engineering proceedings

MDPI

Proceeding Paper

Unveiling Cyber Threats: An In-Depth Study on Data Mining Techniques for Exploit Attack Detection †

Abdallah S. Hyassat [1,2,*], Raneem E. Abu Zayed [2], Eman A. Al Khateeb [2], Ahmad Shalaldeh [1], Mahmoud M. Abdelhamied [1] and Iyas Qaddara [3]

[1] Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman 19111, Jordan; a.shalaldeh@ammanu.edu.jo (A.S.); m.abdelhamied@ammanu.edu.jo (M.M.A.)
[2] Department of Computer Science, University of Jordan, Amman 11942, Jordan; rny9210231@ju.edu.jo (R.E.A.Z.); aym9210232@ju.edu.jo (E.A.A.K.)
[3] Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman 19111, Jordan; i.qaddara@ammanu.edu.jo
* Correspondence: a.hyassat@ammanu.edu.jo
† Presented at the International Conference on Electronics, Engineering Physics and Earth Science (EEPES 2025), Alexandroupolis, Greece, 18–20 June 2025.

Abstract

The number of people and applications using the internet has increased substantially in recent years. The increased use of the internet has also resulted in various security issues. As the volume of data increases, cyber-attacks become increasingly sophisticated, exploiting vulnerabilities in network structures. The incorporation of modern technologies, particularly data mining, emerges as an essential method for analyzing huge amounts of data in real time, enabling the proactive detection of anomalies and potential security breaches. This research seeks to identify the most robust machine learning model for exploit detection. It applies five feature selection techniques and eight classification models to the UNSW-NB15 dataset. A comprehensive evaluation is conducted based on classification accuracy, computational efficiency, and execution time. The results demonstrate the efficiency of the Decision Tree model using Random Forest for feature selection in the real-time detection of exploit attacks, exhibiting an accuracy of 87.9%, along with a very short training (0.96 s) and testing time (0.29 ms/record).

Keywords: data mining techniques; exploit attack; machine learning; intrusion detection systems; cybersecurity analytics

check for updates

Academic Editor: Grigor Mihaylov

Published: 25 August 2025

Citation: Hyassat, A.S.; Abu Zayed, R.E.; Al Khateeb, E.A.; Shalaldeh, A.; Abdelhamied, M.M.; Qaddara, I. Unveiling Cyber Threats: An In-Depth Study on Data Mining Techniques for Exploit Attack Detection. *Eng. Proc.* **2025**, *104*, 28. https://doi.org/10.3390/engproc2025104028

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Data mining aims to automate the process of extracting meaningful patterns and insights from datasets [1]; this is a powerful analytical technique. Data mining plays a pivotal role in many different sectors, including education [2], medical [3], and cybersecurity [4]. Cybersecurity is the practice of protecting networks, programs, and systems from any attack or threat. Data is growing exponentially, and cyber threats are constantly evolving. The increase in cyber-attacks is primarily caused by the design of computer systems and communication networks. Vulnerabilities and misconfigurations in networks, hardware, and software create opportunities for hackers to exploit and carry out attacks [5]. Therefore, the merger of data mining technologies is crucial for analyzing massive datasets, which enables real-time detection of anomalies and potential security breaches. Data mining has

emerged as a crucial tool for efficiently analyzing and interpreting complex information. By unraveling hidden correlations and identifying patterns indicative of potential security threats, data mining offers a proactive approach to fortifying digital defenses. This research aims to study different data mining techniques for exploit attack detection by applying eight supervised learning algorithms—namely, Decision Tree (DT), Random Forest (RF), XGBoost, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Multi-Layer Perceptron (MLP), and AdaBoost [6,7]—alongside five feature selection methods, including Recursive Feature Elimination (RFE), SelectKBest, Logistic-Regression-based ranking, Random-Forest-based importance, and the Genetic Algorithm (GA) [8,9]. In this paper, the UNSW-NB15 dataset [10] is used. This dataset consists of attributes related to network traffic, including IP addresses, the type of service, and the protocol used.

The main contributions of this paper are as follows:

- Introduce a novel contribution by addressing the literature gap where previous works did not adequately consider exploit attacks, despite their significance.
- Analyze the detection of exploit attacks using different data mining techniques
- Evaluate the performance of different models and different feature selection methods.

2. Literature Review

There has been a significant amount of research on intrusion detection. Many of these studies have focused on the use of machine learning techniques to classify traffic as either normal or anomalous.

A novel model for network intrusion detection systems (NIDS) is proposed in [11]. They integrate Long Short-Term Memory (LSTM) networks with an attention mechanism in order to enhance the binary classification of network traffic data. The research focuses on the drawbacks of the existing machine learning and deep learning models, including SVM and KNN, that have issues with feature selection and low accuracy. To this end, the authors work with the vast UNSW-NB15 dataset and assert the need to incorporate both spatial and temporal features of the network traffic to enhance the model's capacity to differentiate between normal and malicious traffic.

A hybrid deep learning (DL) model for intrusion detection was proposed in [12]. To reduce the dimensionality of the data, radial basis function-based support vector regression (RBFSVR) is used. The hybrid model integrates VGG 19 and 2D-CNN. VGG19 is used for feature extraction while 2D CNN is utilized for classification. The hybrid model is used in the fog layer for real-time threat recognition. Considering both binary and multi-attack classification for intrusion, Mohiuddin et al. in [13] built a model based on meta-heuristic algorithms using the modified wrapper-based whale sine–cosine algorithm (MWWSCA) for selecting the relevant features. For classification, a weighted Gradient Boosting (XGBoost) algorithm was proposed. The model achieved an accuracy of 99% for binary classification and 99% for multi-attack classification.

As part of the effort to enhance intrusion detection in fog computing, Mohamed and Ismael proposed a method for intrusion detection based on the back propagation neural network detailed in [14]. For feature selection, standard deviation is used to select the most relevant features. The Genetic Algorithm is applied to efficiently optimize the weights and the biases of the neurons of this network. Similarly, Barhoushet al. [15] focused on enhancing feature selection using an improved metaheuristic algorithm. An improved SSA algorithm was introduced. During the initialization phase, the Opposition-Based Learning (OBL) method was used to enhance population diversity. Secondly, to improve exploration, the Elite Opposition-Based Learning (EOBL) and a Variable Neighborhood Search (VNS) method were used. Finally, to discretize the continuous candidate solutions, the Sigmoid binary transform function was utilized.

Kumar et al. in [16] introduced an Intrusion Detection Systems (IDS) specifically designed for classifying attacks on the Internet of Things (IoT) networks. The study employs multiple machine learning (ML) models to safeguard IoT networks effectively. The study employs a combination-based grouping method for optimal class feature selection, followed by a filter-based feature selection technique to further optimize the chosen features. Notably, the model achieves an accuracy of 98.72% using the Random Forest classifier.

A multi-classification model was proposed in [17], using a Multilayer Perceptron Neural Network (MLP). Information gain (IG) and Random Forest (RF) are used for dimensionality reduction. Recursive feature elimination (RFE) is employed to boost feature reduction. The features are reduced from 23 out of 42 features. The model achieved an accuracy of 84%24. In [18], Srivastava employed a novel Intrusion detection model. The model employs Wasserstein Conditional Generative Adversarial Network—Gradient Penalty (WCGAN-GP) for oversampling and the Genetic Algorithm (GA) for feature selection to achieve the best feature vector for the classification issue. This research further employs a novel fitness function for the Genetic Algorithm's convergence. To assess the effectiveness of various machine learning models, a comprehensive experimental investigation using the NSL-KDD and UNSW-NB15 datasets is conducted in this work. According to observations, XGBoost achieved the best accuracy, at 95.54%.

3. Methodology

In this study, the UNSW-NB15 dataset is employed to assess the effectiveness of machine learning models in detecting exploit attacks. The UNSW-NB15 dataset is one of the most commonly used benchmark datasets. It contains a broad spectrum of network threats such as exploit attacks [10]. Figure 1 illustrates the proposed methodology used in this study, which encompasses several key stages: data pre-processing, feature selection, training/testing split, model development, and evaluation. A total of five feature selection techniques and eight classification algorithms were applied to identify the most effective model for exploit detection.

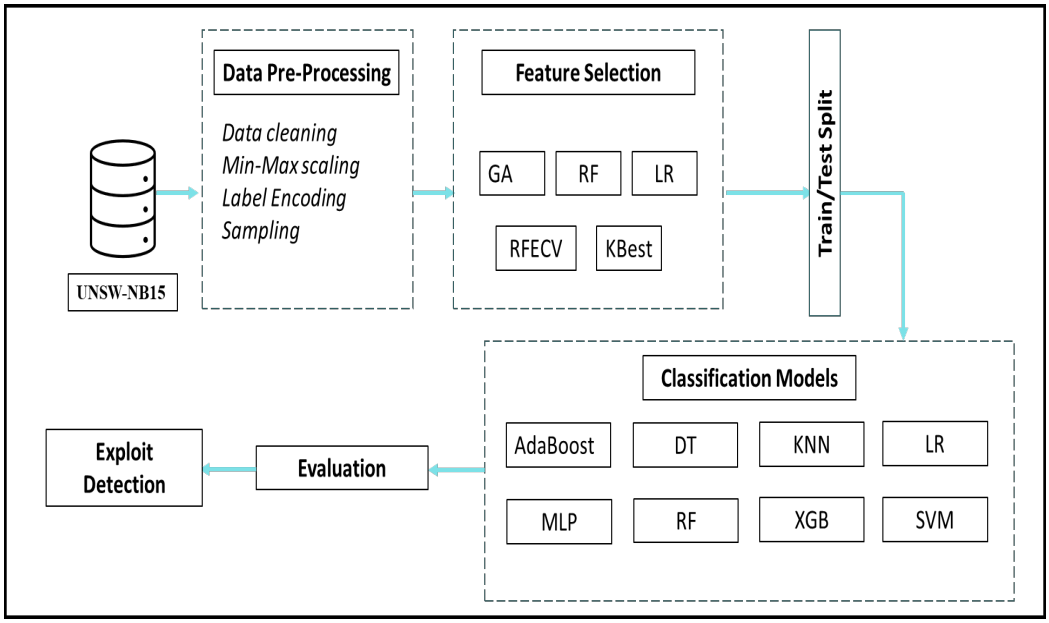


Figure 1. The proposed methodology.

3.1. Data Preprocessing

Data preprocessing is an essential phase in the methodology since it verifies whether the UNSW-NB15 dataset is formatted correctly for the models and removes any irrelevant

or corrupted data before modeling. The phase undergoes four steps: data integration, data cleaning, data sampling, normalization, and label encoding. Firstly, in data integration, the four original files of the dataset are combined into a single file containing all records.

Data cleaning: This refers to the removal of invalid or missing data. First, extraneous whitespace is removed. Second invalid or missing data (invalid data may include records with incorrect or inconsistent values) are removed. Here, there are three columns that have missing values. These records should be removed for the sake of feeding a machine learning model with consistent and clean data that has been tested and validated.

Data Sampling: We introduced a targeted data sampling approach that focuses our analysis specifically on exploit attacks within the UNSW-NB15 dataset. Since the dataset has various types of attacks, and our primary interest lies in exploits, we treated exploit instances as normal records. There were 44,525 instances of exploits within the dataset.

The non-exploit instances were included by sampling from other types of attacks to create a balanced dataset for our analysis. The goal was to ensure an equivalent representation of non-exploit instances compared to the number of exploit instances. All types of attacks in the dataset were considered, and random instances were selected to match the number of exploits. Certain attack categories consisted of a limited number of instances, so no sampling was necessary for these categories. For instance, the "worms" attack category had 174 instances, and these instances were included without the need for additional sampling.

Data Normalization: Data normalization ensures that the features have the same scale and range, which might improve the model's performance. The UNSW-NB15 dataset includes certain numerical information, such as packet sizes and time intervals. The Min–Max scaler is used to normalize the dataset's numerical features.

Label Encoding: The variable assigns a standard numeric value representing the categorization of a variable. The UNSW-NB15 dataset contains categorical features, including service, state, and protocol type. These features need to be converted into numerical forms before they can be input into a machine learning model.

3.2. Feature Selection

In this step, the most relevant features from the dataset are identified and selected to improve model efficiency and predictive performance. This plays a major role in reducing dimensionality, minimizing overfitting, and enhancing interpretability [19,20]. This approach applies five distinct feature selection techniques, each producing a unique subset of features. All eight classification algorithms are trained and evaluated independently on each selected subset, allowing for a comprehensive comparison of how different feature spaces influence model performance.

3.3. Model Development, Training, and Evaluation

A crucial phase in our methodology involves the development and training of machine learning models to effectively detect exploit attacks. This process includes selecting appropriate models and training them through supervised learning, utilizing the preprocessed UNSW-NB15 dataset. Our approach aims to create models that are capable of accurately identifying exploit attacks. We chose several well-known machine learning methods widely utilized in related tasks, including Random Forest, XGBoost, KNN, SVM, AdaBoost, Decision Tree, Logistic Regression, and MLP Classifier [21,22]. Except for the XGBoost algorithm, implemented using the XGBoost library, the remaining algorithms were developed using the python version 3.12 and Sklearn library version 1.7.1 with default parameters.

Furthermore, each model underwent training and testing using the feature results of five feature selection models.

We adjusted each machine learning model during the training process so that it could recognize and distinguish exploit attacks from network activity. To reduce classification mistakes and improve the model's capacity to correctly categorize instances of network traffic, fine-tuning was performed. We used the testing set to evaluate the models' performance after training. Each model predicted the class label independently for a given instance of network traffic.

4. Experiments and Results

As we evaluated multiple machine learning (ML) models for exploit detection using the UNSW-NB15 dataset, we will discuss the details of our experiments and the results in this section. First, the dataset and testing environment are described. Second, the evaluation metrics used to assess the model's performance are described. Finally, the last part will explain the results of our research.

4.1. Dataset

In this research, we employed the UNSW-NB15 dataset. This dataset comprises network traffic data that can be used to build and assess a network intrusion detection system. The dataset was constructed by researchers at the University of New South Wales in 2015, and it has become one of the most popular and significant benchmark datasets for the comparative evaluation of various kinds of intrusion detection techniques. There are about two million network connections in the dataset; each connection is categorized as either a subclass of one of nine types of attacks, or normal. The features of this dataset include the source and destination IP addresses and ports, the size of the packet, and the time the packet was created, among other features. The other category of service features in the set includes basic forms of the service such as HTTP, FTP, TELNET as well as specific service features such as HTTP user agent and Web browser referrer. It is noteworthy that the UNSW-NB15 dataset is constantly used by researchers and is frequently cited in scientific production. Table 1 shows the distribution of the dataset's records.

Table 1. Distribution of dataset records.

Category	Count
Normal	2,218,761
Fuzzers	24,246
Reconnaissance	13,987
Shellcode	1511
Analysis	2677
Backdoors	2329
DoS	16,353
Exploits	44,525
Generic	215,481
Worms	174
Total	2,540,044

4.2. Evaluation Measures

We calculated several performance measures to evaluate the models' efficacy in identifying exploit attacks. Separate analysis of each model revealed important details about its advantages and disadvantages, as well as how well it could detect exploit attacks, among other kinds of network threats.

Evaluation metrics are an important tool for assessing the efficiency of machine learning models. The evaluation is based on accuracy, precision, recall, and the F1 score. These measurements are described as follows [23,24]:

$$Accuracy = \frac{\text{True Positive} + \text{True Negative}}{\text{All instances}} \quad (1)$$

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{false Positive}} \quad (2)$$

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{false Negative}} \quad (3)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

4.3. Experiments

It is common practice in machine learning to ensure that the model is able to generalize well to new, unseen data. One popular method for carrying this out is to use a fraction of the dataset for training, and the remaining fraction for testing. In this case, the fraction of the dataset used for training is set to 80%, while the remaining 20% is used for testing. Each machine learning model used in this study was then trained and evaluated using testing data to assess its performance on unseen samples.

As discussed earlier, eight different classification models (Random Forest, XGBoost, KNN, SVM, AdaBoost, Decision Tree, Logistic Regression, MLP Classifier) were tested with each feature selection technique on the UNSW-NB15 dataset. The performance results are depicted in the following tables and figures.

Table 2 shows the performance comparison of classification models using the K-best feature selection technique based on all evaluation metrics. The DT model performed best in terms of accuracy (0.876), with a good tradeoff between precision and recall. It is also notable that RF exhibited high levels of performance with almost equal accuracy scores. Likewise, when we used both Logistic Regression (Table 3) and Random Forest (Table 4) as feature selection techniques, DT and RF performed best in terms of accuracy (≈88%).

Table 2. Comparison of classification models' performance using the K-best feature selection technique.

Model	Accuracy	Recall	Precision	F1 Score	AUC
AdaBoost	0.796	0.916	0.739	0.818	0.796
Decision Tree	**0.876**	0.934	**0.837**	0.883	**0.876**
KNN	0.775	0.828	0.748	0.786	0.774
Logistic Regression	0.673	0.473	0.790	0.592	0.674
MLP Classifier	0.710	0.814	0.673	0.733	0.710
Random Forest	0.875	**0.956**	0.824	**0.885**	0.875
SVM	0.667	0.861	0.621	0.721	0.666
XGBoost	0.844	0.955	0.782	0.860	0.844

Table 3. Comparison of classification models' performance using the logistic regression feature selection technique.

Model	Accuracy	Recall	Precision	F1 Score	AUC
AdaBoost	0.7906	0.8635	0.7541	0.8051	0.7904
Decision Tree	**0.8791**	0.9081	**0.8587**	0.8827	**0.8790**
KNN	0.8094	0.8394	0.7925	0.8153	0.8093
Logistic Regression	0.7425	0.6920	0.7707	0.7292	0.7426
MLP Classifier	0.7913	0.8684	0.7531	0.8064	0.7911
Random Forest	0.8790	0.9266	0.8465	**0.8847**	0.8789
SVM	0.6985	0.6242	0.7342	0.6747	0.6986
XGBoost	0.8433	**0.9402**	0.7880	0.8574	0.8431

Table 4. Comparison of classification models' performance using the Random Forest feature selection technique.

Model	Accuracy	Recall	Precision	F1 Score	AUC
AdaBoost	0.7973	0.9220	0.7384	0.8201	0.7970
Decision Tree	**0.8791**	0.9081	**0.8587**	0.8827	**0.8790**
KNN	0.7785	0.8312	0.7526	0.7899	0.7784
Logistic Regression	0.6759	0.4905	0.7813	0.6026	0.6763
MLP Classifier	0.6872	0.7147	0.6909	0.6945	0.6871
Random Forest	0.8790	0.9261	0.8468	**0.8847**	0.8789
SVM	0.6664	0.8606	0.6205	0.7211	0.6660
XGBoost	0.8467	**0.9460**	0.7896	0.8608	0.8465

In Tables 5 and 6, the Decision Tree and RF models are shown to exhibit high levels of performance across all metrics, with the same accuracy score (0.8665 with RFECV, and 0.8704 with GA, respectively). They both have high recall, implying that they can identify almost all positive cases, and an excellent balance of precision and recall, as indicated by the F1 score. We benchmarked each classifier under all feature selection techniques, and the accuracy comparison is shown in Figure 2.

Table 5. Comparison of classification models' performance using the RFECV feature selection technique.

Model	Accuracy	Recall	Precision	F1 Score	AUC
AdaBoost	0.7922	0.9249	0.7315	0.8169	0.7920
Decision Tree	**0.8665**	0.9967	0.7912	0.8821	**0.8663**
KNN	0.7799	0.6723	**0.8577**	0.7537	0.7801
Logistic Regression	0.5010	**1.0000**	0.5010	0.6676	0.5000
MLP Classifier	0.5519	0.6139	0.5597	0.5450	0.5518
Random Forest	**0.8665**	0.9980	0.7906	**0.8823**	**0.8663**
SVM	0.6402	0.9445	0.5877	0.7245	0.6395
XGBoost	0.8325	0.9809	0.7568	0.8544	0.8322

Table 6. Comparison of classification models' performance using the Genetic Algorithm feature selection technique.

Model	Accuracy	Recall	Precision	F1 Score	AUC
AdaBoost	0.7955	0.9181	0.7379	0.8182	0.7953
Decision Tree	**0.8704**	0.9619	**0.8135**	0.8815	**0.8702**
KNN	0.8068	0.8233	0.7977	0.8103	0.8068
Logistic Regression	0.6457	0.4792	0.7200	0.5754	0.6461
MLP Classifier	0.7152	0.9239	0.6591	0.7663	0.7148
Random Forest	0.8704	**0.9642**	0.8123	**0.8817**	0.8702
SVM	0.6400	0.9437	0.5877	0.7243	0.6394
XGBoost	0.8408	0.9517	0.7794	0.8570	0.8406

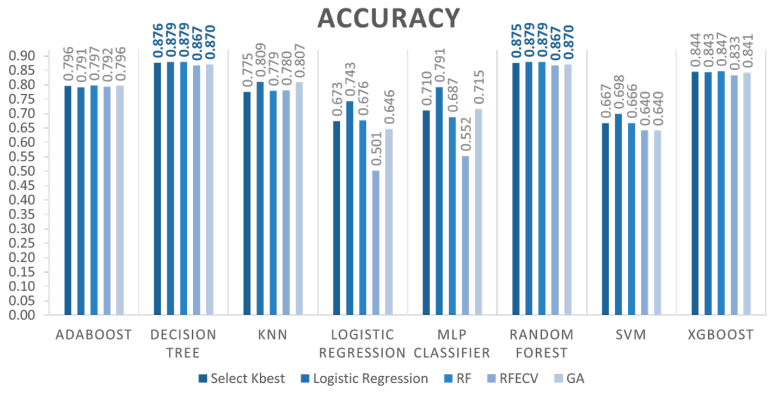


Figure 2. Comparison of classification models' accuracy using all feature selection techniques.

Table 7 views the comparison between the best classification models using all performance metrics.

Table 7. Comparison of the classification models' performance using all feature selection techniques.

Feature Selection Technique	Evaluation Metric	Best Classification Model	Best Value
SelectKBest	Accuracy	Decision Tree and Random Forest	0.8759
Logistic Regression		Decision Tree and Random Forest	0.8791
RF		Decision Tree and Random Forest	0.8791
RFECV		Decision Tree and Random Forest	0.8665
GA		Decision Tree and Random Forest	0.8704
SelectKBest	Recall	Random Forest	0.9560
Logistic Regression		XGBoost	0.9402
RF		XGBoost	0.9460
RFECV		Logistic Regression	1.0
GA		Random Forest	0.9642
SelectKBest	Precision	Decision Tree	0.8374
Logistic Regression		Decision Tree	0.8587
RF		Decision Tree	0.8587
RFECV		KNN	0.8577
GA		Decision Tree	0.8135
SelectKBest	F1 Score	Random Forest	0.8850
Logistic Regression		Random Forest	0.8847
RF		Random Forest	0.8847
RFECV		Random Forest	0.8823
GA		Random Forest	0.8817

Since time is critical in security threat detection, we have also compared the models' performance according to training and testing time. Figures 3 and 4 depict these comparisons.

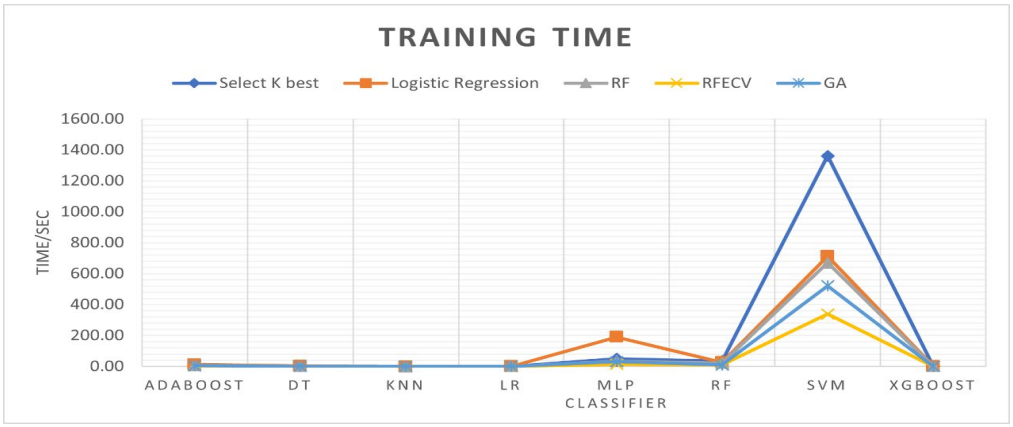


Figure 3. Models' training time comparison.

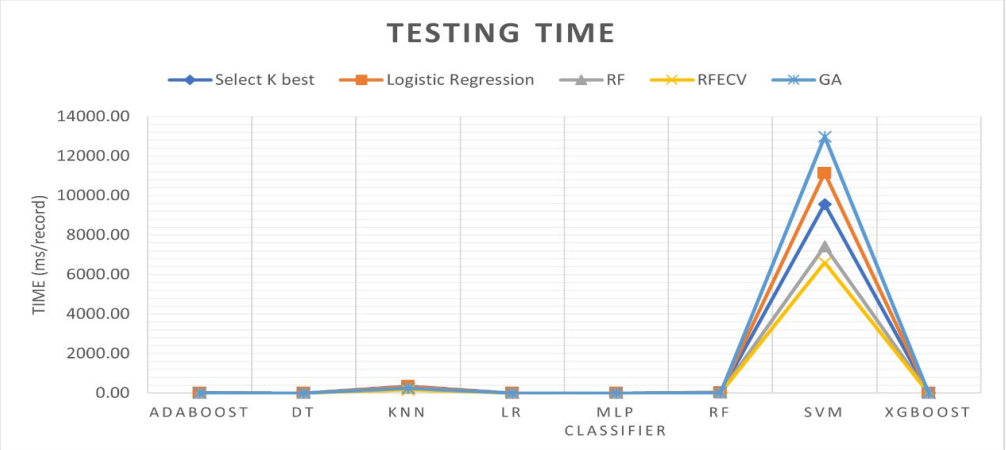


Figure 4. Models' testing time comparison.

5. Conclusions

This study focuses on how data mining boosts cybersecurity by helping detect exploit attacks. To perform a detailed comparison, we used the UNSW-NB15 dataset and applied eight known machine learning methods together with five methods of feature selection. Comparing all methods, Decision Tree combined with Random Forest performed best, having high accuracy (87.9%), high precision (85.9%), and an outstanding F1-score (88.3%), along with low training and test resources. The results confirm the effectiveness and practical readiness of the proposed method for real-time security systems.

It is also highlighted in the findings that choosing appropriate features helps to make the model more accurate and efficient. The use of wrappers and embedded approaches allowed for better detection without the problem of overfitting.

Further research in this field can take many forms. More detailed classification could focus on identifying various kinds of exploit attacks. One way to achieve this is to develop the method with the aim of improving detection accuracy, which could be further improved by combining deep learning models. If the approach is applied in real streaming settings, its reliability can be evaluated during live network activity.

Author Contributions: Conceptualization, A.S.H. and R.E.A.Z.; Methodology, A.S.H. and E.A.A.K.; Software, A.S. and I.Q.; Validation, A.S.H., E.A.A.K. and M.M.A.; Formal analysis, A.S.H. and E.A.A.K.; Investigation, A.S.H., E.A.A.K. and R.E.A.Z.; Resources, I.Q. and M.M.A.; Data curation, A.S.H. and A.S.; Writing—original draft preparation, A.S., E.A.A.K. and R.E.A.Z.; Writing—review and editing, E.A.A.K. and M.M.A.; Visualization, E.A.A.K. and R.E.A.Z.; Supervision, I.Q.; Project administration, A.S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this study is publicly available. The UNSW-NB15 dataset can be accessed at: https://research.unsw.edu.au/projects/unsw-nb15-dataset. (accessed on 20 August 2025).

Acknowledgments: The authors would like to express their sincere gratitude to Al-Ahliyya Amman University for its continuous support and encouragement of scientific research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zaki, M.J.; Meira, W. *Data Mining and Analysis: Fundamental Concepts and Algorithms*; Cambridge University Press: Cambridge, UK, 2014.
2. Feng, G.; Fan, M. Research on Learning Behavior Patterns from the Perspective of Educational Data Mining: Evaluation, Prediction and Visualization. *Expert. Syst. Appl.* **2024**, *237*, 121555. [CrossRef]
3. Mansouri, S. Application of Neural Networks in the Medical Field. Journal of Wireless Mobile Networks. *Ubiquitous Comput. Dependable Appl. (JoWUA)* **2025**, *14*, 69–81. [CrossRef]
4. Alsaaidah, A.; Almomani, O.; Abu-Shareha, A.A.; Abualhaj, M.M.; Achuthan, A. ARP Spoofing Attack Detection Model in IoT Network Using Machine Learning: Complexity vs. Accuracy. *J. Appl. Data Sci.* **2024**, *5*, 1850–1860. [CrossRef]
5. Almaiah, M.A.; Saqr, L.M.; Al-Rawwash, L.A.; Altellawi, L.A.; Al-Ali, R.; Almomani, O. Classification of Cybersecurity Threats, Vulnerabilities and Countermeasures in Database Systems. Computers. *Mater. Contin.* **2024**, *81*, 3189–3220. [CrossRef]
6. Almomani, O.; Alsaaidah, A.; Shareha, A.A.A.; Alzaqebah, A.; Almomani, M. Performance Evaluation of Machine Learning Classifiers for Predicting Denial-of-Service Attack in Internet of Things. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 263–271. [CrossRef]
7. Abualhaj, M.M.; Al-Khatib, S.; Al Shafi, N.; Qaddara, I.; Hyassat, A. Utilizing Gray Wolf Optimization Algorithm in Malware Forensic Investigation. *J. Comput. Cogn. Eng.* **2025**, 1–12. [CrossRef]
8. Al-Amiedy, T.A.; Anbar, M.; Belaton, B.; Bahashwan, A.A.; Abualhaj, M.M. Towards a Lightweight Detection System Leveraging Ranking Techniques with Wrapper Feature Selection Algorithm for Selective Forwarding Attacks in Low Power and Lossy Networks of IoTs. In Proceedings of the 2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA), Sana'a, Yemen, 6–7 August 2024; pp. 1–17. [CrossRef]
9. Chen, X.; Jeong, J.C. Enhanced Recursive Feature Elimination. In Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA 2007), Cincinnati, OH, USA, 13–15 December 2007; pp. 429–435. [CrossRef]
10. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
11. Alsharaiah, M.; Abualhaj, M.; Baniata, L.; Al-saaidah, A.; Kharma, Q.; Al-Zyoud, M. An Innovative Network Intrusion Detection System (NIDS): Hierarchical Deep Learning Model Based on Unsw-Nb15 Dataset. *Int. J. Data Netw. Sci.* **2024**, *8*, 709–722. [CrossRef]
12. Binbusayyis, A. Hybrid VGG19 and 2D-CNN for Intrusion Detection in the FOG-Cloud Environment. *Expert. Syst. Appl.* **2024**, *238*, 121758. [CrossRef]
13. Mohiuddin, G.; Lin, Z.; Zheng, J.; Wu, J.; Li, W.; Fang, Y.; Wang, S.; Chen, J.; Zeng, X. Intrusion Detection Using Hybridized Meta-Heuristic Techniques with Weighted XGBoost Classifier. *Expert. Syst. Appl.* **2023**, *232*, 120596. [CrossRef]
14. Mohamed, D.; Ismael, O. Enhancement of an IoT Hybrid Intrusion Detection System Based on Fog-to-Cloud Computing. *J. Cloud Comput.* **2023**, *12*, 41. [CrossRef]
15. Barhoush, M.; Abed-alguni, B.H.; Al-qudah, N.E.A. Improved Discrete Salp Swarm Algorithm Using Exploration and Exploitation Techniques for Feature Selection in Intrusion Detection Systems. *J. Supercomput.* **2023**, *79*, 21265–21309. [CrossRef]
16. Kumar, V.; Kumar, V.; Singh, N.; Kumar, R. Enhancing Intrusion Detection System Performance to Detect Attacks on Edge of Things. *SN Comput. Sci.* **2023**, *4*, 802. [CrossRef]
17. Yin, Y.; Jang-Jaccard, J.; Xu, W.; Singh, A.; Zhu, J.; Sabrina, F.; Kwak, J. IGRF-RFE: A Hybrid Feature Selection Method for MLP-Based Network Intrusion Detection on UNSW-NB15 Dataset. *J. Big Data* **2023**, *10*, 15. [CrossRef]
18. Srivastava, A.; Sinha, D.; Kumar, V. WCGAN-GP Based Synthetic Attack Data Generation with GA Based Feature Selection for IDS. *Comput. Secur.* **2023**, *134*, 103432. [CrossRef]
19. Abualhaj, M.M.; Al-Zyoud, M.; Alsaaidah, A.; Abu-Shareha, A.; Al-Khatib, S. Enhancing Malware Detection through Self-Union Feature Selection Using Firefly Algorithm with Random Forest Classification. *Int. J. Intell. Eng. Syst.* **2024**, *17*, 376–389. [CrossRef]
20. Saeed, M.H.; Hama, J.I. Cardiac Disease Prediction Using AI Algorithms with SelectKBest. *Med. Biol. Eng. Comput.* **2023**, *61*, 3397–3408. [CrossRef] [PubMed]
21. Ramakrishnan, K.; Balakrishnan, V.; Wong, H.Y.; Tay, S.H.; Soo, K.L.; Kiew, W.K. Face Mask Wearing Classification Using Machine Learning. *Eng. Proc.* **2023**, *41*, 13. [CrossRef]
22. Kazolis, D.; Fotakis, C.D.; Tramantzas, K. Comparison of Functionality and Evaluation of Results in Different Prediction Models. *Eng. Proc.* **2024**, *70*, 31. [CrossRef]

23. Abualhaj, M.M.; Al-Khatib, S.; Hiari, M.O.; Shambour, Q.Y. Enhancing Spam Detection Using Hybrid of Harris Hawks and Firefly Optimization Algorithms. *J. Soft Comput. Data Min.* **2024**, *5*, 161–174. [CrossRef]
24. Mukasheva, A.; Koishiyeva, D.; Sergazin, G.; Sydybayeva, M.; Mukhammejanova, D.; Seidazimov, S. Modification of U-Net with Pre-Trained ResNet-50 and Atrous Block for Polyp Segmentation: Model TASPP-UNet. *Eng. Proc.* **2024**, *70*, 16. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.