectiveness of the proposed approach. The flow-based model generated an additional 13,402 samples, massively increasing the dataset size, even though the original dataset had only 11,598 data entries. After training on the synthetic augmented dataset, the Random Forest model achieved better performance compared to the original dataset evaluation with metrics precision (93%), recall (100%), balanced precision (96%), and the F1 score (91%). The results show that flow-based model-generated synthetic data can significantly enhance malware detection capabilities.

Keywords*: malware detection, accuracy, random forest, flow-based model, balanced dataset, synthetic dataset.*

1. Introduction

In recent years, the spread of malware, known as malicious software, has posed significant threats to the security and privacy of computer systems and networks [1]. Malware is software designed to cause harm to the computer, network, or data. Types of malwares include viruses, worms, trojans, ransomware, spyware, adware, and more. Each has its method of infiltrating and causing damage. Malware can penetrate systems, exfiltrate sensitive information, disrupt operations, and even enable unauthorized access to resources [2], [3], [4], [5].

In the early days (1970s-1980s), viruses were primitive and mainly created to show messages or disrupt system operations. Detection was largely manual, incorporating inspection of code and system behavior. Once viruses became prevalent in the late 1980s and 1990s, antivirus software began to play. These programs noted patterns of known malware signatures in databases. They also proposed a heuristic analysis to detect suspicious behaviors that could be considered malware [6]. After the internet became widespread in the 1990s-2000s, new threats, such as worms and Trojans, could spread quickly. Antivirus companies evolved and created behavioral analysis, cloud-based scanning, and machine learning. In the 2010s and later, creators of various malware developed polymorphic and metamorphic malware capable of changing their signatures with each iteration to avoid detection. The increase in ransomware and advanced persistent threat (APT) attacks was one of the significant risk factors. Using AI and machine learning methods, much of the data is analyzed to detect patterns

Corresponding author: *Neo Onica Matsobane (neonicano@gmail.com)*
Received: *2 August 2024*; Revised: *27 February 2025*; Accepted: *11 March 2025*; Published: *28 March 2025*

and anomalies, which has been extremely helpful in countering these threats. Suspicious files are also isolated and analyzed through behavioral analysis and sandboxing.

Traditional malware detection relies on signature-based systems that analyze known malware, which struggles against evolving threats and unbalanced datasets. Researchers use synthetic data to create balanced datasets that enhance detection accuracy [7], [8]. Key advantages of synthetic data include balanced datasets, which provide an equal representation of benign and malicious samples, addressing class imbalance issues in real-world datasets, and leading to improved detection of malicious behavior [9], [10]. Diverse Malware Samples facilitate the generation of varied malware samples covering a broader range of malicious behaviors, enhancing robustness against new threats. With control over parameters, researchers can manipulate features of generated samples, enabling targeted studies on malware characteristics and improving detection methods [11].

This research investigates the use of synthetic datasets with random forest models for malware detection. Random forests, which include multiple classification trees, learn from randomly selected data samples. The study aimed to detect malware using the random forest technique. Additionally, it addresses the challenges posed by small-scale and imbalanced data sets. These models efficiently learn complex data distributions and are alternatives to traditional generative models. Feature selection and engineering are applied to optimize the random forest model further.

Feature selection solves overfitting and the curse of dimensionality in a data set by reducing the number of features in a model and optimizing the model performance. Feature engineering is the generation of new features from existing features. We evaluated the performance of random forest techniques in detecting malware using performance metrics, namely accuracy, precision, recall, F1 score, balanced accuracy, and geometric means. Our research aims to contribute to developing more robust and adaptable malware detection systems, ultimately improving the security and protection of computer systems and networks by using synthetic data. The objectives of this study are as follows.

i. Using the flow-base model to generate a synthetic malware dataset.

ii. Using proposed performance metrics, evaluate the performance of random forest techniques in detecting malware in synthetic datasets.

The remainder of this paper is as follows. Section 2 provides a comprehensive examination of the relevant literature. Section 3 briefly describes the methodology for generating the results and collecting data. Section 4 presents the analysis of the results. Section 5 provides a conclusion and delves into future work, acknowledgment, competing interest statement, data and materials accessibility, and finally, references used in the study.

2. Literature Review

Garcia and Muga [6] classified the malware families using malware images. It is also worth noting that the data set used to evaluate the method was the Mallimg Dataset, containing 9,342 malware samples in 25 families. These data differ from our study, as we proposed using a new Android dataset called CICMalDroid2020. Although the Malimg dataset provided diverse malware samples, it may not represent the entire malware landscape. Also, the scale of the dataset limits the generalizability of the findings. Using a different CICMalDroid2020 dataset in the study introduces the possibility of variations in malware characteristics and difficulty levels, which could impact model performance. Both this study and ours employed a random forest model. As evaluated in the study, the random forest model demonstrated a 95.26% classification accuracy in the malware data set. Stratified sampling was used during training set creation to counteract the imbalance of the dataset, thereby reducing the risk of overfitting and undergeneralization.

In contrast, our research proposed a different approach, namely SMOTE, to address the data imbalance problem. The Malimg dataset was tested using a 10-fold cross-validation. This study used performance metrics such as precision, precision, recall, false alarm rate, and F1 score to evaluate the random forest model.

Sawadogo et al. [12] assessed the impact of unbalanced data sets on the effectiveness of machine learning models designed to identify malicious applications. The primary objective of their study was to investigate the effects of imbalanced datasets on algorithms. Additionally, they provided evaluation

metrics that are most appropriate for imbalanced data sets. A balancing metric and commonly used metrics were used to determine the performance of eleven classification algorithms. The results indicated that performance evaluation metrics, such as accuracy, precision, and recall, which are widely used in the literature, do not perform well when faced with unbalanced data sets.

In contrast, Balanced accuracy and Geometric mean metrics are more appropriate. The results of this study were obtained by evaluating precision, precision, F1 score, recall, balanced precision, Matthew's correlation coefficient, Geometric mean, Fowlkes marshmallows, and F1 score. However, the size and diversity of the data set can also affect the generalizability of the findings. A more extensive and more diverse data set would provide a more comprehensive understanding of the impact of unbalanced data sets. Therefore, we used these results in this study to check whether these metrics are best suited to evaluate a model feed with the synthetic data set. From the metrics used, we only used precision, recall, precision, F1 score, balanced accuracy, and geometric metrics.

Rosmansyah and Dabarsyah [13] proposed a new approach to identify malware on Android mobile devices developed using API classes. This study used machine learning to classify programs as benign or malicious. Additionally, the machine learning classification precision rate was contrasted. Based on 51 API packages from 16 API classes, the study classified benign and malicious software using the Random Forest, J48, and Support Vector Machine methods. 412 sample Android applications are used for the implementation, along with 51 packages from 16 API classes, 205 benign applications, and 207 malicious applications. Interpreting the evaluation metrics should be considered in conjunction with the specific characteristics of the data set and the goals of the malware detection task. The malware landscape is constantly evolving, and new threats emerge regularly. The findings of the study may not apply to future malware variants. The data set used in the previous study differs from ours. We used the more recent CICMalDroid2020 dataset, which contains Android data. Additionally, while the study used cross-validation and percentage split for testing, our research evaluated classifier performance using precision, F1 score, recall, balanced accuracy, and geometric mean. According to research [13], the percentage of packages that used benign and malware samples revealed that Telephony Manager and Connectivity Manager are the most often used by malware applications. The SVM, J48, and Random Forest algorithms are all machine learning algorithms that can be used to improve malware detection. For cross-validation tests using the random forest algorithm and percentage split tests using the SVM algorithm, the precision rates were 92.4% and 91.4%, respectively. Similarly to our research, [13] the researchers used the Random Forest technique to classify malware. Their findings align with ours, demonstrating the superior precision of the Random Forest algorithm in this context. In the cross-validation test, 92.4% of the samples used the Tree Random Forest algorithm, and 91.4% used the SVM algorithm, so the average output for all samples (without categorization) was 91.9%. Researchers indicated that Random Forest gives the best results in malware detection research. Lastly, these results support our motivation for using the random forest technique as a malware detection classifier in our study.

J. Singh [14] supports previous researchers [13] by providing an analysis that shows that the random forest technique provides the best precision in the malware detection data set. As a result, this analysis supports our motivation to use a random forest technique to detect malware on a synthetic dataset. Using 57 attributes, [15] developed a machine-learning algorithm to differentiate between benign and malicious Windows PE files. Unlike our study, they used the Brazilian Malware dataset of 100,000 samples with 57 labels. The Random Forest model achieved an impressive 99.7% accuracy, outperforming existing systems. This demonstrates the potential of Random Forest to identify malicious files effectively. An unbalanced data set is one of the biggest obstacles to malware detection, according to [16]. The data set for the study was preprocessed and balanced to improve malware identification accuracy and minimize false positives and negatives. Although the study employed static analysis to extract features, our research utilized normalization for feature extraction. Both studies preprocessed the feature set using ranking techniques to eliminate ineffective features.

Our research introduces a combined approach to address the challenges of imbalanced datasets. Random Forest and a balanced synthetic dataset generated through flow-based modeling. This aims to improve the accuracy and generalizability of malware detection model models. Table 1, convey an overview of the literature review.

Table 1. Literature review overview.

Article	Dataset	Methodology	Findings
[6]	Mallimg dataset	Stratified sampling was utilized during training set creation to counter the imbalance of the dataset.	Both studies used a random forest model and achieved 95.26% classification accuracy. The authors applied stratified sampling to mitigate dataset imbalance, whereas our research employs SMOTE. Mallimg used 10-fold cross-validation, and we evaluated our model with metrics like precision, recall, false alarm rate, and F1 score.
[12]	CICMalDroid 2020 dataset	The authors examined the effects of imbalanced datasets on Android malware detection, contrasting machine learning models through feature extraction, imbalance handling techniques (e.g., SMOTE, class weighting), and evaluation metrics such as precision, recall, and AUC-ROC.	Examined how imbalanced datasets affect machine learning models for identifying malicious applications. They aimed to evaluate the impact of these datasets on algorithms and identified suitable evaluation metrics. Their findings revealed that standard metrics like accuracy, precision, and recall perform poorly with unbalanced data, while balanced accuracy and geometric mean are more effective. The study assessed various metrics but noted that the size and diversity could influence the generalisability of results. Consequently, they used these metrics to evaluate models using synthetic datasets, focussing on precision, recall, F1 score, balanced accuracy, and geometric mean.
[13]	Android dataset with a total of 412 samples	The study classified applications as benign or malicious using Random Forest, J48, and Support Vector Machine (SVM) methods based on 51 API packages from 16 classes, analyzing 412 Android applications (205 benign, 207 malicious).	Rosmansyah and Dabarsyah proposed a machine-learning approach to identify malware on Android devices using API classes. The study supports the efficacy of the Random Forest algorithm in malware detection, aligning with the existing literature that emphasizes its superior performance, achieving 91,9% of the average precision.
[14]	Executable files	Analyzing malware and benign samples using static and dynamic techniques to extract discriminative features.	This research supports previous [13] by providing an analysis that shows that the random forest technique provides the best precision in the malware detection data set. As a result, this analysis supports our motivation to use a random forest technique to detect malware on a synthetic dataset.
[15]	Brazilian Malware dataset	The data set for the study was pre-processed and balanced to improve malware identification accuracy and minimise false positives and negatives. Although the study employed static analysis to extract features.	Using 57 attributes, a machine learning algorithm was developed to distinguish between benign and malicious Windows PE files, achieving 99.7% accuracy with the Brazilian Malware dataset of 100,000 samples. This highlights the effectiveness of the Random Forest model in malware detection. An unbalanced dataset poses challenges. While the study used static analysis for feature extraction, our research utilizedutilised normalization. Both studies used ranking techniques to remove ineffective features.

3. Methodology

This section provides an overview of the data, the methods used to analyze it, and how the analysis was carried out.

3.1. Data collection

This study used the latest Android Malware Secondary dataset, CICMalDroid2020, collected from numerous sources, including Contangio Security Blog, AMD, VirusTotal, and MalDozer [17], [18]. Samples from Android up to 2018 are included in this collection. The data are publicly available at the Canadian Institute of Cybersecurity. These secondary data include five types of samples: banking malware, adware, riskware, SMS malware, and benign malware. Normalizing the attributes in the CICMalDroid2020 dataset is imperative because they have large values and nonnormally distributed data. It consists of 140 variables and 11,598 data entries. All data entries are of float data type. The dataset does not have missing values. The categorical size of the dataset is imbalanced, as shown by the diagram below Figure 1, which illustrates the distribution of the malware dataset.

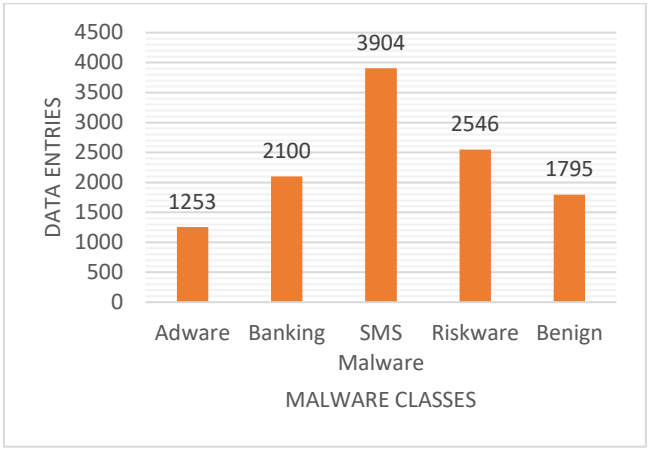


Figure 1. The original data set was imbalanced malware data distribution.

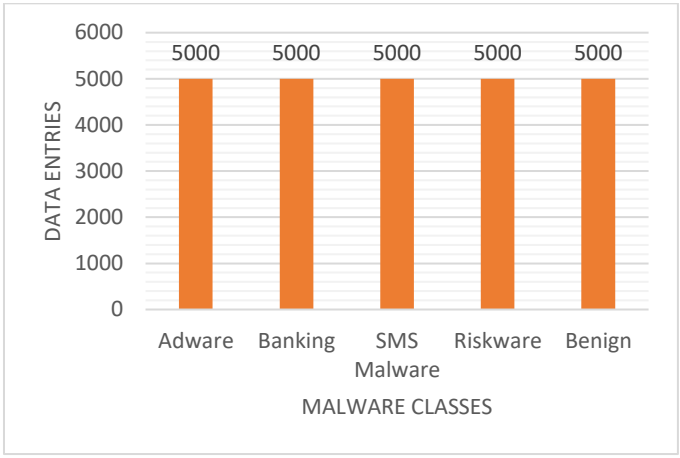


Figure 2. Balanced synthetic data distribution.

3.2. The method used to generate synthetic data sets

We employed a flow-based model to augment the existing dataset, specifically the autoregressive model. This model sequentially generates new data, predicting the value of each element based on the preceding values. The advantages of flow-based models are that they are often more efficient to train than GANs and VAEs, especially for high-dimensional data. The invertibility of the transformations allows for exact likelihood evaluation, which can be helpful for various tasks. Flow-based models can be designed to capture complex data distributions, making them suitable for various applications. The sequence of transformations can provide information on the structure of the data. The original dataset contained 11,598 data entries; the flow-based model generated 13,402 samples, resulting in 25,000 data entries.

Since the data was unbalanced, to address the issue of the unbalanced dataset, we proposed the synthetic minority oversampling technique (SMOTE). SMOTE generates synthetic data by creating new samples based on a small subset of the minority class. Unlike simple oversampling, SMOTE generates synthetic samples rather than replicating existing ones. The categorical size of the dataset is balanced, as shown by the diagram below Figure 2, which illustrates the distribution of the malware data set.

The results generated were graphically represented for analysis. Table 2 presents the parameters used to evaluate the algorithms.

Table 2. The parameters used.

Algorithm	Training parameters
Random forest	Max depth =10; Max features =auto; min sample leaf=3; min samples split 5; min weight fraction leaf 0.0; n-estimator =2000; verbose =0

Table 3 shows the environment used to train and test machine learning algorithms. The operating system used to run the project was Windows 10, running on an HP Intel Core i5 8th generation computer.

Table 3. Computational environment.

Parameters	Values
Operating system	Windows 10
Dataset	CICMalDroid2020, 8 GB
Machine Model	HP Intel Core i5 8th
Random memory	Anaconda-2021 and Jupyter notebook
Tools	

3.3. Evaluation metrics

The performance of random forest techniques was assessed using several metrics, including precision, recall, F1 score, balanced accuracy, geometric mean, and overall accuracy. Four of these performance measures are detailed below, with TN indicating True Negative, TP denoting True Positive, FT referring to False Positive, and FN indicating False Negative.

a) **Accuracy:** is the ratio of correct predictions for both TP and TN attacks to the total number of attacks tested.

$$Accuracy = \frac{TP+TN}{FP+FN+TP+TN} \quad (1)$$

b) **Precision:** Also referred to as positive predictive value, this metric assesses the effectiveness of classification algorithms. It is determined by taking the ratio of True Positives (TP) to the total of True Positives and False Positives (TP + FP).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

c) **Recall:** Recall aims to assess True Positive (TP) instances of False Negative (FN) items that have not been classified. This concept is nearly identical to the detection rate (DR), actual positive rate (TPR), and recall.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

d) **F1-Score:** The F1 score serves as a metric that strikes a balance between precision and recall.

$$F1_{Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

e) **Balanced accuracy (BA):** The accuracy of a classifier is measured using this metric. When classes are unbalanced, this method is used [16].

$$BA = \frac{Precision+Recall}{2} \quad (5)$$

f) **Geometric Mean (GM):** This metric maximizes the TP and the TN rates while keeping them relatively balanced [16].

$$GM = \sqrt{TP} * TN \quad (6)$$

3.4. Synthetic dataset

Synthetic data are artificial data generated on statistical properties and characteristics of accurate data while ensuring privacy and confidentiality. Synthetic data is generated algorithmically to mimic the characteristics of real datasets, not based on any observations. Synthetic data safeguards sensitive information, allowing data sharing without privacy breaches. Additionally, it addresses data scarcity and class imbalance in machine learning, enhancing model training and testing. Furthermore, tailored methods reduce biases in the data sets, promoting fairness in the models. Exploratory analysis enables testing and analysis without risking real-world data or privacy issues [19].

Approaches to generating synthetic data are appropriate depending on the type of data we care about and what we want to achieve. There are various ways to do this; a few include the following. Generative Adversarial Networks (GANs), Flow-based models, Variational Autoencoders (VAEs), Copulas, and Synthetic Minority Over-sampling Technique (SMOTE) [20], [21].

In this study, we used Flow-based models to generate synthetic datasets. A simple probability distribution, such as a standard Gaussian, is progressively transformed into a more complex distribution that approximates accurate data distribution through a sequence of invertible transformations to construct a flow-based model. Synthetic Data Generation challenges include data quality, bias checks, real-world testing, and computational costs. Synthetic datasets offer solutions to privacy, scarcity, and bias issues, unlocking new opportunities for researchers and organizations in data science and artificial intelligence [22].

4. Experiments and Results

The experimental results were obtained by evaluating the Random Forest models described in Section 3.3. Training was conducted on 80% of the data, and testing was conducted on the remaining 20%. The performance of the Random Forest model was evaluated using a comprehensive set of metrics, including precision, recall, accuracy, F1 score, balanced accuracy (BA), and geometric mean (GM). As illustrated in Figure 3, the results compare the Random Forest performance of the malware detection model trained on a standard (imbalanced and small-scale) dataset and a balanced synthetic dataset.

The Random Forest achieved an overall accuracy of 85% on a standard dataset, while the balanced synthetic dataset achieved 92%. The balanced synthetic dataset increases the accuracy by 7%, proving that we have a more reliable model. The original dataset has 0.89 precision and 0.90 recall. On the contrary, the model's performance is significantly improved on the well-balanced synthetic dataset, achieving 0.93 precision and

1.00 recall. These results suggest that the model has better potential to correctly classify malicious (true positive) and benign (true negative) samples and reduce false positives and false negatives. For the balanced synthetic dataset, the model's recall rate reached 100%, which means that the model correctly identified all malware instances.

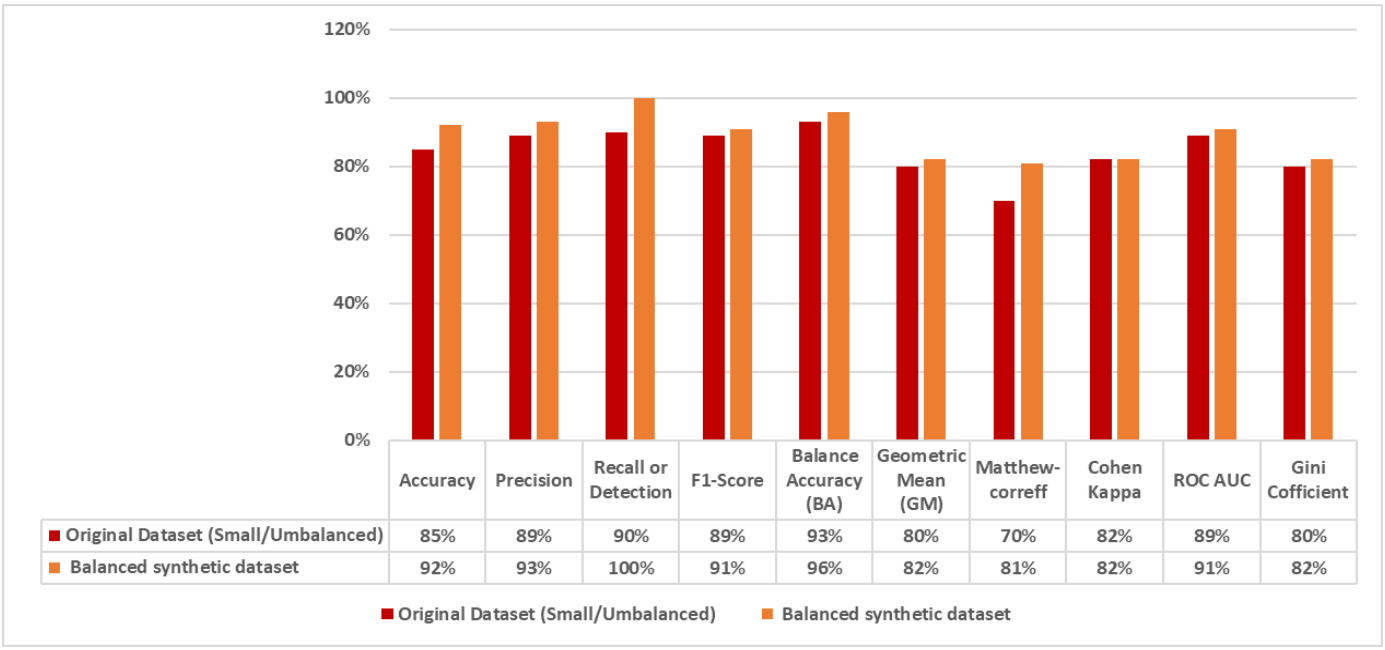
On a balanced synthetic dataset, the Random Forest achieved a 91% F1 score; on the original dataset, it achieved 89%, thus confirming the Random Forest's optimal precision-recall balance performance. The balanced precision metric slightly improved over the original precision metric by 3%, showing the potency of how synthetic data can handle this specific challenge. The GM metric evaluates the average performance of the model across all classes. This case demonstrates a very high GM value of 82% on a balanced synthetic dataset, which means the model is good at predicting positive and negative instances.

The improved correlation coefficients of Matthews, Cohen's Kappa, ROC AUC, and Gini coefficient were also obtained using a balanced synthetic dataset compared to the original dataset. The Random Forest model on this balanced synthetic dataset shows outstanding performance in malware detection, summarized by the following metrics.

Cohen's Kappa: The training set showed a perfect score of 100%, which means that the model accuracy is much greater than random chance. Cohen's Kappa in the testing set was 82%, which is still a considerable improvement compared to random detection. Area Under the Receiver Operating Characteristic Curve (AUROC): The AUROC score of 100% in the training set and 91% in the testing set confirms that the model can effectively distinguish benign from malicious samples regardless of the classification threshold.

The Gini coefficient of 100% on the training set and 82% on the testing set indicates that the model performance is significantly better than random guessing. The Gini coefficient of the test set is slightly lower, yet it still indicates good performance. Matthews correlation coefficient (MCC): 100% on the training set and 81% on the testing set, demonstrating the model's proficiency in correctly classifying positive and negative samples while remaining unaffected by class imbalance.

The study's results demonstrate the application of flow-based models to generate synthetic data to improve the performance of malware detection classifiers, particularly in the case of small-scale and unbalanced datasets.



	Accuracy	Precision	Recall or Detection	F1-Score	Balance Accuracy (BA)	Geometric Mean (GM)	Matthew-correff	Cohen Kappa	ROC AUC	Gini Cofficient
Original Dataset (Small/Umbalanced)	85%	89%	90%	89%	93%	80%	70%	82%	89%	80%
Balanced synthetic dataset	92%	93%	100%	91%	96%	82%	81%	82%	91%	82%

Figure 3. The performance of the Random Forest Method.

5. Conclusion

This paper explored random forest models as an effective method for malware detection on synthetic data generated using flow-based models. Overall, the results show the promise of this method in overcoming the limitations of classic malware detection techniques. Small-scale and unbalanced data sets pose challenges to standard approaches. Random Forest models on synthetic data yielded higher scores on accuracy, precision, recall, F1 score, balanced accuracy, and geometric mean compared to traditional methods.

They similarly showed generalization to unseen data and were robust to variations in malware. Synthetic data solved the typical problem of class imbalance in real-world malware datasets. Synthetic data could be generated on the chosen parameters, which offered the advantage of flexibility regarding the study of malware properties and improving detection models.

The Random Forest model, although powerful, can be computationally intensive for larger datasets and may restrict real-world applications when scaled. New threats keep emerging, and the malware landscape is ever-changing. Since the data-building process was done until October 2023, we need to update our model occasionally with new malware variants. Future work could investigate methods to improve the training procedure or reduce the complexity of the model.

The system's performance may only apply to new and unseen malware families. Furthermore, random forest models are also not generally well understood, and techniques to explain how the model makes a decision could improve the transparency and trustworthiness of the model. Moreover, integrating synthetic data with other techniques, such as deep learning or adversarial training, may enhance the detection of such attacks. The findings of this study suggest that, with proper training and application of flow-based models for synthetic data generation, a Random Forest model can be deemed to have viable predictive capacity for malware detection. This method can help improve the protection and security of a computer system and a network by discussing the constraints and future research paths.

6. Future Work

Several areas for future work can be noted based on the results of this research. Using hybrid models (e.g., Random Forest + deep learning, Random Forest + adversarial training) could also improve detection performance and compensate for individual model weaknesses. Explaining the prediction process of a Random Forest classifier would clarify the model's weaknesses and strengths and thus provide insight into how this model makes decisions. Such approaches can minimize the computational cost of Random Forest training, but the most informative training samples need to be prioritized by applying active learning strategies. Fine-tuning a pre-trained model on task-specific data has been proven beneficial, as it can reduce the amount of labeled data required and improve performance, as the model would be taking advantage of knowledge from related tasks. Investigate approaches for real-time malware detection using Random Forest and attaching threat intelligence information to improve model detection for recent malware variants and emerging threats. Improving these facets would aid in evolving superior illicit software identification methodologies that are effective, efficient, and comprehensible.

Competing Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this document.

Data Availability Statement

The dataset is available at the following link: https://www.unb.ca/cic/datasets/maldroid-2020.html

Consent to Participate and for Publication

The authors declare their consent to participate in this work as well as to its publication.

Authors' Contributions

Ms. Matsobane, designated as the first and corresponding author, played a pivotal and multifaceted role in this research endeavour. Her contributions were extensive and fundamental, encompassing the formulation and refinement of the problem statement, the conceptualization and development of the initial idea proposal, and the meticulous implementation of the research methodology. Furthermore, she conducted rigorous analyses of the data, meticulously drafted the initial manuscript, and spearheaded the comprehensive writing of the research paper. Ms. Matsobane was instrumental in the overall conception and design of the study, and she significantly enhanced the research by providing and utilizing advanced analytical tools and computational methods. She actively participated in in-depth discussions of the results, critically revised and substantially improved the paper's clarity and coherence and meticulously reviewed the draft to ensure the inclusion of significant intellectual content.

Mr. Mokwena, as a co-author, provided crucial oversight and guidance throughout the research process. He expertly supervised all stages of the work, meticulously proofread the manuscript for accuracy and consistency, thoroughly checked the analytical methods and results, and ultimately approved all aspects of the research, ensuring the integrity and validity of the final publication.

Acknowledgments

I express my sincere gratitude to God for giving me the strength I needed to complete this research. Special thanks go to my supervisor, Prof. SM Mokwena. I am grateful for his help, direction, and consolation during my research. I thank my family for their unconditional love, support, and constant encouragement. I am incredibly grateful to the National E-Science Postgraduate Teaching and Training Platform for sponsoring me financially. I thank the University of Limpopo for allowing me to further my studies

References

[1] G. H. N. Anuththara, S. S. U. Senadheera, S. M. T. V Samarasekara, K. M. G. T. Herath, M. V. N. Godapitiya, and D. I. De Silva, "Volume-12 Issue-2," *International Journal of Recent Technology and Engineering (IJRTE)*, pp. 2277–3878, 2023, doi: 10.35940/ijrte.B7671.0712223.

[2] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions," Mar. 01, 2023, *MDPI*. doi: 10.3390/electronics12061333.

[3] R. U. Khan, X. Zhang, M. Alazab, and R. Kumar, "An improved convolutional neural network model for intrusion detection in networks," in *Proceedings - 2019 Cybersecurity and Cyberforensics Conference, CCC 2019*, Institute of Electrical and Electronics Engineers Inc., May 2019, pp. 74–77. doi: 10.1109/CCC.2019.000-6.

[4] N. Martins, J. M. Cruz, T. Cruz, and P. Henriques Abreu, "Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review," 2020, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2020.2974752.

[5] H. Attou *et al.*, "Towards an Intelligent Intrusion Detection System to Detect Malicious Activities in Cloud Computing," *Applied Sciences (Switzerland)*, vol. 13, no. 17, Sep. 2023, doi: 10.3390/app13179588.

[6] F. C. C. Garcia and F. P. Muga II, "Random Forest for Malware Classification," Sep. 2016, Accessed: Feb. 26, 2025. [Online]. Available: https://arxiv.org/abs/1609.07770v1

[7] U. E. H. Tayyab, F. B. Khan, M. H. Durad, A. Khan, and Y. S. Lee, "A Survey of the Recent Trends in Deep Learning Based Malware Detection," Dec. 01, 2022, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/jcp2040041.

[8] L. Yuan, S. Yu, Z. Yang, M. Duan, and K. Li, "A data balancing approach based on generative adversarial network," *Future Generation Computer Systems*, vol. 141, pp. 768–776, Apr. 2023, doi: 10.1016/j.future.2022.12.024.

[9] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: a comprehensive survey," *Journal of Defense Modeling and Simulation*, vol. 19, no. 1, pp. 57–106, Jan. 2022, doi: 10.1177/1548512920951275.

[10] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," 2020, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2019.2963724.

[11] Z. Wang, Q. She, and T. E. Ward, "Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy," Mar. 31, 2021, *Association for Computing Machinery*. doi: 10.1145/3439723.

[12] Z. Sawadogo, G. Mendy, J. M. Dembele, and S. Ouya, "Android malware detection: Investigating the impact of imbalanced data-sets on the performance of machine

learning models.," in *International Conference on Advanced Communication Technology, ICACT*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 435–441. doi: 10.23919/ICACT53585.2022.9728833.

[13] Westyarian, Y. Rosmansyah, and B. Dabarsyah, "Malware detection on Android smartphones using API class and machine learning," in *Proceedings - 5th International Conference on Electrical Engineering and Informatics: Bridging the Knowledge between Academic, Industry, and Community, ICEEI 2015*, Institute of Electrical and Electronics Engineers Inc., Dec. 2015, pp. 294–297. doi: 10.1109/ICEEI.2015.7352513.

[14] J. Singh and J. Singh, "A survey on machine learning-based malware detection in executable files," Jan. 01, 2021, *Elsevier B.V.* doi: 10.1016/j.sysarc.2020.101861.

[15] A. Kumar *et al.*, "Malware detection using machine learning," in *Communications in Computer and Information Science*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 61–71. doi: 10.1007/978-3-030-65384-2_5.

[16] D. T. Dehkordy and A. Rasoolzadegan, "A new machine learning-based method for android malware detection on imbalanced dataset," *Multimed Tools Appl*, vol. 80, no. 16, pp. 24533–24554, Jul. 2021, doi: 10.1007/s11042-021-10647-z.

[17] S. Mahdavifar, A. F. Abdul Kadir, R. Fatemi, D. Alhadidi, and A. A. Ghorbani, "Dynamic Android Malware Category Classification using Semi-Supervised Deep Learning," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, IEEE, Aug. 2020, pp. 515–522. doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00094.

[18] S. Mahdavifar, D. Alhadidi, and A. A. Ghorbani, "Effective and Efficient Hybrid Android Malware Classification Using Pseudo-Label Stacked Auto-Encoder," *Journal of Network and Systems Management*, vol. 30, no. 1, Jan. 2022, doi: 10.1007/s10922-021-09634-4.

[19] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[20] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative Adversarial Networks: An Overview," Jan. 01, 2018, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/MSP.2017.2765202.

[21] M. Jamoos, A. M. Mora, M. AlKhanafseh, and O. Surakhi, "A New Data-Balancing Approach Based on Generative Adversarial Network for Network Intrusion Detection System," *Electronics 2023, Vol. 12, Page 2851*, vol. 12, no. 13, p. 2851, Jun. 2023, doi: 10.3390/ELECTRONICS12132851.

[22] M. Goyal and R. Kumar, "Machine Learning for Malware Detection on Balanced and Imbalanced Datasets," *2020 International Conference on Decision Aid Sciences and Application, DASA 2020*, pp. 867–871, Nov. 2020, doi: 10.1109/DASA51403.2020.9317206.