Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 10, No. 1, February, Pp. 47-52

Integrating ensemble learning and information gain for malware detection based on static and dynamic features

Ramadhan Rakhmat Sani[1], Fauzi Adi Rafrastara*[1], Wildanil Ghozi[1]
Department of Information System, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia[1]

Article Info
Keywords:
Malware Detection, Ensemble Learning, Information Gain, Gradient Boosting, Static and Dynamic Features

Article history:
Received: July 07, 2024
Accepted: October 03, 2024
Published: February 01, 2025

Cite:
R. R. Sani, F. A. Rafrastara, and W. Ghozi, "Integrating Ensemble Learning and Information Gain for Malware Detection based on Static and Dynamic Features", *KINETIK*, vol. 10, no. 1, Feb. 2025.
https://doi.org/10.22219/kinetik.v10i1.2051

*Corresponding author.
Fauzi Adi Rafrastara
E-mail address:
fauziadi@dsn.dinus.ac.id

Abstract
The rapid advancement of malware poses a significant threat to devices, like personal computers and mobile phones. One of the most serious threats commonly faced is malicious software, including viruses, worms, trojan horses, and ransomware. Conventional antivirus software is becoming ineffective against the ever-evolving nature of malware, which can now take on various forms like polymorphic, metamorphic, and oligomorphic variants. These advanced malware types can not only replicate and distribute themselves, but also create unique fingerprints for each offspring. To address this challenge, a new generation of antivirus software based on machine learning is needed. This intelligent approach can detect malware based on its behavior, rather than relying on outdated fingerprint-based methods. This study explored the integration of machine learning models for malware detection using various ensemble algorithms and feature selection techniques. The study compared three ensemble algorithms: Gradient Boosting, Random Forest, and AdaBoost. It used Information Gain for feature selection, analyzing 21 features. Additionally, the study employed a public dataset called 'Malware Static and Dynamic Features VxHeaven and VirusTotal Data Set', which encompasses both static and dynamic malware features. The results demonstrate that the Gradient Boosting algorithm combined with Information Gain feature selection achieved the highest performance, reaching an accuracy and F1-Score of 99.2%.

1. Introduction

Malware or malicious software, a term encompassing harmful software, has evolved significantly since its inception in the 1970s and 1980s, paralleling advancements in information and communication technology. From early experiments like the "Creeper" virus and the "Morris" worm, currently malware has transformed into a sophisticated threat, like trojan, spyware, ransomware, and adware [1], [2]. Malware, with its primary goal of damaging or infecting computer, is known as a virus [3]. As soon as viruses emerged in the computer world, antivirus development began. The existence of viruses became threatened due to the presence of antivirus software. Consequently, simple viruses evolved into stealth viruses, which have the ability to hide from antivirus detection [4]. Furthermore, there is a type of malware that spreads massively like a worm, carrying out specific missions according to its creator's intentions. This malware is known as Computer Worm [5] . In addition, there is also a type of malware called a Botnet, which can be remotely controlled by its creator to infiltrate target systems and exploit them secretly [3], [6].

Recently, a type of malware that has become a significant issue for many individuals and industries is known as Ransomware. Ransomware exhibits destructive behavior similar to viruses, accompanied by ransom demands via Bitcoin [7]. These ransoms are used to restore data or systems to their original state, considering that ransomware typically encrypts specific files or all files on the target computer[8]. According to the statistics, around 57% of victims paid to recover their data, but less than 28% recovered it [9].

Today's malware extends beyond causing harm, encompassing sophisticated capabilities like data theft, user surveillance, and even control over victims' computer systems[10]. The way malware spreads and copies itself has become more complex. Older malware, called monomorphic, simply makes copies that look exactly the same. This makes them easy to catch with up-to-date antivirus software. However, newer types of malwares, like polymorphic, metamorphic, and oligomorphic, can create copies that appear different each time [11], [12]. This makes fingerprint-based detection useless because there are too many variations for antivirus software to store and analyze effectively. This is why we need better detection methods that focus on what the malware does, not how it looks. Machine learning can be used to analyze how malware behaves and identify threats even if they change their appearance.

Researchers have explored machine learning-based malware detection methods, as conducted by [13], [14]. Abujazoh et al. compared the performance of classification algorithms like SVM, k-Nearest Neighbor (kNN), and Decision Tree (DT) for malware detection. To address dataset imbalance, they divided the dataset into eight parts and

applied under-sampling. Each dataset part was added with 595 non-malware (goodware) files to achieve balanced classes. Subsequently, the researchers evaluated SVM, kNN, and DT on each dataset. DT outperformed the other two algorithms in terms of both performance and consistency. The best DT performance was achieved on dataset part 8 using Chi Square feature selection (30 best features), with an accuracy of 98.53%.

In another study, [14]investigated the performance of the kNN algorithm with various *k* values and feature selection methods for malware detection. The highest accuracy and F1-Score of 96.9% were obtained using kNN, with *k* = 3 and Information Gain feature selection (32 features).

Considering that the potential impact of a malware attack can be severe, these accuracy scores need further improvement to achieve zero tolerance for false positives and false negatives. To address this challenge, this study evaluates the performance of Ensemble Algorithms and Feature Selection methods to detect malware. The Information Gain feature and Chi-Square selection techniques are paired with each ensemble algorithm, including Random Forest, AdaBoost, and Gradient Boosting. An evaluation is subsequently performed to identify the combination that delivers the most effective performance in malware detection.

2. Research Method

The execution of this research is divided into 4 stages, as shown in Figure 1: Dataset Preparation, Pre-Processing, Modeling, and Evaluation. The explanation of this figure will be discussed in the following paragraph.

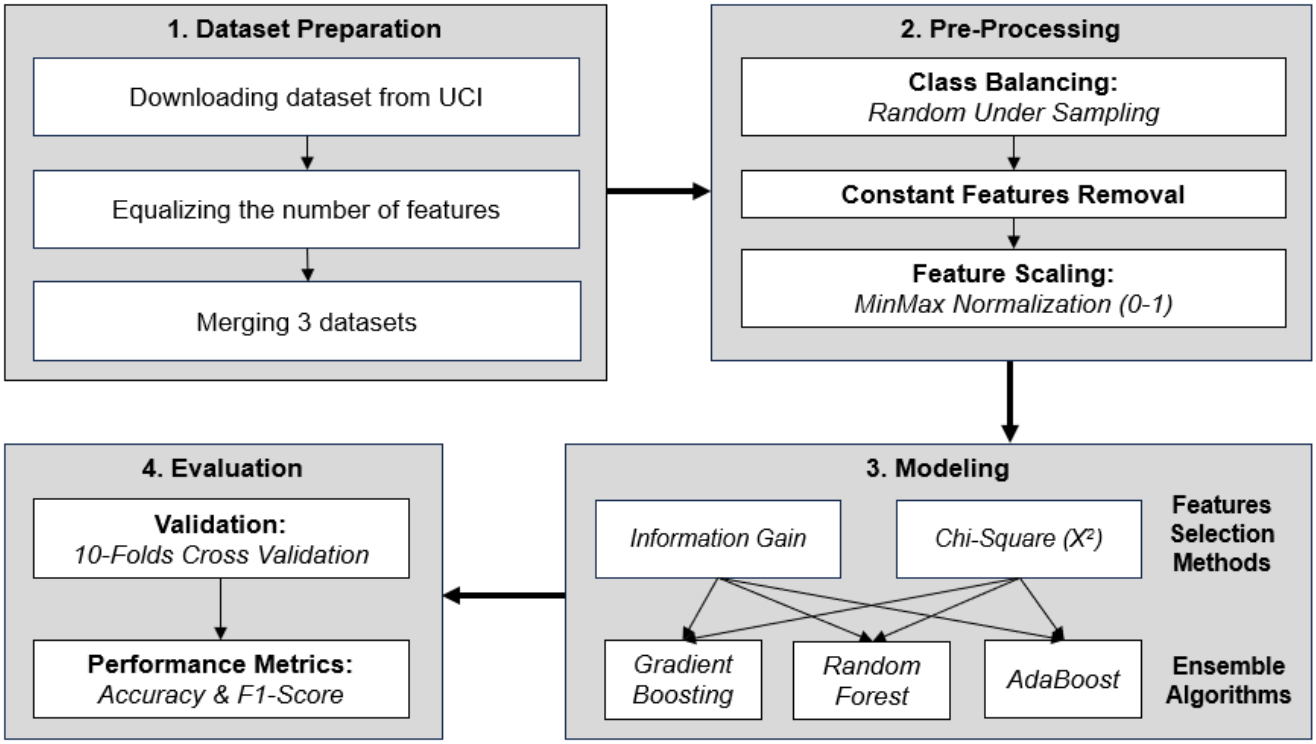

Figure 1. Research Stages

The first stage is dataset preparation. The initial step in this stage is to download the dataset from the UCI Machine Learning Repository [15]. The dataset used in this study is a public dataset with details as seen in Table 1.

Table 1. Dataset Specification

Dataset Name	Malware static and dynamic features VxHeaven and VirusTotal Data Set
Number of Files	Three [3] files: 1. A file containing goodware 2. A file with malware from VirusTotal, and 3. A file with malware from VxHeaven
Number of Records	Goodware: 595; Malware VirusTotal: 2955; Malware VxHeaven: 2698
Number of Features	Goodware: 1085; Malware VirusTotal: 1087; Malware VxHeaven: 1087
Missing Values	None

After downloading the dataset, the next step is to standardize the features. Since the three dataset files contained different numbers of features as shown in Table 1, some features such as vbaVarIndexLoad, SafeArrayPtrOfIndex, and filename were removed, leaving 1084 features for each dataset file. Subsequently, the dataset files were merged into a single dataset file. In this stage, a class attribute was added, where malware data was assigned as 1 and goodware data was assigned as 0. Thus, class 0 represents goodware files, and class 1 represents malware files.

The second stage is Preprocessing. In this stage, the dataset is processed before it is ready for modeling. The first data processing step in this stage is Class Balancing. After merging the three dataset files, an imbalance occurred, where class 0 contained 595 data points and class 1 contained 5653 data points. The class ratio was 1:9.5, categorized as medium imbalance [16]. Therefore, the dataset needed to be balanced first. The class balancing method used was Random Under Sampling (RUS), which involved reducing the number of data points in the majority class to match the number in the minority class. After applying RUS method, the dataset contained an equal number of data points for each class, specifically 595 data points.

The next step involved removing constant features–those with unchanged values or zero variance [17]. These features provided no useful information to the model. By eliminating them, we reduced the dataset's dimensionality, making the model more efficient and easier to interpret. Constant feature removal is a crucial step in optimizing machine learning models to improve performance and interpretability. In our dataset, we identified 936 such features, which were subsequently removed, leaving 148 relevant features for further analysis.

The next step in pre-processing stage is feature scaling. Feature scalling was applied to both numerical and categorical features. Its purpose is to normalize the feature ranges and reduce bias [18]. In this study, we used the MinMax Scaler or MinMax Normalization method with a range of 0 to 1 to normalize the features. The numerical features were directly processed using MinMax Normalization (0-1), while the categorical features were treated like ordinal features and normalized accordingly. Therefore, both numerical and categorical features have a consistent range between 0 and 1.

After normalizing all features, the following step is feature selection. Because this study aims to find the best combination of two feature selection methods and three ensemble algorithms to perform classification, feature selection was conducted during the modeling stage (Stage 3). In this stage, the experiments were conducted using various feature selection methods (Information Gain & Chi-Square), ensemble algorithms (Gradient Boosting, Random Forest, and AdaBoost), and a range of features, from 15 to 25. Ensemble algorithms were used in this research since it has ability to provide better performance compared to any single algorithms, like Decision Tree and kNN, in term of accuracy, precision, recall, and F1-score [19], [20].

To assess the success of each algorithm, validation is required before conducting the evaluation. In this study, we employed 10-fold cross-validation (Cross Validation with k=10). This method helps prevent overfitting in the model [21], [22]. After validation, we evaluated the model using the Confusion Matrix. The Matrix provides insights into prediction results by calculating True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. TP represents instances correctly predicted as positive (e.g., identifying malware samples as malicious), while TN corresponds to instances correctly predicted as negative (e.g., identifying benign samples as non-malicious). On the other hand, FP arises when instances are incorrectly predicted as positive (e.g., classifying benign samples as malicious), and FN occurs when the model fails to identify truly malicious samples. These metrics aid in evaluating classification model performance, including accuracy, recall, precision, and FScore. In our research, we focused on accuracy and FScore as the evaluation metrics. Accuracy reflects how often the model predicts correctly [23]. The accuracy formula can be seen in Equation 1.

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

The next evaluation metric we used is FScore. This metric strikes a balance between precision (Equation 2) and recall (Equation 3) [24]. The formula for calculating the FScore can be seen in Equation 4.

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

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

$$FScore = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{4}$$

The result were then compared to state-of-the-art like Decision Tree [13] and kNN [14], to validate whether the proposed model performs optimally.

3. Results and Discussion

This study utilized the same dataset as used in [13], [14], namely the 'Malware static and dynamic features VxHeaven and VirusTotal Data Set,' which was downloaded from the UCI Machine Learning Repository. After the preparation and preprocessing steps, three ensemble algorithms—Gradient Boosting, Random Forest, and AdaBoost—were implemented along with two feature selection methods: Information Gain and Chi-Square. The number of features tested varied from 15 to 25. The results of evaluating the combination of ensemble algorithms and feature selection methods can be seen in Table 2.

Table 2. The Experiment Results

Features	Gradient Boosting				Random Forest				AdaBoost			
	Inf. Gain		Chi-Square		Inf. Gain		Chi-Square		Inf. Gain		Chi-Square	
	Acc	FScore	Acc	FScore	Acc	FScore	Acc	FScore	Acc	FScore	Acc	FScore
15	95.5%	95.5%	97.9%	97.9%	96.1%	96.1%	97.9%	97.9%	96.2%	96.2%	97.4%	97.4%
16	95.7%	95.7%	97.8%	97.8%	95.5%	95.5%	97.6%	97.6%	96.3%	96.3%	97.5%	97.5%
17	95.8%	95.8%	98.1%	98.1%	95.5%	95.5%	97.8%	97.8%	96.1%	96.1%	97.3%	97.3%
18	98.9%	98.9%	98.1%	98.1%	98.5%	98.5%	97.7%	97.7%	97.1%	97.1%	97.4%	97.4%
19	98.9%	98.9%	98.1%	98.1%	98.5%	98.5%	97.5%	97.5%	97.3%	97.3%	97.2%	97.2%
20	98.7%	98.7%	98.2%	98.2%	98.6%	98.6%	98%	98%	97.2%	97.2%	97.4%	97.4%
21	**99.2**%	**99.2**%	98.2%	98.2%	98.7%	98.7%	97.7%	97.7%	98.4%	98.4%	97.5%	97.5%
22	99.2%	99.2%	98.1%	98.1%	98.8%	98.8%	97.5%	97.5%	98.2%	98.2%	97.1%	97.1%
23	99.2%	99.2%	98.1%	98.1%	98.8%	98.8%	97.5%	97.5%	98.2%	98.2%	97.4%	97.4%
24	99.1%	99.1%	98.2%	98.2%	98.9%	98.9%	97.5%	97.5%	98.2%	98.2%	97.5%	97.5%
25	99%	99%	98.1%	98.1%	98.8%	98.8%	97.8%	97.8%	98.2%	98.2%	97.2%	97.2%

Based on the experiment results shown in Table 2, the combination of Gradient Boosting and Information Gain with 21 features achieved the best performance compared to other combinations, achieving 99.2% accuracy and FScore. This score was also attained by Gradient Boosting and Information Gain with 22 and 23 features. Since fewer features are more efficient in terms of computation and complexity, we proposed Gradient Boosting and Information Gain with 21 features as the model for this study. The list of 21 features used in this study is provided in Table 3.

Table 3. The list of 21 Features with Highest IG Score

No.	Features	IG Score
1.	Minor_image_version	0.761
2.	Minor_operating_system_version	0.716
3.	Major_operating_system_version	0.639
4.	Size_of_stack_reverse	0.626
5.	Compile_date	0.593
6.	Minor_linker_version	0.566
7.	Major_image_version	0.555
8.	Major_subsystem_version	0.523
9.	Dll_characteristics	0.485
10.	Minor_subsystem_version	0.477
11.	CheckSum	0.408
12.	Major_linker_version	0.395
13.	Characteristics	0.389
14.	Number_of_IAT_entires	0.257
15.	Number_of_IAT_entires.1	0.257
16.	Pushf	0.243
17.	Size_of_stack_commit	0.239
18.	Files_operations	0.221
19.	.text:	0.205
20.	Count_dll_loaded	0.202
21.	SizeOfHeaders	0.195

Figure 2 illustrates the classification results in confusion matrix. According to this matrix, the model misclassified the data 22 times out of 1190 attempts. Specifically, it misclassified benign files as malware ten times and malware as benign twelve times. In the context of malware detection, understanding False Positive Rate (FPR) and False Negative Rate (FNR) is crucial for evaluating the performance of classification models in addition to accuracy and FScore. FPR represents the proportion of benign (non-malicious) instances that are incorrectly classified as malware [25]. High FPR can lead to unnecessary actions, such as blocking legitimate software or generating false alarms. On the other hand, FNR measures the proportion of actual positive instances (e.g., true malware samples) that the model misses or incorrectly classifies as negative (e.g., benign or non-malicious) [26]. High FNR poses significant risk, like security breaches, data theft, or system compromise. In our study, the model achieved FPR of 1.67% and FNR of 2.02%. In the realm of security, even a single false negative can have dire consequences for users following a malware attack. Thus, continuous improvements are essential to enhance the model's performance.

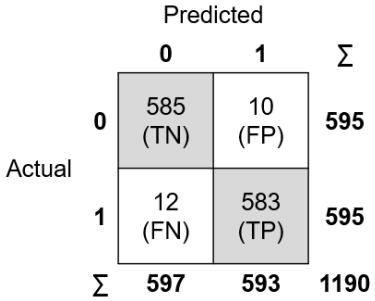

Figure 2. Confusion Matrix

The performance of the combination of Gradient Boosting algorithm and Information Gain on 21 features were compared to the models proposed by [13], [14]. The results can be seen in Table 4. Thus, our proposed method outperformed two state-of-the-art models. Not only does our method excel in terms of accuracy and FScore performance, but it also demonstrates greater efficiency by using the fewest features. Gradient Boosting combined with Information Gain (21 features) is six times faster than Gradient Boosting without any feature selection method. Table 5 presents the execution time, comparing Gradient Boosting with and without feature selection.

Table 4. The Comparison Results with State-of-the-art Methods

References	Algorithm	FS Method	Number of Features	Accuracy	FScore
Proposed	**Gradient Bossting**	**Information Gain**	**21**	**99.2%**	**99.2%**
(13)	Decision Tree	Chi-Square	30	98.5%	-
(14)	kNN	Information Gain	32	96.9%	96.9%

Table 5. The Execution Time of Gradient Boosting Algorithm with and without Feature Selection

	Execution Time (in seconds)
With Feature Selection (21 features)	4
Without Feature Selection (1084 features)	24

4. Conclusion

The hybrid model combining Gradient Boosting with Information Gain feature selection method yields an effective and efficient predictive model. This combination outperforms two other hybrid models: Decision Tree+Chi-Square and kNN+Information Gain. Specifically, Gradient Boosting+Information Gain achieves an accuracy and FScore of 99.2%, surpassing the runner-up algorithm–Decision Tree+Chi-Square–by 0.7%. Additionally, applying Information Gain as a feature selection method significantly reduces the computation time. The computational speed increases six times faster compared to scenarios without feature selection. In our study, the model achieved FPR of 1.67% and FNR of 2.02%. However, it is important to note that security demands zero tolerance for missclassifcation. Any false negatives could have severe consequences for users after a malware attack. Therefore, further improvements are necessary to enhance the model's performance.

Acknowledgements

Special thanks are extended to the Institute of Research and Community Service (LPPM) and the Faculty of Computer Science, Universitas Dian Nuswantoro for their support in facilities and funding for this research.

References

[1] M. N. Alenezi, H. Alabdulrazzaq, A. A. Alshaher, and M. M. Alkharang, "Evolution of Malware Threats and Techniques: a Review," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 12, no. 3, pp. 326–337, Dec. 2020. https://doi.org/10.17762/ijcnis.v12i3.4723

[2] C. S. Yadav and S. Gupta, "A Review on Malware Analysis for IoT and Android System," *SN Comput Sci*, vol. 4, no. 2, pp. 1–45, Mar. 2023. https://doi.org/10.1007/s42979-022-01543-w

[3] F. A. Rafrastaraa, R. A. Pramunendar, D. P. Prabowo, E. Kartikadarma, and U. Sudibyo, "Optimasi Algoritma Random Forest menggunakan Principal Component Analysis untuk Deteksi Malware," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 3, pp. 217–223, Jul. 2023. https://doi.org/10.47233/jteksis.v5i3.854

[4] M. Chen and M. Yan, "How to protect smart and autonomous vehicles from stealth viruses and worms," *ISA Trans*, vol. 141, pp. 52–58, Oct. 2023. https://doi.org/10.1016/j.isatra.2023.04.019

[5] B. Bakić, M. Milić, I. Antović, D. Savić, and T. Stojanović, "10 years since Stuxnet: What have we learned from this mysterious computer software worm?," *2021 25th International Conference on Information Technology, IT 2021*, Feb. 2021. https://doi.org/10.1109/IT51528.2021.9390103

[6] S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, "Hybrid Botnet Detection Based on Host and Network Analysis," *Journal of Computer Networks and Communications*, vol. 2020, no. 1, p. 9024726, Jan. 2020. https://doi.org/10.1155/2020/9024726

[7] N. Shahid *et al.*, "Mathematical analysis and numerical investigation of advection-reaction-diffusion computer virus model," *Results Phys*, vol. 26, p. 104294, Jul. 2021. https://doi.org/10.1016/j.rinp.2021.104294

[8] W. Z. A. Zakaria, M. F. Abdollah, O. Mohd, S. M. W. M. S. M. M. Yassin, and A. Ariffin, "RENTAKA: A Novel Machine Learning Framework for Crypto-Ransomware Pre-encryption Detection," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 378–385, 2022. https://dx.doi.org/10.14569/IJACSA.2022.0130545

[9] M. Robles-Carrillo and P. García-Teodoro, "Ransomware: An Interdisciplinary Technical and Legal Approach," *Security and Communication Networks*, vol. 2022, no. 1, p. 2806605, Jan. 2022. https://doi.org/10.1155/2022/2806605

[10] P. Feng, J. Ma, C. Sun, X. Xu, and Y. Ma, "A novel dynamic android malware detection system with ensemble learning," *IEEE Access*, vol. 6, pp. 30996–31011, 2018. https://doi.org/10.1109/ACCESS.2018.2844349

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

[12] A. Sharma and S. K. Sahay, "Evolution and Detection of Polymorphic and Metamorphic Malwares: A Survey," *Int J Comput Appl*, vol. 90, no. 2, pp. 7–11, Jun. 2014. https://doi.org/10.5120/15544-4098

[13] M. Abujazoh, D. Al-Darras, N. A. Hamad, and S. Al-Sharaeh, "Feature Selection for High-Dimensional Imbalanced Malware Data Using Filter and Wrapper Selection Methods," *2023 International Conference on Information Technology: Cybersecurity Challenges for Sustainable Cities, ICIT 2023 - Proceeding*, pp. 196–201, 2023. https://doi.org/10.1109/ICIT58056.2023.10226049

[14] C. Supriyanto, F. Adi Rafrastara, A. Amiral, S. Rosa Amalia, M. Daffa Al Fahreza, and M. Faizal bin Abdollah, "Malware Detection Using K-Nearest Neighbor Algorithm and Feature Selection," *Jurnal Media Informatika Budidarma*, vol. 8, no. 1, pp. 412–420, Jan. 2024. https://doi.org/10.30865/MIB.V8I1.6970

[15] "Malware static and dynamic features VxHeaven and Virus Total - UCI Machine Learning Repository." Accessed: Jan. 14, 2025.

[16] F. A. Rafrastara, C. Supriyanto, C. Paramita, Y. P. Astuti, and F. Ahmed, "Performance Improvement of Random Forest Algorithm for Malware Detection on Imbalanced Dataset using Random Under-Sampling Method," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 8, no. 2, pp. 113–118, May 2023. https://doi.org/10.30591/jpit.v8i2.5207

[17] Y. Prihantono and K. Ramli, "Model-Based Feature Selection for Developing Network Attack Detection and Alerting System," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 2, pp. 322–329, Apr. 2022. https://doi.org/10.29207/resti.v6i2.3989

[18] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl Soft Comput*, vol. 97, p. 105524, Dec. 2020. https://doi.org/10.1016/j.asoc.2019.105524

[19] A. Q. Md, S. Kulkarni, C. J. Joshua, T. Vaichole, S. Mohan, and C. Iwendi, "Enhanced Preprocessing Approach Using Ensemble Machine Learning Algorithms for Detecting Liver Disease," *Biomedicines*, vol. 11, no. 2, Feb. 2023. https://doi.org/10.3390/biomedicines11020581

[20] A. A. Ceran, Y. Ar, Ö. Tanrıöver, and S. Seyrek Ceran, "Prediction of software quality with Machine Learning-Based ensemble methods," *Mater Today Proc*, vol. 81, pp. 18–25, Jan. 2023. https://doi.org/10.1016/j.matpr.2022.11.229

[21] G. Battineni, G. G. Sagaro, C. Nalini, F. Amenta, and S. K. Tayebati, "Comparative Machine-Learning Approach: A Follow-Up Study on Type 2 Diabetes Predictions by Cross-Validation Methods," *Machines 2019, Vol. 7, Page 74*, vol. 7, no. 4, p. 74, Dec. 2019. https://doi.org/10.3390/machines7040074

[22] G. Orrù, M. Monaro, C. Conversano, A. Gemignani, and G. Sartori, "Machine learning in psychometrics and psychological research," *Front Psychol*, vol. 10, p. 492685, Jan. 2020. https://doi.org/10.3389/FPSYG.2019.02970/BIBTEX

[23] S. Dev, B. Kumar, D. C. Dobhal, and H. Singh Negi, "Performance Analysis and Prediction of Diabetes using Various Machine Learning Algorithms," *Proceedings - 2022 4th International Conference on Advances in Computing, Communication Control and Networking, ICAC3N 2022*, pp. 517–521, 2022. https://doi.org/10.1109/ICAC3N56670.2022.10074117

[24] G. Gupta, A. Rai, and V. Jha, "Predicting the Bandwidth Requests in XG-PON System using Ensemble Learning," *International Conference on ICT Convergence*, vol. 2021-October, pp. 936–941, 2021. https://doi.org/10.1109/ICTC52510.2021.9620935

[25] V. P. D and V. P, "Detecting android malware using an improved filter based technique in embedded software," *Microprocess Microsyst*, vol. 76, p. 103115, Jul. 2020. https://doi.org/10.1016/j.micpro.2020.103115

[26] K. Sudharson, C. Rohini, A. M. Sermakani, Dhakshunhaamoorthiy, P. Menaga, and M. Maharasi, "Quantum-Resistant Wireless Intrusion Detection System using Machine Learning Techniques," *2023 7th International Conference On Computing, Communication, Control And Automation, ICCUBEA 2023*, 2023. https://doi.org/10.1109/ICCUBEA58933.2023.10392127