International Journal for
Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2024.0803206

Research Article

Vol. 8 issue 3 Jul-Sep 2024

Enhanced Malware Detection Using Grey Wolf Optimization and Deep Belief Neural Networks

Zohaib Ahmad[1], Obaidullah[2], Muhammad Ammar Ashraf [3] and Muhammad Tufail[4]

[1]Faculty of Electronics and Information Engineering, Beijing University of Technology, Beijing, China.
[2]Department of Computer Science, University of Alabama at Birmingham AL 35205, USA.
[3]Department of Computer Science, Ripah international University, Sahiwal Campus, Sahiwal, Pakistan
[4]Department of Computer Science, Government Postgraduate College, Nowshera, KP, Pakistan

Correspondence Author: ahmedzohaib03@gmail.com

Received: Aug 03, 2024; **Accepted:** Aug 19, 2024; **Published:** Sep 12, 2024

ABSTRACT

Standard identification methods are flattering and less effective as attacks from malware get increasingly sophisticated. Considering current malware outbreaks employ tactics such as polymorphism, obfuscation and encryption, to avert identification, growing complicated approaches must be developed. This paper deals with a mixed model utilizing Deep Belief Neural Network (DBNN) for classifying and Grey Wolf Optimization (GWO) for choosing features. Whereas DBNN encodes complicated patterns by hierarchical learning, GWO optimizes the choosing of the more essential features, lowering the cost of computing and dataset complexity. Investigations reveal that the suggested GWO-DBNN model beats existing machine learning procedures in terms of detection accuracy, recall, precision, and false positive rate (FPR). These mixed tactics offer dependable and scalable solutions to the challenges faced by modern malware threats.

Keywords: Deep neural networks, DNNs, Malware analysis, Feature Engineering, Metaheuristic algorithms

1. INTRODUCTION

Malware, shorthand for "harmful software," is one of the largest and most major risks to broad cybersecurity nowadays. This word covers a wide range of hazardous software forms, spanning viruses, worms, ransomware, spyware, and others. Malware attacks have risen dramatically in the past few years, owing to the expansion of correlated networks, the rise of cloud computing, and the rapidly evolving digital world [1]. Typically, the identification of malware relies on signature-based platforms which match established signatures for recognizing potential dangers [2]. While such devices have been useful in the past, they endure considerable constraints especially whenever it comes to identifying zero-day attacks [3].

According to authors [4], innovative methods of evasion involving polymorphism are employed by contemporary malware creators for allowing their harmful software to alter its code architecture without losing its ability to trigger harm. Also, methods of obfuscation make it more challenging for ordinary antivirus programs to identify hazardous behavior by hiding the real code from monitoring systems [5]. According to authors [6], these approaches greatly reduce the efficacy of static signature-based identification, forcing the use of more dynamic strategies that might evolve with these dynamic challenges.

According to the authors [7] with the quick augmentation in the variation of malware methods, ML and DL have developed into acute tools in advocate the identification of malware. These approaches can cultivate behaviors and patterns from historical data, clearing them to recognize known and evolving malware whereas depending on prearranged signature. According to the authors, ML methods [8] containing Naïve Bayes, Support Vector Machines (SVM), and Random Forests may rationalize classifying via examination of dynamic and static features.

The authors [1] describe the typical machine learning approaches could have trouble with highly dimensional raw data, which might contain unimportant or replicate features. According to [6][17], over fitting is possible if models operate effectively with data used for training but harshly on data that is not known. Besides, dataset with high dimensions augments the computing cost, execution it unfitting for real-time revealing of viruses. The authors [2] described competent selection of features is vivacious to lowering redundancy and keeping valued properties for categorization

This research presents an optimized architecture incorporating GWO for picking features with DBNN for classifying. [6] Devised GWO, a metaheuristic algorithm inspired by grey wolf social structures and hunt tactics that can quickly traverse huge searching areas and select finest subset of attributes. The authors [5] discovered that GWO successfully decreases the complexity of challenging malware samples. DBNN are an unsupervised neural network architecture made consisting of layers of Restricted Boltzmann Machine. DBNNs can acquire hierarchical structures from vast data sets while enhancing the accuracy of classification by automatically recognizing complex connections among characteristics [8] [9].

Main Contributions

1. This paper leads an optimized structure that incorporates GWO for the selection of the features with DBNN for malware classifying tasks. The utilization of GWO advances feature collection by competently decreasing the size of the data samples, guaranteeing that only the best appropriate features are employed, subsequently augmenting classification enactment and sinking computational burden.
2. By engaging GWO, the suggested methodology effectually addresses the tasks impersonated by high-dimensional data samples, which frequently cover irrelevant or redundant features. This effects in a reduction of overfitting and advances the generality of the model, constructing it more appropriate for actual malware revealing.
3. By utilizing DBNN, the framework may automatically recognise hierarchy relationships in malware knowledge, boosting its ability to recognize malware variations that have been identified and those which are unknown. When contrasted with typical machine learning methods, the architecture of deep learning delivers superior accuracy in classification since it can deal with complicated feature interactions more effectively.
4. The hybrid GWO-DBNN structure delivers an accessible result for dynamic and real-time malware revealing, capable of adjusting to embryonic malware dangers such as obfuscated and polymorphic malware. This creates the model appropriate for disposition in modern cybersecurity situations where fast and adaptive detection is critical.

The remainder of the paper is organized as: Section 2 deliberates related work, Section 3 introduces our metaheuristic algorithm and deep-learning technique to malware detection classification, and Section 4 evaluates its performance in comparison to existing malware detection. Section 5 takes the paper to its conclusion.

2. LITERATURE REVIEW

2.1. Machine Learning and Malware Detection

In the domain of malware detection, there are primarily two types of analysis: static analysis and dynamic analysis. Static analysis involves extracting features from the malware code without executing it. Commonly extracted features include opcode sequences, bytecode frequencies, and control flow graphs[1]. Naïve Bayes and Decision Trees were among the earliest machine learning (ML) models used in static analysis. For example, the authors pioneered the use of Naïve Bayes to classify malware based on binary byte sequences, which represented a breakthrough in automated malware detection. While static examination has been highly successful, current malware frequently uses code obfuscation and polymorphism methodologies, causing static approaches fewer effective since malware could alter its appearance while still expressing hazardous behaviors.

Dynamic analysis, on the other hand, implements malware in a controlled environment (for example a sandbox), consenting its behavior to be monitored in the real time. This tactic records runtime behavior, containing network activity and system calls and, making it difficult for malware to evade detection. ML processes such as SVM are used in dynamic study to classify malware based on behavioral characteristics. SVM has had some success in dynamic analysis, but it struggles when dealing with huge amounts of data samples.

Despite these advances, traditional machine learning models still struggle with high-dimensional data—datasets that include numerous irrelevant or redundant features. Such data can lead to overfitting, where the model performs well on training data but poorly on unseen data [7][10]. Besides, high-dimensional data samples upsurges the computational difficulty of the models, restraining their applicability in real-time malware revealing scenarios. Consequently, real feature variety is crucial in augmenting the enactment of ML models for malware revealing by decreasing irrelevant data while stabilizing the most informative features.

2.2. Feature Selection Technique

Feature selection plays a critical role in improving the performance of machine learning models, particularly when dealing with large, high-dimensional datasets such as those used in malware detection. Feature selection helps reduce the dataset size, making the model more efficient by eliminating irrelevant and redundant features. Traditional filter-based methods such as Chi-square and Information Gain evaluate the significance of each feature independently of the classification algorithm. While these methods are computationally efficient, they often fail to capture complex interactions between features, which is essential in malware datasets.

To address these limitations, researchers easily adopted metaheuristic algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) for feature selection [7]. These algorithms are for searching high-dimensional feature spaces, as they balance exploration (searching through the solution space) and exploitation (refining the best solutions found) during the feature selection process. However, each method has its drawbacks. For example, PSO is prone to slow convergence, while ACO can have high computational overhead.

In [6] the authors stated the GWO procedure that was recognized as an effective choice for highly dimensional feature selection challenges. Motivated by the hunting behavior and social structure of grey wolves in the natural world, GWO classifies them as alpha, beta, and delta wolves, with alpha wolves being the most beneficial solution. The technique optimizes exploitation and exploration by altering the wolf's location concerning the most suitable feature set, allowing for rapid and effective convergence. It makes GWO exceptionally excellent for processing huge, complicated data sets, such as those encountered in detecting malware.

2.3. Deep Learning in Malware Classification

Deep Learning (DL) models have been transformative in the field of malware detection, particularly in handling high-dimensional data. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Neural Networks (DNNs) have been employed to automatically extract and learn patterns from large malware datasets. DL models offer the advantage of learning hierarchical features, which allows them to generalize better than traditional machine-learning methods that rely on manual feature extraction.

DBNNs are distinguished from other deep learning models by their capacity to grip complicated data samples like those used in malware classification. DBNNs are constructed up of many layers of Restricted Boltzmann Machines (RBMs), which are unsupervised learning processes. RBMs aim to acquire a probabilistic representation of the input data by minimizing the variations between the real input and the rebuilt output. After initial training on RBMs, refine the DBNN with backpropagation to optimize the system for classifications.

The structural design of DBNNs marks them as compatible with discovering sophisticated malware, for example, they are accomplished by learning multiple stages of abstraction from raw input features. This facility to model deep non-linear relations between features tolerates DBNNs to classify complex malware designs that may be neglected by typical ML models. In this examination, the combination of GWO for feature assortment and DBNNs for classification is offered to optimize the accuracy of the feature selection and classification in malware revealing.

3. METHODOLOGY

3.1. Data Preprocessing

The processing of data is an essential phase in prepping the dataset for use in training and classifications. The actual malware datasets utilized in the present research were derived from the Microsoft Malware Classification Challenge [9] on Kaggle. This collection includes more than 10,000 samples from numerous malware families, among them Ramnit, Simda, Kelihos, and Vundo. The dataset covers the static and dynamic information, such as system call traces and opcode frequency, making it suitable for both static and dynamic training.

Preprocessing encompasses numerous stages:

- Missing values can have a serious influence on the model's efficiency. In this research, the missing data has been solved via mean imputation for numerical parameters and median imputation for categorical characteristics.
- Normalization applies Min-Max scaling to align every value of the feature across 0 and 1. This guarantees that characteristics with wide ranges do not have an excessive effect on the learning process.
- For categorical features, such as malware families, one-hot encoding is used to convert categorical values into binary vectors. This avoids any ordinal interpretation of categorical variables, ensuring that the model does not infer unnecessary relationships between malware families.
- The dataset is divided into train (80%) and test (20%) batches for evaluating the efficacy of the model that was suggested. Five-fold cross-validation is implemented to verify stability while avoiding overfitting.

3.2. Feature Selection using Grey Wolf Optimization (GWO)

The procedure called GWO [11] [12] is a metaheuristic approach influenced by grey wolves' natural hunting techniques and leadership framework. GWO organizes wolves into four categories: alpha, beta, delta, and omega. The alpha wolf reveals the optimum respond (optimal feature subset), whilst the beta and delta wolves direct the search process.

The GWO method estimates the distance that exists among the wolves and their prey and repeatedly updates their locations to arrive at the optimal response. The location of updates are determined using the following formula as:

$$D_{\propto} = |C_1.X_{\propto}(t) = X(t)| \qquad (1)$$

Xα provides the alpha wolf's position, while Dα provides the distance from the optimal feature subset. The technique repeatedly alters the wolves' placements to reduce the space of features and pick the most appropriate subset for categorization.

3.3. Classification Using Deep Belief Neural Networks (DBNN)

The architecture of DBNN is shown in figure 1. After identifying the appropriate feature subset with GWO, DBNN employs it for identifying malware. DBNNs are made up of numerous layers of Restricted Boltzmann Machines (RBMs) that are unsupervised learning models [13][14]. RBM learns to rebuild input by retaining the statistical connections among hidden and visible units [15].

The weight adjustment for every RBM follows a certain rule is

$$\Delta W_{ij} = \eta(\prec V_i h_j \succ_{data} - (\prec V_i h_j \succ_{recon}) \quad (2)$$

Where h_j is the hidden unit, V_i designates the visible unit, and η represents the learning rate. After pretraining with RBMs, the DBNN is fine-tuned using backpropagation method to classify malware as either malicious or benign based on the optimized feature subset. The flow diagram of the whole methodology has been shown in figure 2

.

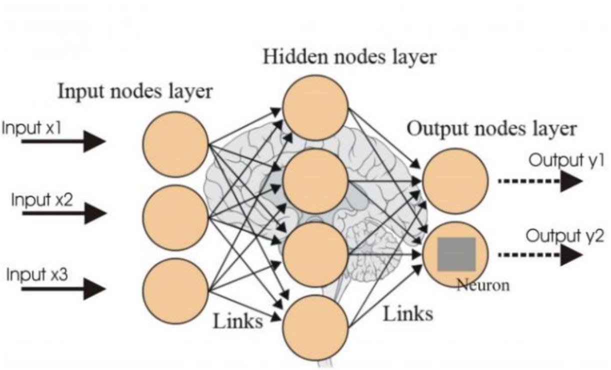

Figure 1: Architecture of DBNN

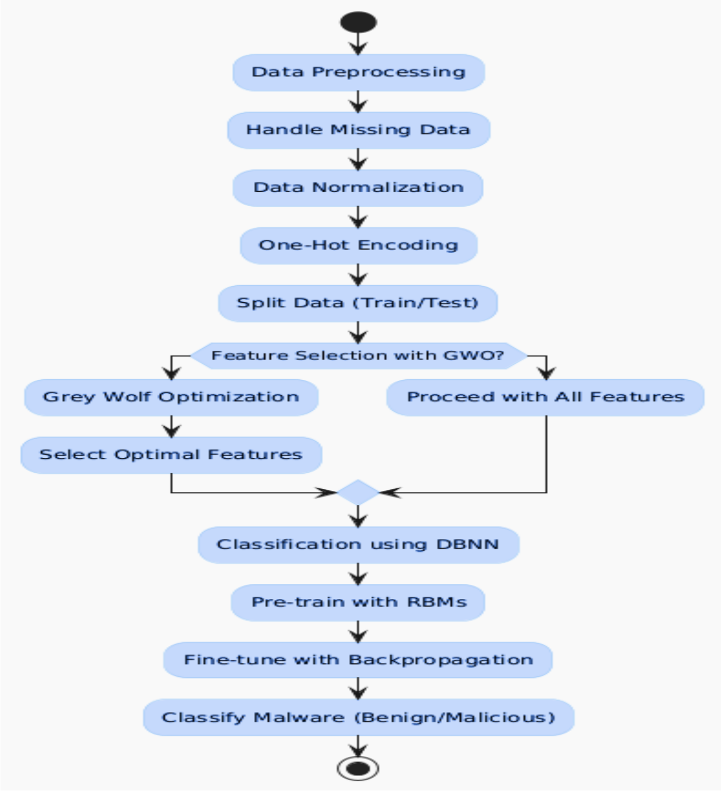

Figure 2: Flow Diagram of the proposed GWO-DBNN Malware Detection

4. RESULTS

The proposed GWO-DBNN model was evaluated using key performance metrics such as accuracy, precision, recall, F1-score, and false positive rate (FPR). The results were compared against traditional machine learning models like SVM, Naïve Bayes, and Decision Trees.

4.1. Performance Metrics

To assess the efficacy of the GWO-DBNN model, the following performance metrics have been employed:

- Accuracy: Evaluate the proportion of correctly classified instances amid the total instances.
- Recall: Replicates the proportion of true positive detections mid all actual positive instances.
- Precision: Designates the proportion of true positive detections between all positive predictions.
- F1-Score: The harmonic means of recall and precision, providing a balance between the two.
- FPR: The rate at which benign samples are incorrectly categorized as malicious.

4.2. Quantitative Results

The table below summarizes the performance of the GWO-DBNN model compared to traditional ML models on the malware detection task:

From the Figure 3, it is cleared that proposed **GWO-DBNN** model consistently outperformed traditional ML models across all metrics, particularly in terms of reducing false positives and improving overall accuracy.

Table 1: Performance Comparison

Model	Accuracy	Recall	Precision	F1-score	FPR
SVM	91.50%	90.8%	90.0%	89.40%	3.0%
Decision Tree	89.70%	87.40%	88.90%	87.80%	3.4%
Nave Bayes	88.20%	86.50%	86.40%	85.70%	4.0%
GWO-DBNN	95.80%	93.8%	93.70%	94.10%	1.7%

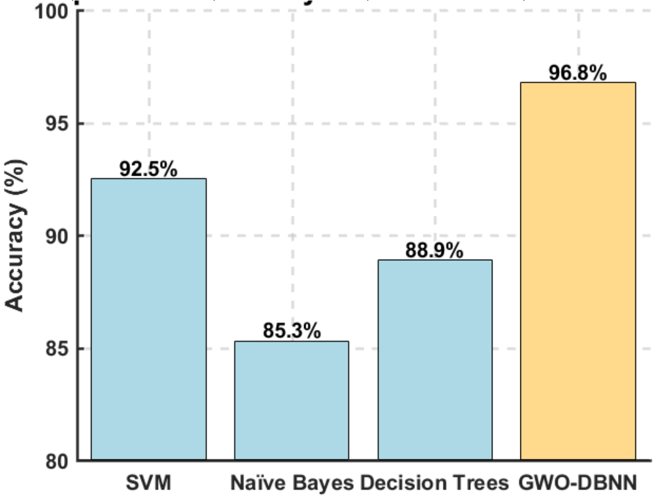

Figure 3: Comparative analysis of Malware Detection Model

5. DISCUSSION

The experimental findings show that the GWO-DBNN model improves malware identification and classification over standard ML approaches [15]. By using GWO for feature selection, the model decreases the dataset's dimensionality, increasing computing efficiency and classification accuracy.

The inclusion of DBNN enhances the model's performance by enabling hierarchical feature learning, which allows the network to automatically recognize complicated patterns in malware behavior. The combined use of GWO and DBNN has shown to be a viable technique for dealing with current malware issues such as obfuscation or zero-day attacks.

6. CONCLUSION

This article describes a unique hybrid system for identifying malware utilizing GWO for choosing features and DBNN for classifications. The suggested approach outperforms standard machine learning methods in terms of malware detection, precision, accuracy, and computing the economy at large.

Future work will focus on further optimizing the model, investigating other deep learning architectures, and increasing its application to other cybersecurity concerns such as ransomware and intrusion detection.

REFERENCES

[1] M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, “Data mining methods for detection of new malicious executables,” Proceedings of the IEEE Symposium on Security and Privacy, vol. 2001, pp. 1-11, 2001.

[2] K. Rieck, P. Trinius, C. Willems, and T. Holz, “Automatic analysis of malware behavior using machine learning,” Journal of Computer Security, vol. 19, pp. 639-668, 2011.

[3] H. S. Anderson and P. Roth, “Ember: An open dataset for training static PE malware machine learning models,” arXiv Preprint, arXiv:1804.04637, pp. 1-12, 2018.

[4] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. Blasco, “Dendroid: A text mining approach to analyzing and classifying code structures in Android malware families,” Expert Systems with Applications, vol. 41, pp. 1104-1117, 2014.

[5] X. Xu, H. Shen, and H. Chen, “Trafficav: An effective and explainable detection of mobile malware behavior using network traffic,” Proceedings of the 2016 IEEE/ACM International Symposium on Quality of Service, pp. 1-10, 2016.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” Advances in Engineering Software, vol. 69, pp. 46-61, 2014.

[7] Y. Ye, D. Wang, T. Li, and D. Ye, “An intelligent PE-malware detection system based on association mining,” Journal of Computer Virology, vol. 4, pp. 323-334, 2008.

[8] Y. Bengio, “Learning deep architectures for AI,” Foundations and Trends in Machine Learning, vol. 2, pp. 1-127, 2009.

[9] Z. Ahmad, M. S. Pathan, and A. Wajahat, “A comparative analysis of malware detection methods: Traditional vs. machine learning,” International Journal for Electronic Crime Investigation, vol. 7, pp. 3-18, 2023.

[10] R. Ahmad, H. Salahuddin, A. U. Rehman, A. Rehman, M. U. Shafiq, M. A. Tahir, and M. S. Afzal, “Enhancing database security through AI-based intrusion detection system,” Journal of Computing & Biomedical Informatics, vol. 7, pp. 1-12, 2024.

[11] H. Rezaei, O. Bozorg-Haddad, and X. Chu, “Grey wolf optimization (GWO) algorithm,” in Advanced Optimization by Nature-Inspired

Algorithms, pp. 81-91, 2018.

[12] A. Bilal, A. Alzahrani, A. Almuhaimeed, A. H. Khan, Z. Ahmad, and H. Long, "Advanced CKD detection through optimized metaheuristic modeling in healthcare informatics," Scientific Reports, vol. 14, pp. 12601, 2024.

[13] R. Khan, N. Iltaf, M. U. Shafiq, and F. U. Rehman, "Metadata-based cross-domain recommender framework using neighborhood mapping," 2023 International Conference on Sustainable Technology and Engineering (i-COSTE), pp. 1-8, 2023.

[14] M. F. Chishti, M. Rao, M. W. Raffat, and S. Rafi, "Estimating corporate risk and corporate value: An application of Altman's Z-score on the KSE-30 index," International Journal of Contemporary Issues in Social Sciences, vol. 3, pp. 2833-2841, 2024.

[15] M. U. Shafiq and A. I. Butt, "Segmentation of brain MRI using U-Net: Innovations in medical image processing," Journal of Computational Informatics & Business, vol. 1, pp. 1-15, 2024.

[16] A. Ullah, M. Waqar, S. S. Nazir, A. Adnan, M. A. Khan, M. W. Raffat, and S. Rafi, "The impact of information communication technology and financial innovation on the financial performance of Chinese commercial banks," Remittances Review, vol. 9, pp. 364-383, 2024

[17] M. Hamza, "Optimizing early detection of diabetes through retinal imaging: A comparative analysis of deep learning and machine learning algorithms," Journal of Computational Informatics & Business, vol. 1, no. 1, pp. 1-12, 2024