ISSN: 2477-5126
e-ISSN: 2548-9356

Performance Improvement of Random Forest Algorithm for Malware Detection on Imbalanced Dataset using Random Under-Sampling Method

Fauzi Adi Rafrastara[1*)], Catur Supriyanto[2], Cinantya Paramita[3], Yani Parti Astuti[4], Foez Ahmed[5]
[1,2,3,4]Study Program in Informatics Engineering, Faculty of Computer Science,
Universitas Dian Nuswantoro, Semarang, Indonesia
[5]Faculty of Engineering, University of Technology Sydney, Sydney, Australia
email: [1]fauziadi@dsn.dinus.ac.id, [2]catur.supriyanto@dsn.dinus.ac.id, [3]cinantya.paramita@dsn.dinus.ac.id,
[4]yanipartiastuti@dsn.dinus.ac.id, [5]foez.ahmed@uts.edu.au

***Abstract* – Handling imbalanced dataset has their own challenge. Inappropriate step during the pre-processing phase with imbalanced data could bring the negative effect on prediction result. The accuracy score seems high, but actually there are many problems on recall and specificity side, considering that the produced predictions will be dominated by the majority class. In the case of malware detection, false negative value is very crucial since it can be fatal. Therefore, prediction errors, especially related to false negative, must be minimized. The first step that can be done to handle imbalanced dataset in this crucial condition is by balancing the data class. One of the popular methods to balance the data, called Random Under-Sampling (RUS). Random Forest is implemented to classify the file, whether it is considered as goodware or malware. Next, 3 evaluation metrics are used to evaluate the model by measuring the classification accuracy, recall and specificity. Lastly, the performance of Random Forest is compared with 3 other methods, namely kNN, Naïve Bayes and Logistic Regression. The result shows that Random Forest achieved the best performance among evaluated methods with the score of 98.3% for accuracy, 98.3% for recall, and 98.3% for specificity.**

***Keywords* – Random forest, imbalanced dataset, random under-sampling, malware, classification.**

I. INTRODUCTION

Malware is a software that has malicious activities. Malware attacks a variety of devices, such as PC, laptop, tablet and smartphone [1]. Malware has various types, ranging from computer virus, trojan horse, spyware, worm, botnet, even ransomware [2]. Each type of malware has different characteristics in terms of behavior. Malware with the primary aim of damaging or infecting the victim's computer is called a computer virus. Computer virus can hide its activity so that common users cannot see and feel its existence inside the computer. What user can see from the viruses is what they have done [3][4].

Computer virus then develops into a stealth virus which has the ability to hide, not only from the user's eyes, but also from antivirus detection. It allows the infection to spread massively across computer network [5]. Such technique is actually adopting the strength of computer worm, another malware that can replicate and propagate itself to all connected computers [6]. Unlike computer virus, worm does not need user intervention to start an attack. They spread quickly all over the network. 359.000 computers can be infected under 14 hours [7].

Meanwhile, there is also malware that can be controlled remotely by the attacker to infiltrate the target system and exploit silently. This type of malware is known as a Botnet [8][9]. It consist of many computers that connected to internet and controlled remotely by bot-master for destructive reason. Botnet is responsible for some security issues, includes DDoS attacks, spreading spam and taking individual client data [10]. Currently, the research about conventional botnet has been shifted to the field of Internet of Things (IoT), and called as IoT Botnet [11].

Then, there is a malware which has recently become serious threat for many people and also industries. This malware is as destructive as computer virus, but includes a ransom via bitcoin to recover the encrypted data [4]. Unfortunately, only less than 28% of victims who paid the ransom, managed to get back all their data [12]. In 2017, a famous ransomware, called Wannacry, infected over 300.000 computer victims in 150 countries [13]. This huge case has become the global issue and attract more interest on many researches all over the world to study ransomware.

Malware evolves not only based on its type, but also its ability to avoid detection. Conventional antivirus with signature-based detection can only work on monomorphic or traditional viruses. In polymorphic viruses where the parent of a malware can produce the offspring with different signatures, makes conventional antivirus cannot handle it effectively [3]. Polymorphic malware has the ability to randomly generate signatures for new files (offspring). The aim of this feature, mainly, is to avoid antivirus detection [14]. This will make difficult for antivirus to detect since the signature are different and too costly to store all signatures in virus database. Thus, handling such kind of malware becomes very ineffective. As a solution, a smart system is needed that has the ability to analyze and detect malware, both statically and dynamically.

In this study, malware detection was carried out using the Random Forest algorithm. To optimize the classification process, the Random Under-Sampling (RUS) method is applied to overcome imbalanced dataset that obtained from the UCI Machine Learning Repository. The performance results of the Random Forest algorithm are then compared with 3 other algorithms, namely kNN, Naïve Bayes and Logistic Regression to find out which algorithm gives the highest accuracy, recall and specificity results.

The rest of the paper is organized as follows. Section II and III explain the literature review and research methods. Whereas, sections IV and V discuss the result and conclusion.

*) **corresponding author**: Fauzi Adi Rafrastara
Email: fauziadi@dsn.dinus.ac.id

II. RELATED RESEARCH

Random Forest is an ensemble-based Decision Tree algorithm that can be used in both classification and regression cases. Random Forest is classified as a successful algorithm and has been widely applied either in academic or industry [15].

In the field of information security, the Random Forest algorithm is also popular to be used in malware classification. Researchers of [16] utilized Random Forest to classify malware. The dataset used is the Malimg Dataset which consists of 9,342 malware samples with imbalanced class. To prevent overfitting, researchers employed a stratified sampling method. As a result, the accuracy obtained is 95.62%.

Researchers of [17] also carried a Random Forest-based algorithm for detecting malware on the Android platform. They used Hemdds dataset that consists of 1065 goodware and 1065 malware (balanced dataset). As a result, the accuracy obtained using the Random Forest is 89.91%.

Khammas [18] applied the Random Forest algorithm to detect Ransomware. 1680 executable files were analyzed, of which 840 files belonged to the Ransomware class and the remaining 840 were goodware (balanced dataset). Researcher applied feature selection so that the number of features is reduced from 7000 features to 1000 features. Experimental results with 1000 features produced the best accuracy performance, which is 97.74%.

Shhadat et. al. [19] conducted experiments on imbalanced data with 984 malware files and 172 goodware files. Random Forest is used to provide a rating of frequently used features so that important features can be identified easily. Features that are considered unimportant, they will be removed to reduce the dimensions. To overcome the problems of imbalanced datasets, researchers used the 15-fold-cross-validation sampling method. Furthermore, the ready dataset is tested with several methods at once, starting from KNN, SVM, Bernoulli Naïve Bayes, J48 Decision Tree, Random Forest, Logistic Regression and hard voting (combinations of Logistic Regression, SVM, Bernoulli Naïve Bayes and Decision tree). The first test was conducted for binary classification. As a result, the Decision Tree has the highest accuracy score, 98%, followed by Random Forest (97.8%), Hard Voting (97%), KNN (96.1%), SVM (96.1%), Logistic Regression (95%) and Bernoulli Naïve Bayes (91%). Meanwhile, in testing for multi-class classification, Random Forest has the highest accuracy score (95.8%). Decision Tree is in second place with 92%, followed by Hard Voting (92%), Logistic Regression (90%), SVM (88.6%), KNN (88%) and Bernoulli Naïve Bayes (81.8%).

Based on the studies above, another method is needed to overcome the imbalanced dataset in order to increase the performance of the Random Forest algorithm. As is known, mistakes in classifying malware carry very high risks. Therefore, even if researchers in [17] obtained accuracy score 97.74%, it still needs to be improved.

III. RESEARCH METHOD

There are at least four stages which conducted in this study, namely Data Collection, Pre-Processing, Model Deployment and Evaluation (see Figure 1).

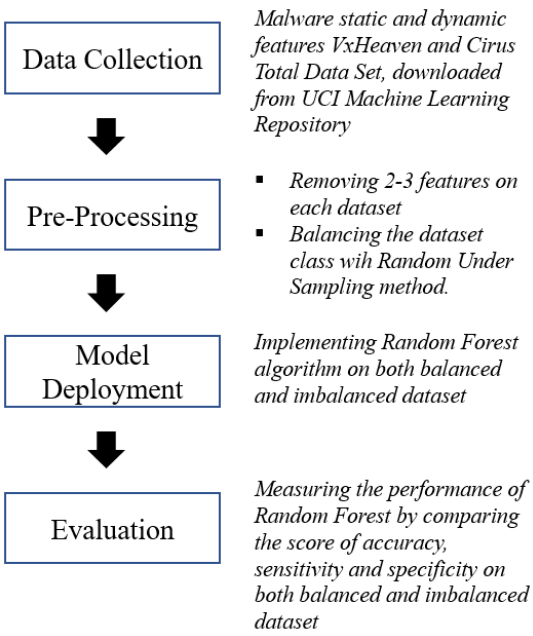

Fig 1. Research stages

A. *Hardware and Software*

It is undeniable that one of the factors which determine the smoothness and success of a research is the supporting instruments. In computer science-based research, hardware and software instruments play a very essential role. Good software without the support of qualified hardware devices, will not be able to run optimally. Conversely, hardware with top quality, but if it is not balanced with the right software, it will not help much. Therefore, hardware and software are very important in supporting the smoothness and success of a research.

In this study, a personal computer has been prepared with the following hardware specifications:

- Processor : Intel Xeon E5620
- RAM : 16 GB
- HD : 3 TB
- VGA : Radeon RX550

Meanwhile, the software employed in this research experiment were Microsoft Excel and Orange (downloaded from https://orangedatamining.com/). Microsoft Excel was used to process the dataset, including balancing the data using the Random Under-Sampling (RUS) method. On the other hand, Orange software was used to implement the Random Forest model for both unbalanced and balanced data. By using the widget of Evaluation → Test and Score, the performance's result of the Random Forest model on both data were obtained, including the score of accuracy, recall/sensitivity and specificity.

B. *Data Collection*

This study used public data downloaded from the UCI Machine Learning Repository. The details of the dataset are as follows:

- Dataset name : Malware static and dynamic features VxHeaven and Virus Total Data Set

- Number of files : 3 (goodware, malware from VirusTotal and malware from VxHeaven)
- Number of records : Goodware: 595; VirusTotal: 2955; VxHeaven: 2698
- Number of features : Goodware: 1085; VirusTotal: 1087; VxHeaven: 1087 (label excluded)
- Missing Value : No

The downloaded dataset consists of 3 separate files. The first file is the result of recording activities of various non-malware files in the sandbox. The recording results provide 1085 features (excluding labels) which will later be analyzed to determine the pattern of goodware type files. The second file is a file that contains 1087 features (excluding labels) as a result of capturing the activity of 2955 malware files obtained from VirusTotal. While the third file is the result of recording the behavior of 2698 malware that obtained from VxHeaven, with the same number of features as before, namely 1087 (excluding label).

When those data of malware and goodware are combined, then imbalanced condition cannot be avoided. A dataset is said to be imbalanced when one class has significantly greater number of samples than other classes [20]. Handling imbalanced dataset is much needed since it can affect the algorithms become biased by predicting the overall accuracy towards the class with bigger observations [21][22].

C. *Pre-Processing*

The three downloaded dataset files have major constraints, such as in terms of features and the number of labels. Regarding features, the goodware dataset has fewer features than the VirusTotal and VxHeaven datasets. We pre-processed those datasets so that they have the same numbers and feature's name. There was one feature in the three datasets that was deleted, namely the filename feature. Then in the VirusTotal and VxHeaven dataset, there are 2 additional features, which after being observed, they turn out do not have a significant value because they only contain a value of 0 for all records. Therefore, the two different features (vbaVarIndexLoad and SafeArrayPtrOfIndex) were removed, leaving 1084 features that exactly match the goodware dataset. Thus, the three datasets were ready for the further processed.

After merging the dataset files, we have 6248 data with unbalanced classes. The labels provided are category '0' for malware and '1' for goodware. In this dataset, there are 595 data in the goodware category and 5653 data in the malware category. If a comparison is made, the ratio is 1:9.5. This ratio certainly can illustrate how unbalanced the available data is. Therefore, a method is needed to deal with this problem, considering that imbalanced data can have a negative impact on the performance of an algorithm.

In this study, to overcome the problem of imbalanced data, we used Random Under-Sampling (RUS) method. Under-sampling is one of the efficient method and widely used by researchers when dealing with imbalanced dataset [23][24]. Under-sampling means decreasing the amount of instances with majority class so that it has the same number with minority class. The RUS method was implemented using Microsoft Excel, where data records from VirusTotal were combined with data from VxHeaven, so that 5653 data were collected. Furthermore, those data was randomly distributed using the rand() function, then sorted in ascending order. The top 595 data were taken, then combined with data that was included in the goodware category with 595 data as well. In the end, the final dataset used in this study has 1190 data, with 595 data labeled 0 and 595 other data labeled 1.

In this study, two experiments were carried out, namely an experiment with balanced dataset (Experiment 1) and an experiment with imbalanced dataset (Experiment 2). The illustration of those two experiments can be seen in figure 2.

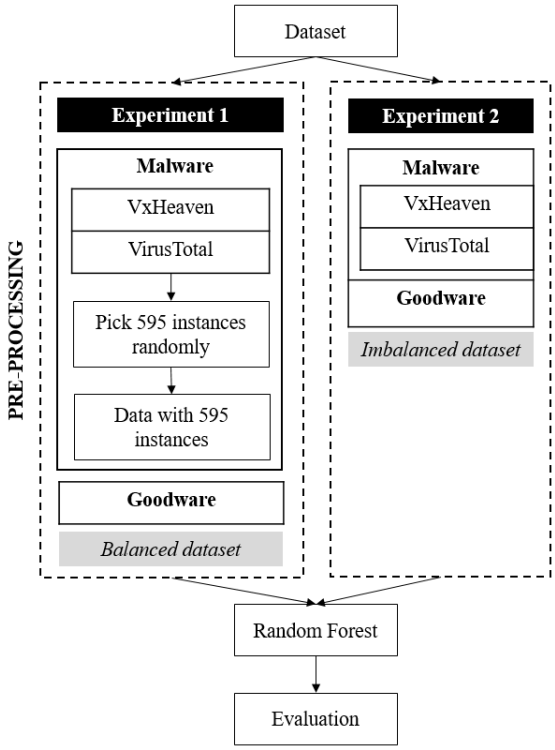

Fig 2. The flow of experiments on balanced on imbalanced dataset.

D. Model Deployment

In the modeling stage, Random Forest was chosen because it is known to have good performance. One of the advantages of Random Forest compared to other algorithms is because Random Forest uses the ensemble concept (bagging) so that the results can be more optimal.

As mentioned in the previous sub-chapter, there were 2 experiments in this study. The first experiment was to balance the data first, so that data labeled 1 (goodware) has the same amount as data labeled 0 (malware). The data was then processed using the Random Forest algorithm before being evaluated by calculating the Accuracy and Recall values.

Whereas in experiment 2, the data that processed using the Random Forest algorithm was original data by combining data from three files (goodware, VxHeaven malware, VirusTotal malware), without Random Under-Sampling to balance the data. With such data, then we found imbalanced classes, with a ratio of around 1:9.5. Next, the accuracy and recall values for the imbalanced data were calculated.

In the final stage, the results of accuracy and recall calculations in the 2 experiments were compared to find out how significant the effect of the Random Under-Sampling

ISSN: 2477-5126
e-ISSN: 2548-9356

method is in improving the performance of the Random Forest algorithm.

E. Evaluation

Apart from accuracy, the evaluation metrics used in this study are sensitivity and specificity. Accuracy is used to get an idea of how often the classification algorithm guesses correctly. The formula of Accuracy can be seen in formula 1. TP means True Positive or outcome where the model correctly predicts the positive class. TN means True Negative, in which the model correctly predicts the negative class. FP means False Positive, it is the result when the model determines something is true when it is actually false. Lastly, FN means False Negative. In contrary to FP, FN measures the model predicts something is false when it is actually true.

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

Meanwhile, sensitivity or commonly known as Recall, is used to find out how many times a model predicts a positive value for a pos detection, recall plays an important role, because the false negative generated by the system can be fatal. A malware file will have a very bad impact if it fails to be detected as malware [25].

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

The last metrics is specificity. It is mainly used to confirm the negative value. A good prediction is a prediction that has a specificity and sensitivity score of 100%. The formula of specificity can be seen in formula 3.

$$Specificity = \frac{TN}{TN+FP} \quad (3)$$

Based on the justification above, the evaluation metrics used in this study consist of three metrics, namely accuracy, sensitivity/recall and specificity.

IV. RESULT AND DISCUSSION

An experiment was conducted by applying 3 different machine learning algorithms to the identical dataset which was already pre-processed in advance. This dataset consists of 1190 data, of which 595 are labeled as '0' (malware) and the rest 595 of data are labeled as '1' (goodware). The details of the dataset was discussed in details in section III.

After conducting an experiment, the result was obtained and can be seen in Table I. According to table I, Random Forest got the highest score in term of Accuracy, Recall as well as Specificity. Random Forest outperformed three other algorithms, namely kNN, Naïve Bayes and Logistic Regression. Even though Logistic Regression has 96.1% of Recall, unfortunately its score is drop to 88.7% on specificity metrics.

Figure 3 gives better illustration to see how well Random Forest over kNN, Naïve Bayes, and Logistic Regression. Random Forest have the best and the most balanced score in terms of Accuracy, Recall, and Specificity.

Imbalanced datasets cannot be processed directly to obtain the predictive performance of a model. If it is deliberately calculated to get the accuracy value, it will certainly get a very high value, considering that one class will dominate the other class. In this unbalanced data, if forced to calculate the accuracy value, then the score is 99.2%. Of course, this is a very high number, even close to perfect. Unfortunately, that number cannot be used considering the existence of imbalanced dataset. The model will tend to be biased and fail to identify the minority class. Therefore, calculating the accuracy of the model cannot be simply applied without balancing the dataset in advance.

TABLE I.
PERFORMANCE COMPARISON OF FOUR CLASSSIFICATION ALGORITHMS

Algorithm	Accuracy	Recall	Specificity
Random Forest	98.3%	98.3%	98.3%
kNN	94.2%	96.6%	91.8%
Naïve Bayes	92.4%	91.4%	93.3%
Logistic Regression	92.4%	96.1%	88.7%

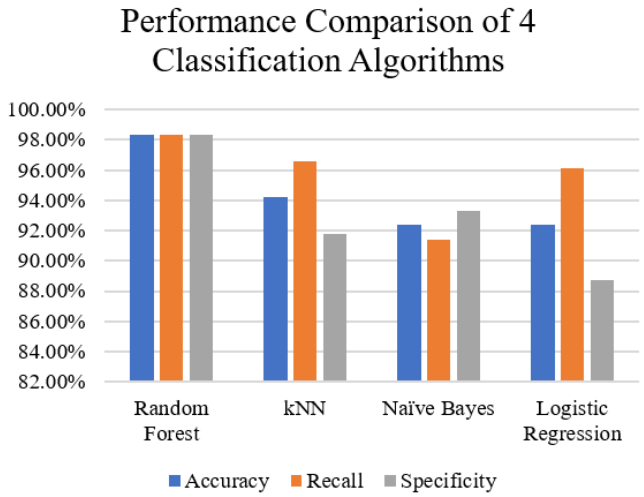

Fig 3. Performance comparison of 4 classification algorithms (Random Forest, kNN, Naïve Bayes, and Logistic Regression).

In this study, the dataset used is a dataset containing malware and goodware classifications with an imbalanced ratio of around 1:9.5, where malware is the majority class. Before being processed by the model, the dataset was balanced using the Random Under-Sampling (RUS) method so that it has a 1:1 ratio with goodware and malware classes, each of which has 595 instances.

Furthermore, the Random Forest algorithm was applied to the balanced dataset with the 5-fold cross validation sampling method. The results can be seen in table I. Random Forest has a high accuracy score, which is 98.1%, even superior to 3 other popular algorithms, such as kNN, Naïve Bayes and Logistic Regression. Table I shows that Random Forest has the best accuracy, recall and specificity scores, namely 98.3% for accuracy, 98.3% for recall and 98.3% for specificity.

When we explored further using the confusion matrix (Figure 4), it can be seen that the Random Forest algorithm successfully predicted 585 malware files out of 595 samples. It means that there were 10 malware files failed to be correctly predicted by the Random Forest. This case is an important concern considering that the fatality impact of

malware is very dangerous. Even so, this performance is the best when compared to 3 other algorithms, such as kNN, Naïve Bayes and also Logistic Regression.

		Prediksi		
		0	1	∑
Aktual	0	**585**	**10**	595
	1	**10**	**585**	595
	∑	595	595	1190

Fig 4. Confusion Matrix of Random Forest Algorithm

V. CONCLUSION

Malware detection is a challenging task since malware keeps evolving over time. Signature-based detection is no longer effective. In recent years, machine learning-based detection has been the focus of research by scientists worldwide. This study also conducted experiments on malware detection using machine learning.

This study discusses the optimization of the Random Forest algorithm on imbalanced datasets for classifying files whether they are classified as goodware or malware. The method used to balance the data is called Random Under Sampling. The recall result obtained in the Random Forest algorithm is 98.3%. This score is higher than the recall score of kNN (96.6%), Naïve Bayes (91.4%) and Logistic Regression (96.1%). Therefore, Random Under Sampling method is suitable to be applied in imbalanced dataset with Random Forest as a machine learning classifier in malware detection problem.

VI. FUTURE WORKS

In the next study, several methods will be applied to balance the data, bearing in mind that a recall score of 98.3% and a specificity of 98.3% are still relatively vulnerable to malware cases. Malware detection requires a score of 100% given its high fatality impact. In addition, feature selection will also be applied in subsequent studies, because the number of features used in this research is still relatively high, 1087 features. By decreasing the features, the performance of algorithm can be optimized, especially in terms of time taken or processing time.

CONFLICT OF INTEREST

The authors declare that the paper entitled "Performance Improvement of Random Forest Algorithm for Malware Detection on Imbalanced Dataset using Random Under-Sampling Method" is free from conflict of interest.

ACKNOWLEDGMENT

The authors would like to thank to Faculty of Computer Science and Institute of Research and Community Research, Universitas Dian Nuswantoro for all support, including facilities and funding.

REFERENCES

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

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

[3] F. A. Rafrastara and F. M. A, "Advanced Virus Monitoring and Analysis System," 2011. [Online]. Available: http://sites.google.com/site/ijcsis/.

[4] Fauzi Adi Rafrastara, *Belajar Membuat Virus Komputer Mulai dari NOL*. Semarang: Neomedia Press, 2007.

[5] H. Shah and D. M. G. Comissiong, "Computer Virus Model with Stealth Viruses and Antivirus Renewal in a Network with Fast Infectors," *SN Comput. Sci.*, vol. 2, no. 5, pp. 1–8, 2021, doi: 10.1007/s42979-021-00780-9.

[6] A. Pratama and F. A. Rafrastara, "Computer Worm Classification," *Int. J. Comput. Sci. Inf. Secur.*, vol. 10, no. 4, pp. 21–24, 2012.

[7] N. Ochieng, W. Mwangi, and I. Ateya, "Optimizing Computer Worm Detection Using Ensembles," *Secur. Commun. Networks*, vol. 2019, 2019, doi: 10.1155/2019/4656480.

[8] A. Nugraha and F. A. Rafrastara, "BOTNET DETECTION SURVEY," 2011.

[9] D. Georgoulias, J. M. Pedersen, M. Falch, and E. Vasilomanolakis, "Botnet business models, takedown atempts, and the darkweb market: a survey," *ACM Comput. Surv.*, 2022, doi: 10.1145/3575808.

[10] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evol. Intell.*, vol. 13, no. 2, pp. 283–294, 2020, doi: 10.1007/s12065-019-00310-w.

[11] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, "Internet of things botnet detection approaches: Analysis and recommendations for future research," *Appl. Sci.*, vol. 11, no. 12, 2021, doi: 10.3390/app11125713.

[12] M. Robles-Carrillo and P. García-Teodoro, "Ransomware: An Interdisciplinary Technical and Legal Approach," *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/2806605.

[13] W. Z. A. Zakaria, M. F. Abdollah, O. Mohd, M. S. M. M. Yassin, and A. Ariffin, "RENTAKA: A Novel Machine Learning Framework for Crypto-Ransomeware Pre-encryption Detection," *IJACSA) Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 5, 2022, [Online]. Available: www.ijacsa.thesai.org.

[14] F. Sulianta, "Comparison of The Computer Viruses from Time to Time," *ASIA CAUCASUS English Ed.*, vol. 23, no. 1, p. 2022, 2022, [Online]. Available: https://doi.org/10.37178/ca-c.23.1.139.

[15] F. Hidayat and T. M. S. Astsauri, "Applied random forest for parameter sensitivity of low salinity water Injection (LSWI) implementation on carbonate reservoir," *Alexandria Eng. J.*, vol. 61, no. 3, pp. 2408–2417, 2022, doi: 10.1016/j.aej.2021.06.096.

[16] F. C. C. Garcia and F. P. Muga, "Random Forest for Malware Classification," pp. 1–4, 2016, [Online]. Available: http://arxiv.org/abs/1609.07770.

[17] H. J. Zhu, T. H. Jiang, B. Ma, Z. H. You, W. L. Shi, and L. Cheng, "HEMD: a highly efficient random forest-based malware detection framework for Android," *Neural Comput. Appl.*, vol. 30, no. 11, pp. 3353–3361, 2018, doi: 10.1007/s00521-017-2914-y.

[18] B. M. Khammas, "Ransomware Detection using Random Forest Technique," *ICT Express*, vol. 6, no. 4, pp. 325–331, 2020, doi: 10.1016/j.icte.2020.11.001.

[19] I. Shhadat, B. Bataineh, A. Hayajneh, and Z. A. Al-Sharif, "The Use of Machine Learning Techniques to Advance the Detection and Classification of Unknown Malware," *Procedia Comput. Sci.*, vol. 170, no. 2019, pp. 917–922, 2020, doi: 10.1016/j.procs.2020.03.110.

[20] Q. Fan, Z. Wang, D. Li, D. Gao, and H. Zha, "Entropy-based fuzzy support vector machine for imbalanced datasets," *Knowledge-Based Syst.*, vol. 115, pp. 87–99, 2017, doi: 10.1016/j.knosys.2016.09.032.

[21] J. C. Alejandrino, J. P. Bolacoy, and J. V. B. Murcia, "Supervised and unsupervised data mining approaches in loan default prediction," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, pp. 1837–1847, 2023, doi: 10.11591/ijece.v13i2.pp1837-1847.

[22] M. Anis and M. Ali, "Investigating the Performance of Smote for Class Imbalanced Learning: A Case Study of Credit Scoring Datasets," *Eur. Sci. Journal, ESJ*, vol. 13, no. 33, p. 340, 2017, doi: 10.19044/esj.2017.v13n33p340.

[23] C. Drummond and R. C. Holte, "Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling," *Phys. Rev. Lett.*, vol. 91, no. 3, 2003.

[24] B. M. Serinelli, A. Collen, and N. A. Nijdam, "Training guidance with KDD Cup 1999 and NSL-KDD data sets of ANIDINR: Anomaly-based network intrusion detection system," *Procedia Comput. Sci.*, vol. 175, no. 2019, pp. 560–565, 2020, doi: 10.1016/j.procs.2020.07.080.

[25] S. Shakya and M. Dave, "Analysis, Detection, and Classification of Android Malware using System Calls," 2022, [Online]. Available: https://arxiv.org/abs/2208.06130v1%0Ahttps://arxiv.org/ftp/arxiv/papers/2208/2208.06130.pdf.