KAMPUS AKADEMIK PUBLISING
Jurnal Sains Student Research
Vol.3, No.6 Desember 2025
e-ISSN: 3025-9851; p-ISSN: 3025-986X, Hal 447-453
DOI : https://doi.org/10.61722/jssr.v3i6.6480

Comprehensive Evaluation of Machine Learning and Deep Learning Approaches for Malware Detection

Alfema Kansha Nandita
Republic of Indonesia Defense University
Alamat: Bogor, West Java, Indonesia
Korespondensi penulis: alfemanandita@gmail.com

Abstrak. *In an era of advancing technology, breakthrough innovations, state- of-the-art hardware development, increased computing capacity through cloud computing and supercomputers, and collaboration between research institutions and industries are the primary driv- ing forces. However, this is also accompanied by the rise in malware attacks due to increasingly complex systems and the adoption of new technologies providing vulnerabilities for attackers. High pro- cessing capabilities are exploited to develop undetectable malware, while collaboration among cybercriminals is on the rise. There- fore, effective cybersecurity protection and efforts are becoming increasingly crucial. In this study, we compared GRU, Random For- est, Gradient Boosting, LSTM, and RNN models for malware attack detection, with experimental results showing that the Gradient Boosting model achieved the highest accuracy of 99.98%.*
Keywords: *Cybersecurity, Malware Detection, Machine Learning, GRU, Random Forest, Gradient Boosting, LSTM, RNN.*

INTRODUCTION

Cybersecurity threats, particularly in the form of malware attacks, continue to evolve in complexity and frequency, posing significant challenges for organizations worldwide. Rapid and accurate detec- tion and mitigation of these threats are crucial to safeguard sensitive data, systems, and networks. Machine learning (ML) techniques have proven to be invaluable tools in the field of cybersecurity, providing the capability to effectively detect and classify malware attacks.

This study focuses on leveraging ML methods GRU, Random Forest, Gradient Boosting, LSTM, and RNN for malware attack detection. The aim of this research is to explore the effectiveness of these methods in distinguishing between malicious and benign network activities, as well as identifying and categorizing newly emerging unidentified malware types. Thus, our objective is to enhance detection strategies and proactive threat response.

Previous research has extensively investigated ML-based ap- proaches for malware detection. For instance, the study by Smith et al. (2020) utilized LSTM to analyze network traffic patterns and classify attacks with high accuracy. Studies by Jones et al. (2019) also compared the performance of various ML models, including Random Forest and Gradient Boosting, in the context of malware detection. Their findings highlighted the superiority of Random Forest in handling high data noise, while Gradient Boosting excelled in managing high-dimensional data.

Furthermore, other studies such as the work by Brown et al. (2018) focused on the use of RNN, including GRU, for network in- trusion detection, demonstrating the

Received Oktober 01, 2025; Revised November 01, 2025;Desember 01, 2025
** Alfema Kansha Nandita, alfemanandita@gmail.com*

effectiveness of these models in recognizing complex temporal patterns in traffic data. While these studies offer valuable insights, there is still room for improve- ment in detailed comparison of these five methods in the context of the latest malware detection.

Compared to existing literature, this study aims to provide a comprehensive evaluation and comparison of these five ML tech- niques in the context of malware attack detection. Researchers strive to identify the strengths and weaknesses of each method, explore performance metrics such as accuracy, precision, recall, and F1-score, and analyze their suitability for real-time deployment in cybersecurity operations. By building upon previous research and addressing gaps in current methodologies, this study aims to ad- vance state-of-the-art technology in ML-based malware detection and contribute to a stronger cybersecurity framework.

METHOD

Malware Attack Detection

Malware, short for "malicious software," is a type of software or program designed to infiltrate, damage, or steal data from computer devices, servers, or networks without the owner's consent. Malware attacks are malicious attempts utilizing such software to compro- mise systems, disrupt operations, and extract sensitive information, posing significant risks to cybersecurity. Malware attack detection using machine learning involves the use of artificial intelligence algorithms to identify patterns and anomalies in data that may indicate the presence of malware. Machine learning models can be trained on large datasets of known malware samples to learn to recognize new malware variants based on their characteristics. The advantage of using machine learning for malware detection is its ability to adapt to new and evolving threats, improving detection accuracy over time. However, the challenge lies in the constant cat- and-mouse game between cyber attackers creating sophisticated malware and defenders developing countermeasures. The threat of malware attacks remains high as cybercriminals continue to develop advanced techniques to evade detection and exploit vulner- abilities in systems. Detection using machine learning can enhance cybersecurity defenses, but it is crucial to regularly update and refine these systems to stay ahead of emerging threats.

The use of GRU (Gated Recurrent Unit), Random Forests, Gradi- ent Boosting, LSTM (Long Short-Term Memory), and RNN (Recur- rent Neural Network) methods in malware detection with machine learning provides a diverse approach to enhance detection effec- tiveness. GRU, with its ability to model sequential data like text or time sequences, can be used to analyze patterns and behaviors of malware. Random Forests, as an ensemble learning method using a number of decision trees, can help identify typical characteristics of malware and distinguish them from legitimate software. Gradient Boosting, with its sequential model enhancement approach, can improve detection accuracy by combining results from multiple models. LSTM, with its ability to understand long-term relation- ships in sequential data, can help recognize complex patterns and relationships in malware activities. Lastly, RNN, which can remem- ber previous information in sequential data, can be used to identify unusual patterns or suspicious activities characteristic of malware attacks. By combining these methods in malware detection using machine learning, it can enhance the ability to detect and mitigate evolving security threats in the current cyber world.

GRU

The GRU (Gated Recurrent Unit) model is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem and maintain long-term information in sequential data. GRU has the ability to tackle the vanishing gradient issue commonly found in traditional RNNs by using gating mechanisms to regulate information flow. The advantage of GRU lies in its efficiency in handling the vanishing gradient problem compared to LSTM, enabling faster and more efficient learning in sequential data. However, the drawback of GRU is its limited capability to store long-term information like LSTM, which may restrict the model's ability to understand long-term contexts in sequential data. The GRU model is built using the Sequential API from TensorFlow. The model consists of a GRU layer with 128 units and a Dense layer with softmax activation function for 10 output classes representing attack categories. The model is compiled with the 'adam' optimizer and 'sparse_categorical_crossentropy' loss function for multiclass classification. During the training process with 10 epochs and a batch size of 32, there was an improvement in accuracy from around 76.88% at the beginning of training to 88.94% at the end of training. The validation results also demonstrate strong performance with an accuracy reaching 89.25% in the final epoch.

Random Forests

The Random Forest model, as an ensemble algorithm, combines predictions from a number of decision trees to make a final decision based on majority voting. Its strengths include the ability to handle overfitting, work well with categorical data without normalization, and provide feature importance estimates. Despite drawbacks such as slow model development and complexity that makes overall model interpretation challenging, the model successfully achieved a high accuracy rate of 90.28% on the test data, indicating its ro- bustness in classifying data in the dataset. Key factors like merging tree predictions, handling categorical data, and feature importance estimation play crucial roles in the model's success. Despite chal- lenges like slow performance and complexity, the Random Forest model remains effective in accurately distinguishing attack cate- gories, with a classification report offering in-depth insights into the model's performance. The model's success in achieving high ac- curacy and providing valuable insights demonstrates its excellence and relevance in the context of malware attack detection.

Gradient Boosting

Gradient Boosting is a powerful approach in machine learning for building accurate prediction models. By gradually improving the weaknesses of the previous model and combining simple prediction models like decision trees, Gradient Boosting can generate more precise predictions. In the realm of malware attack detection, the application of Gradient Boosting allows for the development of prediction models that can accurately detect attacks, identifying patterns and characteristics of data associated with malware attacks with high precision. The implementation of Gradient Boosting with XGBoost Classifier has proven to be highly effective in malware attack detection, achieving an outstanding accuracy rate of approx- imately 99.99%. The model's accuracy at this level demonstrates its exceptional ability to classify data with high precision, instilling strong confidence in its capability to predict potential malware attacks accurately and reliably. With such high accuracy, Gradient Boosting with XGBoost Classifier establishes itself as a profoundly effective solution for overcoming the challenges of malware attack detection with superior performance.

In the implementation of Gradient Boosting with XGBoost Clas- sifier for malware attack detection, the steps are well-structured. Starting from data separation, encoding categorical features, split- ting data into training and testing sets, initializing the model, to evaluation using metrics such as accuracy, precision, recall, and F1 score. Despite offering high accuracy and the capacity to handle nonlinearity in data, Gradient Boosting may have some drawbacks such as vulnerability to overfitting and time-consuming training processes. However, with awareness of these limitations, Gradient Boosting remains a top choice in developing reliable and accurate prediction models.

LSTM

The Long Short-Term Memory (LSTM) network, a type of recurrent neural network (RNN), is specifically designed to overcome the van- ishing gradient problem inherent in traditional RNNs. Its ability to maintain long-term dependencies and capture sequential patterns effectively makes it advantageous over other RNNs and sequence learning methods. The LSTM architecture includes a cell, input gate, output gate, and forget gate, allowing it to remember information over arbitrary time intervals and regulate the flow of data efficiently. Commonly utilized in various applications such as handwriting recognition, speech recognition, machine translation, and health- care, the LSTM network provides short-term memory that can last for thousands of timesteps, hence the name "long short-term memory." The model evaluation results demonstrate an impressive accuracy rate of 99%, influenced by factors including the LSTM's ca- pability to learn and retain sequential patterns, data normalization with StandardScaler, appropriate dataset splitting for training and testing, parameter tuning, and training iterations. Additionally, the integration of evaluation metrics like confusion matrix, precision, recall, and F1 score contributes to a comprehensive understanding of the model's performance in detecting malware attacks.

RNN

In malware attack detection using the LSTM model in the Recurrent Neural Networks (RNNs) architecture, Recurrent Neural Networks (RNNs) are a type of neural network architecture designed to work with sequential data or data that has sequential relationships in the time domain. RNN has the ability to "remember" previous in- formation in the data sequence and use that information to make predictions in the next step. RNN helps the model understand the sequential patterns of network data such as protocols, services, and connection durations to identify unconventional attack patterns. The implementation of the LSTM model (Long Short-Term Mem- ory) in malware attack detection shows excellent performance with high accuracy up to 99.30%, precision of 99.53, recall of 99.19%, and F1 Score of 99.36%. However, the LSTM model in RNN remains effective in malware attack detection with advantages in under- standing complex sequential patterns, high accuracy, and strong short-term memory capabilities, but also has disadvantages in terms of high computational performance, complex model interpretation, and challenges in processing large data. With a combination of these strengths and weaknesses, the LSTM model in RNN proves its ability to classify data accurately and respond to attacks efficiently.

EXPERIMENTAL SETUP

This experiment is based on the analysis of a comprehensive dataset comprising 82,332 instances of network traffic transactions. Each transaction is characterized by 45 attributes,

including connec- tion duration (‘dur‘), protocol type (‘proto‘), service category (‘ser- vice‘), and connection state (‘state‘). Key numerical features encom- pass packet counts (‘spkts‘ and ‘dpkts‘), byte volumes (‘sbytes‘ and ‘dbytes‘), and various network flow metrics such as rate (‘rate‘), load (‘sload‘ and ‘dload‘), inter-packet arrival times (‘sinpkt‘ and ‘dinpkt‘), and jitter (‘sjit‘ and ‘djit‘). Additionally, TCP parameters like window sizes (‘swin‘, ‘dwin‘), timestamps (‘stcpb‘, ‘dtcpb‘), and round-trip times (‘tcprtt‘, ‘synack‘, ‘ackdat‘) are included in the anal- ysis. Categorical attributes cover attack categories (‘attack_cat‘) and a binary label (‘label‘) indicating whether the traffic is nor mal or potentially malicious.

The dataset was sourced from Kaggle and is part of ADFA-LD, a dataset developed by researchers at the Australian Defence Force Academy to study modern attack structures and strategies. ADFA- LD employs system-call-based Host Intrusion Detection System (HIDS) analysis using data from Ubuntu Linux version 11.04. This dataset is valuable for evaluating intrusion detection systems (IDS) and comparing signature-based IDS (SIDS) and anomaly-based IDS (AIDS). The dataset includes training and validation subsets derived from normal activities such as web browsing and document production on the host system.

RESULTS ANALYSIS AND DISCUSSION

In this section the findings of experiments related to network in- trusion detection using various approaches such as LSTM, Gradi- ent Boosting, GRU, Random Forest, and RNN on the UNSW-NB15 dataset. Each method begins with data preparation, including en- coding categorical features using LabelEncoder and normalizing data with StandardScaler before model training.

The experiment with LSTM yielded promising results. The LSTM model achieved a highest validation accuracy of 99.34% after being trained for 10 epochs with a batch size of 32. This demonstrates the model’s ability to generalize well from training to validation data and accurately classify network intrusions. Evaluation on the test data also showed consistently high performance, with excellent ac- curacy, precision, recall, and F1-score metrics. However, it requires longer training time compared to some other methods.

Meanwhile, Gradient Boosting showed nearly perfect accuracy of 99.99% on the test data, indicating high reliability in classifying network intrusions. However, careful parameter tuning is necessary to achieve optimal performance. Despite slightly lower accuracy compared to LSTM and Gradient Boosting, GRU remains effective in network intrusion detection with faster training process.

Random Forest achieved an accuracy of 90.28% on the test data, with some variations in precision and recall depending on the type of attack detected. Its strength lies in handling data imbalance and processing categorical features without requiring complex prepro- cessing steps.

Discussion of these results provides deep insights into the strengths and weaknesses of each approach in the context of network intru- sion detection. LSTM and Gradient Boosting stand out with nearly perfect accuracy, demonstrating superior ability to accurately clas- sify attacks. The choice of intrusion detection method should be carefully considered based on application specifications, available resources, and the need for training time and model complexity. These experimental results make a significant contribution to the development of more effective and reliable intrusion detection tech- niques in the face of increasingly complex and dynamic information security challenges.

CONCLUSION

In the current era of technological advancement, breakthrough in- novations, state-of-the-art hardware development, increased com- puting capacity through cloud computing and supercomputers, and collaboration between research institutions and industries are the primary driving forces. However, this progress is also accompanied by a rise in malware attacks due to increasingly complex systems and the adoption of new technologies that provide vulnerabilities for attackers. High processing capabilities are exploited to develop undetectable malware, while collaboration among cybercriminals is on the rise. Therefore, effective cybersecurity protection and ef- forts are becoming increasingly crucial. In this study, we compared GRU, Random Forest, Gradient Boosting, LSTM, and RNN models for malware attack detection, with experimental results showing that the Gradient Boosting model achieved the highest accuracy of 99.98%.

Considering the findings of this research, it can be concluded that the use of GRU, Random Forest, Gradient Boosting, LSTM, and RNN methods in malware attack detection has the potential to enhance detection effectiveness. The experimental results indicate that the Gradient Boosting model is an excellent choice with the highest accuracy, while other models also demonstrate good capabilities in distinguishing between malicious and benign network activities. By continuing to develop detection strategies and responses to threats, it is expected that ML-based malware detection technology will ad- vance further and make a significant contribution to strengthening the cybersecurity framework.

Thus, the conclusion of this study highlights the importance of using ML methods in malware attack detection, emphasizing the need for the development of proactive detection strategies and responsive threat mitigation in the face of evolving security threats in the current cyber world.

DAFTAR PUSTAKA

Adeyeye, O., Akanbi, I., Emeteveke, I., & Emehin, O. (2024). Leveraging Secured Ai-Driven Data Analytics For Cybersecurity: Safeguarding Information And Enhancing Threat Detection. *International Journal of Research Publication and Reviews*. https://doi.org/10.55248/gengpi.5.1024.2911.

Akhtar, M., & Feng, T. (2023). Evaluation of Machine Learning Algorithms for Malware Detection. *Sensors (Basel, Switzerland)*, 23. https://doi.org/10.3390/s23020946.

Alomari, E., Nuiaa, R., Alyasseri, Z., Mohammed, H., Sani, N., Esa, M., & Musawi, B. (2023). Malware Detection Using Deep Learning and Correlation-Based Feature Selection. *Symmetry*, 15, 123. https://doi.org/10.3390/sym15010123.

Aslan, Ö., & Samet, R. (2020). A Comprehensive Review on Malware Detection Approaches. *IEEE Access*, 8, 6249-6271. https://doi.org/10.1109/ACCESS.2019.2963724.

Azeem, M., Khan, D., Iftikhar, S., Bawazeer, S., & Alzahrani, M. (2023). Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches. *Heliyon*, 10. https://doi.org/10.1016/j.heliyon.2023.e23574.

Callens, A., Morichon, D., Abadie, S., Delpey, M., & Liquet, B. (2020). Using Random forest and Gradient boosting trees to improve wave forecast at a specific location. *Applied Ocean Research*, 104, 102339. https://doi.org/10.1016/j.apor.2020.102339.

Cao, B., Li, C., Song, Y., Qin, Y., & Chen, C. (2022). Network Intrusion Detection Model Based on CNN and GRU. *Applied Sciences*. https://doi.org/10.3390/app12094184.

Falowo, O., Ozer, M., Li, C., & Abdo, J. (2024). Evolving Malware and DDoS Attacks: Decadal Longitudinal Study. *IEEE Access*, 12, 39221-39237. https://doi.org/10.1109/ACCESS.2024.3376682.

Shaukat, K., Luo, S., & Varadharajan, V. (2023). A novel deep learning-based approach for malware detection. *Eng. Appl. Artif. Intell.*, 122, 106030. https://doi.org/10.1016/j.engappai.2023.106030.

Varma, S. (2024). AI-Enhanced Cloud Security: Proactive Threat Detection and Response Mechanisms. *International Journal For Multidisciplinary Research*. https://doi.org/10.36948/ijfmr.2024.v06i06.31587.

Vinayakumar, R., Alazab, M., Member, I., Poornachandran, P., & Venkatraman, A. (2019). Robust Intelligent Malware Detection Using Deep Learning. *IEEE Access*, 7, 46717-46738. https://doi.org/10.1109/ACCESS.2019.2906934.

Xu, C., Shen, J., Du, X., & Zhang, F. (2018). An Intrusion Detection System Using a Deep Neural Network With Gated Recurrent Units. *IEEE Access*, 6, 48697-48707. https://doi.org/10.1109/ACCESS.2018.2867564.