Enhancing malware detection with genetic algorithms and generative adversarial networks

Abid Dhiya Eddine[1,2], Ghazli Abdelkader[1,2], Bouache Mourad[1,3]
[1]Department of Computer Science, Faculty of Exact Sciences, Tahri Mohamed University, Bechar, Algeria
[2]Innovations in Informatics and Engineering Laboratory (INIE LAB), Tahri Mohamed University, Bechar, Algeria
[3]High Performance Computing Center, University of Stanford, California, United States

Article Info

Article history:

Received Jul 20, 2024
Revised Jan 8, 2025
Accepted Mar 3, 2025

Keywords:

Cybersecurity
Deep learning
Generative artificial intelligence
Genetic algorithm
Malware detection

ABSTRACT

Malware detection is a critical task in cybersecurity, necessitating the creation of robust and accurate detection models. Our proposal employs a holistic methodology for identifying and mitigating malware using deep learning techniques. Initially, a customized genetic algorithm is employed for feature selection, reducing dimensionality and enhancing the discriminatory power of the dataset. Subsequently, a deep neural network is trained on the selected features, achieving high accuracy and robust performance in distinguishing between malware and benign data. Generative adversarial networks are also utilized to evaluate model effectiveness on unseen data and ensure the model's robustness and generalization capabilities. Evaluation of the proposed model demonstrates accurate malware detection with high generalization capabilities. Furthermore, future research should focus on developing and deploying practical tools or systems that implement the proposed model for real-time malware detection in operational environments. This research makes a significant contribution to the field of malware detection and provides excellent opportunities for practical implementation in the field of cybersecurity.

Corresponding Author:

Abid Dhiya Eddine
Department of Mathematics and Computer Sciences, Faculty of Exact Sciences, Tahri Mohamed University of Bechar
Istiklal Street, 08000, Bechar, Algeria
Email: abid.dhiyaeddine@gmail.com

1. INTRODUCTION

Technology is advancing daily and being used more frequently. This advancement in technology also creates a conducive atmosphere for the spread of malware, [1] leading to data breaches, financial losses, and disruptions of critical infrastructure [2]. According to the statistics, every day, the AV-TEST Institute detects over 450,000 novel potentially unwanted applications (PUA) and malicious programs (malware) [3]. Figure 1 shows the total amount of malware and PUA in the last five years according to AV-TEST Institute.

Malware, often known as malicious program, is widely acknowledged as one of the most dangerous cyberthreats and risks to contemporary computer systems. This all-encompassing term describes any code that might have negative damaging effects [4]. It alludes to a broad spectrum of harmful software that might manifest in various ways. These include ransomware, adware, spyware, trojan horses, worms, and viruses. The malware aims to compromise the security and privacy of a user's computer by disrupting its operations, infiltrating unauthorized data or systems, or employing other methods [5]. Traditional malware detection techniques frequently depend on methods that rely on signatures, which find it difficult to keep up with the quick changes in malware types. More advanced methods that can efficiently identify and evaluate malware in real-time are therefore becoming more and more necessary, particularly in situations where harmful activity takes place in volatile memory [6].

AI approaches have become effective tools in combating malware. By utilizing machine learning techniques and deep neural networks, AI powered malware detection systems have the ability to examine extensive volumes of data where the aim is to uncover patterns and anomalies that indicate potentially dangerous activity [7]. These systems have shown promising results in detecting previously unknown malwares, where conventional approaches that rely on signatures may fall short [8]. AI powered solutions in cybersecurity provide a proactive approach by constantly acquiring knowledge from fresh data and adjusting to changing threats [9].

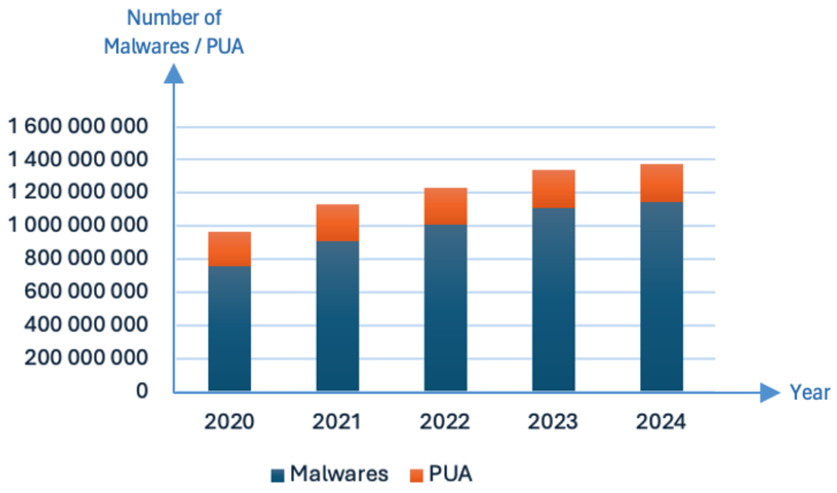

Figure 1. Total amount of malwares and PUA in the last 5 years according to AV-TEST Institute

Recent research has made significant strides in the domain of malware detection. Researchers in [10] developed a user-friendly website for malware detection and prediction, capable of identifying the type of malware encoded in a file. Minerva is new approach presented by [11] for the ransomware detection. Minerva uses all of the operations that files receive during a given time interval to create behavioral profiles of the files in order to detect ransomware. Microsoft Malwares dataset was used in [12] with many machine learning algorithms for classifying the malicious and the benign software based on the analysis of executable file metadata. MalMem-2022 it is a dataset to assess the effectiveness of memory-based obfuscated malware detection methods [13], this dataset used by [14], and also [15] used it but with a customized K-Nearest Neighbors algorithm, and it used by [1] for detecting malwares in big data environment.

Malwares detection also can be applied to mobile phones as it is presented in paper [16], where the main idea is using deep learning techniques and explainable artificial intelligence (XAI) for the detection and the classification of android mobile malwares using CICAndMal2017 dataset [17]. it is another work in the field of mobile phones which is based on stacking, presented a machine learning (ML) model for detecting malware that uses an ensemble approach for Android devices. The authors in this research merged between the CIC-MalMem 2022 and CIC-MalDroid 2020 datasets in the examination of the effectiveness of the proposed model. The Internet of Things also has a share in previous works devoted to detecting malwares, and this is what was embodied in the research [18] and [19]. These studies collectively highlight the potential of machine learning and behavior-based detection in enhancing malware detection capabilities. While all previous research has done well in creating robust models to identify malware, the question arises about the effectiveness of these models with generative data, so we will try to answer this question in this research paper.

In this research, we will use the CIC-MalMem 2022 dataset to build a model to identify malware. We will call genetic algorithm to choose the appropriate features, then we will train the final data using deep neural networks, and then we will test the effectiveness of the model using machine learning evaluation metrics. The next stage is to create a generative adversarial network, then generate data similar to the original data. Finally, we will test the effectiveness of our model with the generative data in order to discover the effectiveness of the generative data in these cases.

2. PROPOSED METHODOLOGY

To enhance malware detection, we implemented a multi-stage approach starting with the preprocessing of the CIC-MalMem 2022 dataset, as illustrated in Figure 2. A customized genetic algorithm (GA) was employed to select the most discriminative features, ensuring the model was built on the most relevant data. Using these selected features, a deep neural network (DNN) was trained to classify samples as benign or malicious. To improve the model's robustness and generalization, a generative adversarial network (GAN) was then utilized to generate synthetic benign and malicious samples. The GAN, consisting of a Generator and a Discriminator, created new data points that closely resembled real-world examples, which were used to further evaluate the DNN. The model's performance was assessed on this synthetic data to determine its effectiveness. If the model performed well, it was deemed robust; otherwise, further refinement was necessary. This integrated approach of feature selection, deep learning, and synthetic data generation ensured that our malware detection model was both accurate and resilient against diverse threats.

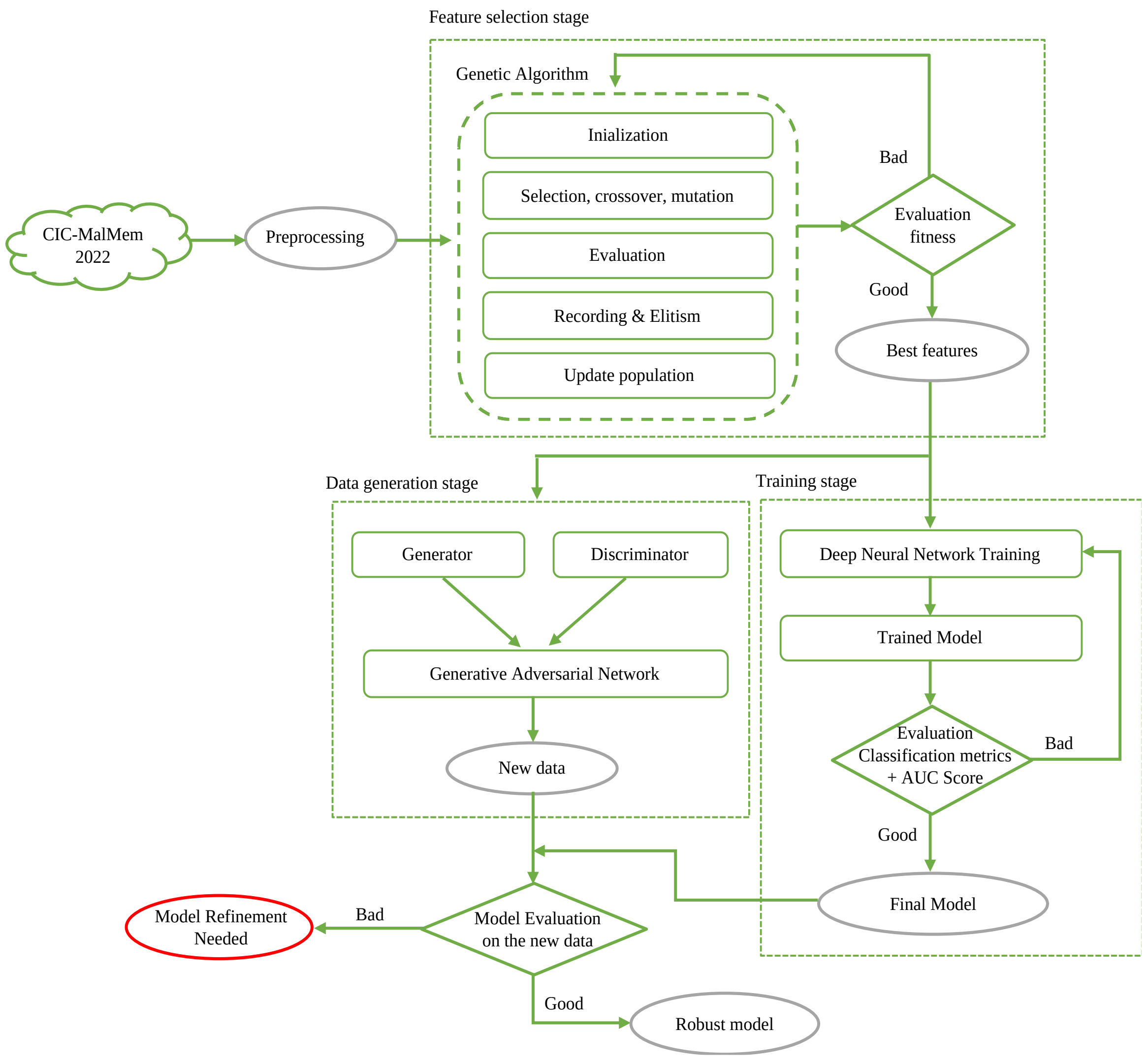

Figure 2. The proposed methodology

2.1. Data preprocessing

As mentioned previously, we used the CIC-MalMem 2022 dataset in our work. Our database contains 58596 rows and 57 columns, divided into benign (29298) and malicious (29298). The malicious class contains three categories which are Spyware (10020), Ransomware (9791), and Trojan (9487). As a pre-processing of the data, we coded the benign class as 0 and the malicious class as 1. As for the category column, we coded the benign as 0, Spyware as 1, Ransomware as 2, and Trojan as 3.

2.2. Feature selection using genetic algorithm

The selection of best features is a technique aims to list the significant features that help build better classifiers and reduce computational overload [20]. Genetic algorithms are optimization techniques inspired by natural selection. They use populations of candidate solutions, favoring the fittest individuals through iterative selection, crossover, and mutation. This mimics the process of evolution, aiding in efficiently finding optimal solutions to complex problems [21]. In our research we created a customized genetic algorithm for feature selection, this algorithm is shown in the algorithm 1.

Algorithm 1. Features selection genetic algorithm

```
Input: data, population_size, generations_number,
mutation_rate.
Output: optimal_features
population ← initialize_population_randomly()
score ← evaluate(population)
foreach generation in generations_number do
   parents ← select_parents_for_crossover(population,
   population_size, score)
   next_generation ← create_next_generation(parents)
   next_generation ← make_mutation(next_generation)
   score ← evaluate(next_generation)
   add_to_results(score, next_generation)
   next_generation ← elitism(next_generation)
   population ← next_generation
end
optimal_features ← get_best_features_from_max_score()
```

Through the iterative application of selection, crossover, and mutation operators, genetic algorithms explore the solution space, gradually improving the quality of solutions over successive generations. By promoting the survival and reproduction of individuals with higher fitness values, genetic algorithms efficiently navigate complex search spaces and discover high-quality solutions to optimization problems in a manner inspired by natural evolution.

2.3. Data training

For the data training, we trained our final data using deep neural networks. A deep neural network is same as an artificial neural network but with many hidden layers positioned between the input and output layers, enabling it to learn complex patterns in data. It is characterized by its depth, which allows it to represent hierarchical features and abstract representations [22]. The Table 1 shows the structure of our deep neural network however Table 2 shows the used parameters for the training process.

Table 1. The structure of the deep neural network

Layer (type)	Output shape
Dense	(none, 128)
Dropout	(none, 128)
Dense	(none, 64)
Dropout	(none, 64)
Dense	(none, 1)
Activation (sigmoid)	(none, 1)

Table 2. The used parameters for training deep neural network

Parameter	Value
Learning rate	0.001
Optimizer	Adam
Loss function	Binary cross entropy
Epochs	30

In order to assess the efficacity of our model, we have used assessment criteria including accuracy, precision, recall, and F1-Score. Accuracy is calculated by the proportion of properly identified cases, and it is computed using (1):

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

The precision measure, which quantifies the accuracy of positive predictions, is computed using (2):

$$Precision = \frac{\mathrm{TP}}{TP+FP} \tag{2}$$

Recall or sensitivity, it is defined by the proportion of true positive cases that were accurately predicted. It is determined using (3):

$$Recall = \frac{\mathrm{TP}}{TP+FN} \tag{3}$$

The F1-Score is a quantitative measure that combines recall and accuracy using their harmonic mean, achieving a balanced compromise between the two metrics [23], it calculated by (4):

$$F1 - Score = \frac{2 \text{ x } Precision \text{ x } Recall}{Precision+Recall} \tag{4}$$

Another method for evaluating the performance of classification models is called receiver operating characteristic (ROC), and it is a graphical representation that shows how a binary classifier system may be made to function as a diagnostic across different threshold settings. The ROC curve illustrates the trade-off between the classifier's sensitivity and specificity by plotting the true positive rate (TPR) versus the false positive rate (FPR) at different threshold levels. area under the curve (AUC) is a scale from 0 to 1, with a higher number denoting the model's superior discriminating power [24].

2.4. Generative adversarial networks-based evaluation

Generative adversarial networks (GANs) are a type of artificial intelligence algorithms that comprise two neural networks, which are generator and discriminator. These networks are trained concurrently using a competitive method. The generator makes authentic synthetic data samples, while the discriminator acquires the ability to differentiate between genuine and counterfeit data. By employing iterative training, GANs have the ability to produce synthetic data of exceptional quality that nearly mirrors the distributions of actual data. This makes GANs very effective tools for many tasks, including picture synthesis, data augmentation, and unsupervised learning [25], [26]. In our case, we used a generative adversarial network in order to generate data similar to the original data and test the effectiveness of our model when confronted with the generated data. Table 3 shows the structure of our generator neural network while Table 4 shows the structure of our discriminator neural network.

Table 3. The generator structure

Generator (Input: 100-dimensional noise vector)	
Layer (type)	Output Shape
Dense	(None, 256)
LeakyReLU	(None, 256)
BatchNormalization	(None, 256)
Dense	(None, 512)
LeakyReLU	(None, 512)
BatchNormalization	(None, 512)
Dense	(None, 1024)
LeakyReLU	(None, 1024)
BatchNormalization	(None, 1024)
Dense	(None, 18)
Activation (Tanh)	(None, 18)

Table 4. The discriminator structure

Discriminator (Input: Data samples with 18 columns)	
Layer (type)	Output Shape
Dense	(None, 512)
LeakyReLU	(None, 512)
Dropout	(None, 512)
Dense	(None, 256)
LeakyReLU	(None, 256)
Dropout	(None, 256)
Dense	(None, 1)
Activation (Sigmoid)	(None, 1)

The generator takes a 100-dimensional noise vector as input. It consists of fully connected (Dense) layers followed by LeakyReLU activation functions and BatchNormalization layers to stabilize training. The final layer uses the Tanh in order to ensure that the output values are within the range [-1, 1], suitable for normalization if necessary.

The discriminator takes data samples with 18 columns as input (real or synthetic). It uses a fully connected (Dense) layers with LeakyReLU activation functions and Dropout layers to avoid as max the overfitting. The final layer uses the Sigmoid activation function to output a probability (0 to 1) indicating the likelihood of the input being real. Table 5 shows the used parameters for training the generative adversarial network.

Table 5. The used parameters for training the generator and the discriminator

Parameter	Value
Learning rate	0.0001
Optimizer	Adam
Loss function	Binary cross entropy
Epochs	200

3. RESULTS AND DISCUSSION

In this research a deep learning model based on genetic algorithms and generative adversarial networks was successfully created and tested. In this section we will present and discuss the results we gotten. Which include feature selection based on genetic algorithms, data training using deep neural network and generation of new data using generative adversarial network and the evaluation of the model using the generated data.

3.1. Feature selection

In the stage of feature selection, we employed a customized genetic algorithm with an adaptive mutation rate, binary chromosome representation, tournament selection, single-point crossover, and an elitism strategy to retain the best solution for the next generation. The fitness function evaluated the classification accuracy using a random forest model, resulting in the selection of 18 features from the original 57, achieving an accuracy of 99.93%. This approach not only simplified the model by reducing its dimensionality but also potentially improved its interpretability and generalization. The high accuracy indicates that the selected features contain significant discriminatory information for distinguishing between malware and benign data, demonstrating the effectiveness of the genetic algorithm in identifying the most relevant features and enhancing the performance of the intrusion detection system.

3.2. Training process

After selecting the optimal set of 18 features using the customized genetic algorithm, we proceeded to train a deep neural network (DNN) on the reduced feature set. The model was trained over multiple epochs, with the hyperparameters meticulously tuned to optimize performance. Table 6 shows our training results.

Table 6. Training results

Evaluation metric	Value
Precision	0.9950
Recall	0.9961
F1-Score	0.9955
Accuracy	0.9956
AUC Score	0.9955

The metrics mentioned in Table 6 indicate a highly accurate model. The precision and recall values, both exceeding 99%, suggest that the model is highly effective in distinguishing between malware and benign samples. The F1-Score, which balances precision and recall, further confirms the model's capability to minimize both false positives and false negatives. However, the AUC score of 0.9955 demonstrates that the model performs exceptionally well across different classification thresholds, indicating strong overall discrimination ability.

Figure 3 presents the confusion matrix of our model which provides further insight into the model's performance. Out of the total instances evaluated, the DNN made only 52 misclassifications, comprising 29 false positives and 23 false negatives. This low error rate underscores the model's robustness and reliability in detecting malware. In summary, the training results illustrate that the DNN, trained on the reduced feature set, achieves near perfect classification performance. The high precision, recall, F1-Score, accuracy, and AUC score collectively demonstrate the model's effectiveness in accurately identifying malware while maintaining a low misclassification rate.

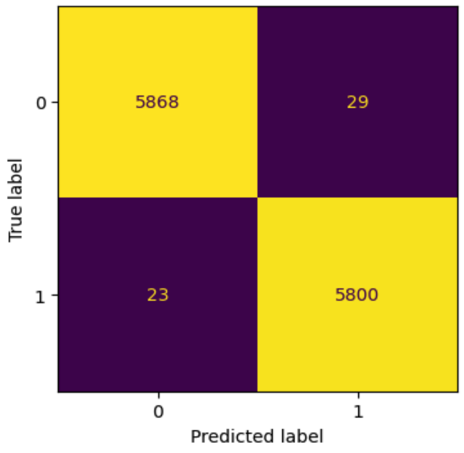

Figure 3. Confusion matrix of our model

3.3. Model evaluation using the generated data

This phase involved generating new data, including both benign and malicious samples, to further evaluate the effectiveness of our model. We created a GAN and used it to generate 100 benign and 100 malicious samples, which were then used to test the model's performance. Figure 4 shows the loss of the generator and discriminator with the malicious generated data; however, Figure 5 shows the loss of the generator and discriminator with the benign generated data. Figure 6 illustrate some generated malicious data while the Figure 7 illustrate some generated benign data. When the generated samples were passed to the model, it recognized the malicious samples with 100% accuracy, while the benign samples were recognized with 96% accuracy. This means that 4 benign samples were incorrectly classified as malicious.

Utilizing a GAN to generate new data helps assess the model's effectiveness against unseen data. Achieving 100% accuracy with the generated malicious samples demonstrates that the GAN effectively captured the underlying distribution of malicious data. However, the 96% accuracy with benign data suggests some room for improvement, possibly due to the inherent diversity and complexity of benign data. This evaluation highlights the model's robustness and ability to generalize to unseen data, even when that data is synthetically generated.

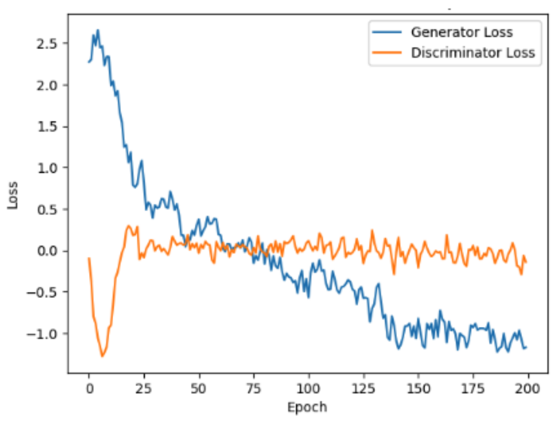

Figure 4. The loss of the malicious generated data

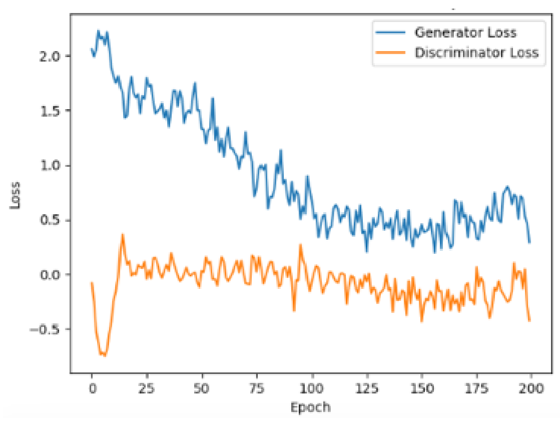

Figure 5. The loss of the benign generated data

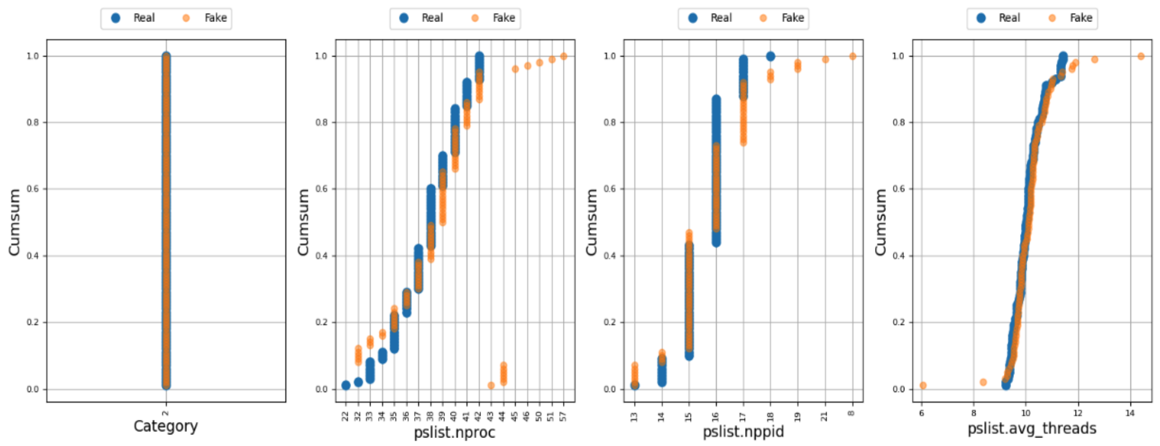

Figure 6. Some of the generated malicious data

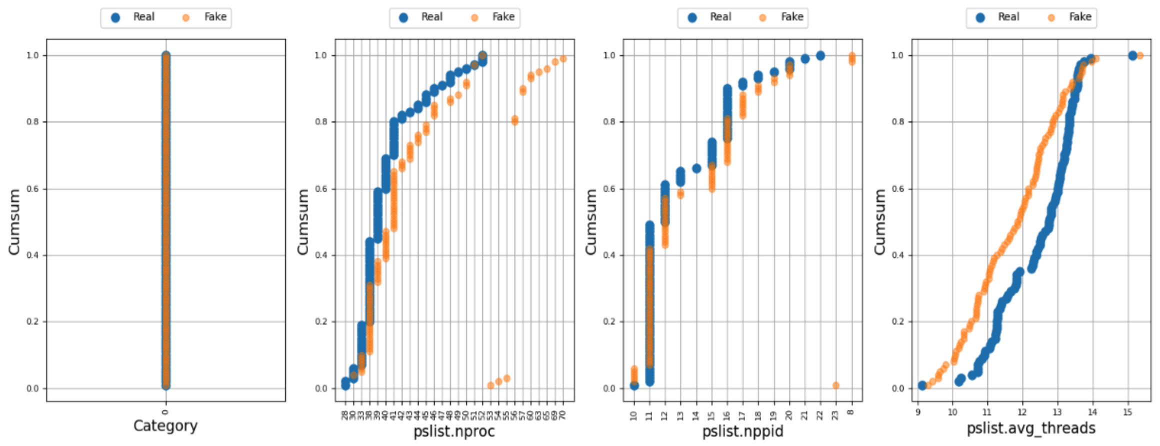

Figure 7. Some of the generated benign data

4. COMPARISON, LIMITATIONS AND FUTURE WORKS

As a simple comparison with previous works, our research is characterized by its use of genetic algorithm to select the most important features in addition to relying on the smallest possible number of features (18), which allows us to show more accurate results during the training process. Our work is also characterized by adding another evaluation stage based on the use of generative adversarial networks in order evaluate the effectiveness of the model, unlike other works that were satisfied with accuracy precision, recall and F1-Score. This can confirm the model's efficiency in working with new and unseen data. As for accuracy, the reason for its decrease compared to works [1], [15] and [19] is due to our use of two dropout layers while training our deep neural network, in order to avoid falling into a state of overfitting. Table 7 resume the comparison of our work with the similar works.

On the other side, our dataset does not represent the full spectrum of real-world malware for this, the model's effectiveness might be limited. We can say also that GANs might not always capture the full diversity of data, as indicated by the 96% accuracy with benign data. Another sensitive point is, that deep neural networks and GANs are often considered black box models, making it difficult to interpret their decision-making process. This lack of transparency can be a limitation in understanding why certain data is classified as malware, The implementation of explainable AI techniques will be useful for these cases. Scaling the model to handle larger datasets or real-time detection scenarios might pose a good challenge. Future research endeavors could advance this work by focusing on the development and implementation of practical tools, software solutions, or real-time systems that leverage the proposed model for effective and efficient malware detection in operational environments with the integration of explainable AI techniques for more transparent and interpretable decision-making processes.

Table 7. Comparison of our work with similar works

Ref	Dataset	Feature selection technique	Number of features	Training technique	Accuracy	Evaluation metrics
[1]	CIC-MalMem-2022	-	52	Logistic regression (LR)	99.97%	Precision Recall F1-Score AUC Score
[10]	Malware Api Call	-	-	RF	96.9%	Precision Recall F1-Score
[14]	CIC-MalMem-2022	-	57	RF	99%	Precision Recall F1-Score
[15]	MalMem-2022	-	55	KNN	99.97%	Precision Recall F1-Score
[16]	-	-	-	CNN	99.53% (AUC Score)	AUC Score Precision
[17]	CIC-MalMem 2022 CIC-MalDroid 2020	-	-	Stacking technique including: − Support vector machine (SVM) − Catboost − Histogram gradient boosting − Random forest (RF)	98% 99.99%	Precision Recall F1-Score
[18]	CIC-MalMem-2022	-	55	RF	98.45%	Precision Recall F1-Score
[19]	CIC-Malmem-2022	-	56	CNN + Bi-LSTM	99.96%	Precision Recall F1-Score
Our research	CIC-MalMem 2022	Genetic algorithm	18	DNN	99.56	Precision Recall F1-Score AUC score Syntheticdata using GANs

5. CONCLUSION

In this research, we utilized a genetic algorithm for feature selection, trained a deep neural network on the selected features, and employed generative adversarial networks to evaluate the effectiveness of our model against the new and unseen data and make it more generalized. The combination of feature selection, DNN training, and data generation techniques demonstrates a holistic and sophisticated approach to malware detection. The high accuracy and performance metrics obtained across different stages of the study indicate the effectiveness of the proposed methodology. These results have significant implications for cybersecurity, as they provide insights into the potential of machine learning models for accurately detecting malware and distinguishing it from benign data. Furthermore, the application of GANs to generate synthetic data played a crucial role in our evaluation process. The GAN-generated data mimicked real-world malware scenarios, providing a robust and diverse testing environment. This allowed us to rigorously assess the performance of our model against previously unseen and potentially sophisticated malware variants. In conclusion, this research represents a comprehensive and impactful contribution to the field of malware detection, showcasing the potential of machine learning techniques for addressing cybersecurity challenges. The impressive results obtained across various stages of the study underscore the effectiveness and promise of the proposed methodology.

ACKNOWLEDGMENTS

This section should acknowledge individuals who provided personal assistance to the work but do not meet the criteria for authorship, detailing their contributions. It is imperative to obtain consent from all individuals listed in the acknowledgments.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Abid Dhiya Eddine	✓	✓	✓	✓		✓		✓	✓	✓	✓		✓	
Ghazli Abdelkader	✓	✓		✓	✓	✓				✓	✓	✓		
Bouache Mourad					✓	✓					✓	✓		

C	:	**C**onceptualization	I	:	**I**nvestigation	Vi	:	**Vi**sualization
M	:	**M**ethodology	R	:	**R**esources	Su	:	**Su**pervision
So	:	**So**ftware	D	:	**D**ata Curation	P	:	**P**roject administration
Va	:	**Va**lidation	O	:	Writing - **O**riginal Draft	Fu	:	**Fu**nding acquisition
Fo	:	**Fo**rmal analysis	E	:	Writing - Review & **E**diting			

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest

DATA AVAILABILITY

The data that support the findings of this study are openly available in Kaggle at *https://www.kaggle.com/datasets/luccagodoy/obfuscated-malware-memory-2022-cic*

REFERENCES

[1] M. Dener, G. Ok, and A. Orman, "Malware detection using memory analysis data in big data environment," *Applied Sciences*, vol. 12, no. 17, p. 8604, Aug. 2022, doi: 10.3390/app12178604.

[2] A. Damodaran, F. Di Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection," *Journal of Computer Virology and Hacking Techniques*, vol. 13, no. 1, pp. 1–12, Dec. 2015, doi: 10.1007/s11416-015-0261-z.

[3] "Malware statistics & trends report," *AV-TEST*. https://www.av-test.org/en/statistics/malware/ (accessed Apr. 04, 2024).

[4] M. H. L. Louk and B. A. Tama, "Tree-based classifier ensembles for PE malware analysis: A performance revisit," *Algorithms*, vol. 15, no. 9, p. 332, Sep. 2022, doi: 10.3390/a15090332.

[5] K. S. Roy, T. Ahmed, P. B. Udas, M. E. Karim, and S. Majumdar, "MalHyStack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis," *Intelligent Systems with Applications*, vol. 20, p. 200283, Nov. 2023, doi: 10.1016/j.iswa.2023.200283.

[6] R. Sihwail, K. Omar, and K. Akram Zainol Ariffin, "An effective memory analysis for malware detection and classification," *Computers, Materials &amp; Continua*, vol. 67, no. 2, pp. 2301–2320, 2021, doi: 10.32604/cmc.2021.014510.

[7] D. Smith, S. Khorsandroo, and K. Roy, "Leveraging feature selection to improve the accuracy for malware detection," Jun. 2023, doi: 10.21203/rs.3.rs-3045391/v1.

[8] Y. Guo, "A review of Machine Learning-based zero-day attack detection: Challenges and future directions," *Computer Communications*, vol. 198, pp. 175–185, Jan. 2023, doi: 10.1016/j.comcom.2022.11.001.

[9] Babajide Tolulope Familoni, "Cybersecurity challenges in the age of AI: Theoretical approaches and practical solutions," *Computer Science & IT Research Journal*, vol. 5, no. 3, pp. 703–724, Mar. 2024, doi: 10.51594/csitrj.v5i3.930.

[10] T. Rajesh, K. Divya, S. Firdouse, S. Areeb, and N. Thakur, "Malware detection and prediction system using advanced machine learning algorithms," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 6, pp. 3718–3729, Jun. 2023, doi: 10.22214/ijraset.2023.54181.

[11] D. Hitaj, G. Pagnotta, F. De Gaspari, L. De Carli, and L. V. Mancini, "Minerva: A file-based ransomware detector," *arXiv:2301.11050*, Jan. 2023.

[12] S. Ritwika and K. B. Raju, "Malicious software detection and analyzation using the various machine learning algorithms," in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Oct. 2022, pp. 1–7. doi: 10.1109/icccnt54827.2022.9984402.

[13] "Malware memory analysis (CIC-MalMem-2022)," *Canadian Institute for Cybersecurity*. https://www.unb.ca/cic/datasets/malmem-2022.html (accessed Apr. 07, 2024).

[14] A. Mezina and R. Burget, "Obfuscated malware detection using dilated convolutional network," in *2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct. 2022, pp. 110–115. doi: 10.1109/icumt57764.2022.9943443.

[15] M. M. Abualhaj, A. A. Abu-Shareha, Q. Y. Shambour, A. Alsaaidah, S. N. Al-Khatib, and M. Anbar, "Customized K-nearest neighbors' algorithm for malware detection," *International Journal of Data and Network Science*, vol. 8, no. 1, pp. 431–438, 2024, doi: 10.5267/j.ijdns.2023.9.012.

[16] S. S. Vanjire and M. Lakshmi, "A novel method of detecting malware on Android mobile devices with explainable artificial intelligence," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 3, pp. 2019–2026, Jun. 2024, doi:

10.11591/eei.v13i3.6986.
[17] A. Joshi and S. Kumar, “Stacking-based ensemble model for malware detection in android devices,” *International Journal of Information Technology*, vol. 15, no. 6, pp. 2907–2915, Aug. 2023, doi: 10.1007/s41870-023-01392-7.
[18] A. Sharma, H. Babbar, and A. K. Vats, “Securing the internet of things: Using machine learning for malware detection with CIC-MalMem dataset,” in *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, Feb. 2024, pp. 1–5. doi: 10.1109/iciptm59628.2024.10563381.
[19] S. S. Shafin, G. Karmakar, and I. Mareels, “Obfuscated memory malware detection in resource-constrained IoT devices for smart city applications,” *Sensors*, vol. 23, no. 11, p. 5348, Jun. 2023, doi: 10.3390/s23115348.
[20] U. M. Khaire and R. Dhanalakshmi, “Stability of feature selection algorithm: A review,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1060–1073, Apr. 2022, doi: 10.1016/j.jksuci.2019.06.012.
[21] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, Oct. 2020, doi: 10.1007/s11042-020-10139-6.
[22] J. Patterson and A. Gibson, *Deep learning: a practitioner’s approach*. O’Reilly, 2017.
[23] N. Japkowicz and M. Shah, *Evaluating learning algorithms: A classification perspective*. Cambridge University Press, 2011. doi: 10.1017/cbo9780511921803.
[24] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.
[25] R. Raut, P. D. Pathak, S. R. Sakhare, and S. Patil, “Generative adversarial networks and deep learning: Theory and applications,” *Generative Adversarial Networks and Deep Learning: Theory and Applications*, pp. 1–208, 2023, doi: 10.1201/9781003203964.
[26] M. Ghayoumi, “Generative adversarial networks in practice,” *Generative Adversarial Networks in Practice*, pp. 1–642, 2023, doi: 10.1201/9781003281344.

BIOGRAPHIES OF AUTHORS

Abid Dhiya Eddine is a master's holder in artificial intelligence and decision making from the Tahri Mohamed University of Bechar, Faculty of Exact Sciences and currently He is a Ph.D. student at the same university. His broad research interests cover topics relating to cybersecurity, artificial intelligence and software engineering. He can be contacted at email: abid.dhiyaeddine@gmail.com.

Ghazli Abdelkader is a Ph.D. in computer Science. He received the diploma of teaching in Computer Science from the University of University of Science and Technology USTO of Oran, Algeria in 2009. He is a lecturer at the University of Tahri Mohamed of Bechar Algeria, His research interests are cryptography, security, artificial intelligence and generative AI. He can be contacted at email: ghazek@gmail.com.

Bouache Mourad holds a Ph.D. in computer science and has conducted postdoctoral research in France, Canada, and the United States. He worked at Yahoo for 10 years, followed by 3 years at Intel. He is currently serving as the Head of Research and Development at Meta's Artificial Intelligence Academy. His research interests include artificial intelligence, smart cities, and promoting Arabic content in technology. He is the founder of Najoom, the first Arabic generative AI platform. He can be contacted at: bouache@gmail.com and Bouache@stanford.edu.