Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 3 (2023) 646 - 651 ISSN Media Electronic: 2580-0760

Comparative Analysis of Various Ensemble Algorithms for Computer Malware Prediction

Yusuf Bayu Wicaksono[1], Christina Juliane[2]
[1,2]Department of Information System, STMIK LIKMI
[1]yusuf.wicaksono@tarunabakti.or.id, [2]christina.juliane@likmi.ac.id

Abstract

By 2022 it is estimated that 29 billion devices have been connected to the internet so that cybercrime will become a major threat. One of the most common forms of cybercrime is infection with malicious software (malware) designed to harm end users. Microsoft has the highest number of vulnerabilities among software companies, with the Microsoft operating system (Windows) contributing to the largest vulnerabilities at 68.85%. Malware infection research is mostly done when malware has infected a user's device. This study uses the opposite approach, which is to predict the potential for malware infection on the user's device before the infection occurs. Similar studies still use single algorithms, while this study uses ensemble algorithms that are more resistant to bias-variance trade-off. This study builds models from data on computer features that affect the possibility of malware infection on computer devices with Microsoft Windows operating system using ensemble algoritms, such as Bagging Classifier, Random Forest, Light Gradient Boosting Machine, Extreme Gradient Boosting Machine, Category Boosting, and Stacking Classifier. The best model is Stacking Classifier, which is a combination of Light Gradient Boosting Machine and Category Boosting Classifier, with training and test results of 0.70665 and 0.64694. Important features have also been identified as a reference for taking policies to protect user devices from malware infections.

Keywords: *malware; machine learning; ensemble algorithm; important features*

1. Introduction

The world is currently experiencing changes due to the adoption of information technology which causes the physical and cyber worlds to overlap. Digital transformation continues to occur, including in various industries [1] thus encouraging new methods of business application to improve organizational performance, capabilities and competitiveness [2]. In recent years, companies which are engaged in almost all industrial fields have taken several initiatives to take advantage from digitalization [3]. By 2022, it is estimated that 29 billion devices are connected to the internet, so that in addition to increase connectivity, it is also estimated that cybercrime will become a major threat [4]. Cybercrime is a deliberate and malicious electronic attempt by one party to break into another party's cyber environment to steal, delete or damage valuable information [5]. One of the most common forms of cybercrime is malware infection. Malware is an acronym for malicious software or malicious software designed to harm end users. Since endpoint devices such as computers are so widely used, this is one of the weakest links in the infrastructure security chain.

Protecting endpoints from being infected by malware is an important role in cybersecurity [6]. Software vulnerabilities that cause failure due to cyberattacks or malware infections are expressed in common vulnerabilities and exposures (CVE). On Figure 1 Microsoft has the highest number of CVEs among software companies.

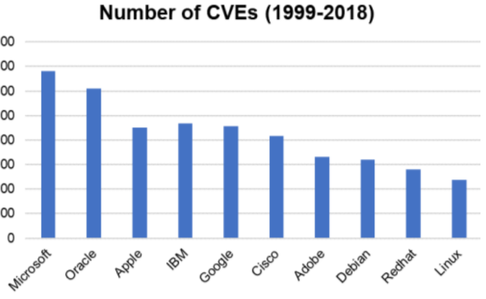

Figure 1. Total Common Vulnerabilities & Exposures (CVE) in software from 1999 to 2018 [7].

In addition, from Figure 2 it can be seen that the Microsoft operating system (Windows) contributes the largest vulnerabilities as much as 68.85% [7]. This becomes interesting to be studied because on the other

Accepted: 08-09-2022 | Received in revised: 01-05-2023 | Published: 02-06-2023

hand Microsoft Windows dominates the computer operating system market (market share) of 85%, with more than one billion users [8].

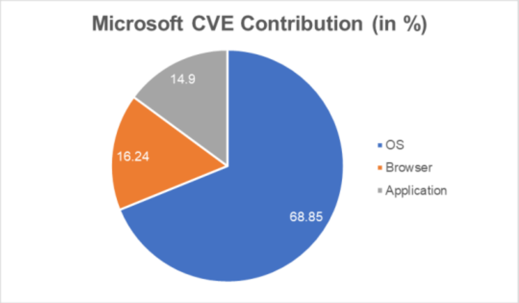

Figure 2. Microsoft operating system accounted for the most CVE at 68.85% [7].

Malware infections continue to grow in complex and diverse ways, exponentially increasing in number. As the challenges increase, the new and sophisticated methods are needed to deal with the malware infections. One method that has been used is machine learning [9]. Machine learning can build models to make predictions based on existing data. In 2018, Microsoft, in collaboration with Northeastern University and Georgia Tech, published data from 9 million computer devices and their features and information on whether or not they were infected with [10]. With this complexity, machine learning ensemble algorithms with classification types are used in this study.

Machine learning is finding models or patterns from labelled data (supervised learning) and unlabeled data (unsupervised learning) using a computer with a certain algorithm. The pattern obtained from the calculation results can be in the form of future predictions or knowledge related to the processed data. Several groups of machine learning algorithms include regression algorithms (for nominal data), classification (for categorical data), and clustering (for unlabeled data). Within each group of algorithms, various algorithms have different perspectives in solving problems. In addition, if machine learning algorithms are combined, it will form an ensemble algorithm. The ensemble algorithm is more complex than the single (base) algorithm because it is composed of those single algorithms, so it can produce better solutions [11].

The main idea behind ensemble algorithm is combining several estimators so that they make better predictions than one estimator can make. However, this does not necessarily mean combining several estimators will produce better results. On the other hand, the combined predictions of several estimators will also make the same error and will be just as wrong as each estimator in the group. Therefore, it is helpful to think of possible ways to reduce the errors made by each estimator [12].

Ensemble algorithms can be classified based on their approach in dealing with the bias-variance trade off problems. First, the ensemble algorithm can be arranged with a homogeneous algorithm. This group can be divided into averaging and boosting categories. Averaging means making decisions based on each homogeneous algorithm’s average performance, so the model will be built in parallel. Ensemble averaging is built using algorithms with low bias values but high variance values, such as Decision Tree, K-Nearest Neighbors, and Support Vector Machine. Examples of averaging group ensemble algorithms include the Bagging Classifier, Random Forest Classifier, and Extra Trees Classifier. While the boosting category means making decisions based on the performance of the homogeneous algorithm which is also derived from the performance of the previous homogeneous algorithm, so that the model is built serially. Ensemble boosting is constructed by an algorithm with a high bias value but low variance. Serial stages that continue to strengthen the algorithm will further reduce the value of the bias that appears. Examples of boosting group ensemble algorithms include the Gradient Boosting Classifier, Extreme Gradient Boosting (XGBoost) Classifier, and Light Gradient Boosting Machine (LGBM) Classifier. Second, the ensemble algorithm can be composed by heterogeneous algorithms. Heterogeneous algorithm means that the built ensemble algorithm can contain different algorithms. The heterogeneous ensemble will combine these algorithms into a new algorithm. The examples of heterogeneous ensembles are Voting Classifier and Stacking Classifier [13].

The related studies in predicting malware infection on computers has been done before as seen in table 1. These studies were built using the algorithm like Light Gradient Boosting Method (LGBM) [14], Logistic Regression , K-Nearest Neighbors, and LGBM [15], as well as LGBM and AutoAI [16].

Table 1. Related researches on the same topic.

Researcher	Year	Algorithm
Shahihi, Farhanian and Ellis	2019	Light Gradient Boosting Method (LGBM)
Pan, Tang and Yao	2020	Logistic Regression , K-Nearest Neighbors , and LGBM
Sokolov and Herndon	2021	LGBM and AutoAI

2. Research Methods

The object in this research is a dataset of malware infections from computer devices. The dataset comes from Microsoft company in collaboration with Northeastern University and Georgia Tech. Kaggle distributes the dataset in its open dataset (https://kaggle .com/c/microsoft-malware-prediction). The entire data-set consists of 8,921,572 rows by 83 columns in .csv (comma-separated values) format. Each row in this data set is associated with a computer device uniquely identified by the MachineIdentifier column. There is

DOI: https://doi.org/10.29207/resti.v7i3.4492

also the HasDetections column which contains data that indicates whether malware is detected on the computer or not. The model will be developed using 83 columns of data to predict the HasDetections value of each computer device.

This study uses the research method to derive a model from the malware infection dataset. This research method consists of steps such as data collection and preparation, data cleaning and transformation, exploratory data analysis, ensemble algorithm selection, algorithm training and model testing, and model evaluation.

3. Results and Discussions

This study used a relatively large dataset size. In order to not using large processing memory, the used data are limited to 200,000 lines. This division will facilitate further analysis because there is a limited computing engine. So it is necessary to restrict the amount of data to be processed. The prediction target column is the HasDetections column. Meanwhile, the other 82 columns will be used as feature data to predict the value of the HasDetections (target) column.

No duplicate data was found after checking. No data is the same, so all data comes from different computer devices. Then when looking at null values, it turns out that columns with empty values are found, as can be seen in the Table 2.

Table 2. 15 columns with the most percentage of empty values.

Column Name	Percentage
PuaMode	99.9735
Census_ProcessorClass	99.5635
DefaultBrowsersIdentifier	95.1485
Census_IsFlightingInternal	83.034
Census_InternalBatteryType	71.0435
Census_ThresholdOptIn	63.5035
Census_IsWIMBootEnabled	63.419
SmartScreen	35.5615
OrganizationIdentifier	30.7685
SMode	5.9315
CityIdentifier	3.6075
Wdft_IsGamer	3.3685
Wdft_RegionIdentifier	3.3685
Census_InternalBatteryNumberOfCharges	3.0325
Census_FirmwareManufacturerIdentifier	2.0505

Cleaning the data in this study was done by removing columns with more than 60% empty data. Thus, seven columns will be omitted, namely the PuaMode, Census_ProcessorClass, DefaultBrowsers Identifier, Census_IsFlightingInternal, Census_Internal Battery-Type, Census_ThresholdOptIn, and Census_Is WIM-BootEnabled columns. Column grouping is also done into columns of numeric, binary, and categorical data types. In numeric column, empty data replacement with constant value negative one (-1) is conducted. In the binary column, the blank data are replaced by the most frequent value that appears (mode) in each column. While in the categorical column, the blank data is replaced by the word 'unknown'. Therefore, there will be no column containing empty data.

Transformation is conducted with encoding technique or coding. Coding is established by coding based on category label (label coding) and the appearance frequency of its category label (coding frequency). Label coding is done by sequentially giving numbers on each label inside the columns. The number of labels' limit used in this study are more than 40 labels. Thus, frequency coding is conducted based on AvSigVersion, OsBuildLab, Census_ OSVersion, AppVersion, and EngineVersion columns because all of those columns have more than 40 category labels. Whereas, the other categorical columns are conducted by label coding.

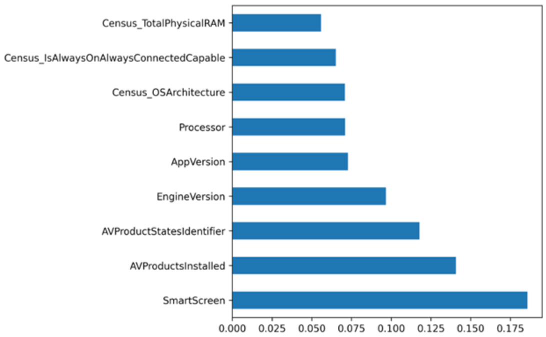

Figure 3. Top nine (absolute) correlation to target column.

From the target side (HasDetections), the data used have almost equal percentages of infection and uninfected, which are 50.03% and 49.97%, respectively. It is important to know the correlation of each column with the target before moving on to the next stage. Data analysis was established by finding correlations among features and targets on the dataset. The existence of the correlation (Figure 3) allows algorithm to be developed so that it can form a suitable model. As mentioned before, the ensemble algorithm includes averaging, boosting, and heterogeneous types. The exact ensemble algorithm to be used are averaging and boosting types, while the heterogeneous type only be used when two or more ensemble algorithms own the best practice and test results. Every algorithm has parameters that define the results of the performance model. The category boosting algorithm does not require a parameter search because it will automatically search for the best parameters.

Algorithm training is carried out using cross-validation with the Stratified method. This method allows the data to be separated into parts that are not related to each other, but still maintain the proportions as the initial data. Cross-validation will reduce the risk of overfitting during algorithm training. In this study, cross-validation was carried out by dividing the data into five parts. Meanwhile, the best parameter search for each algorithm is carried out using the Bayesian method.

DOI: https://doi.org/10.29207/resti.v7i3.4492

This method will search for parameters with more precision than choosing random numbers from a range of parameter values and is much lighter computationally than searching a grid or iterating values per value for each parameter value range. The algorithm will then be trained with 100,000 data to discover the best value and parameters.

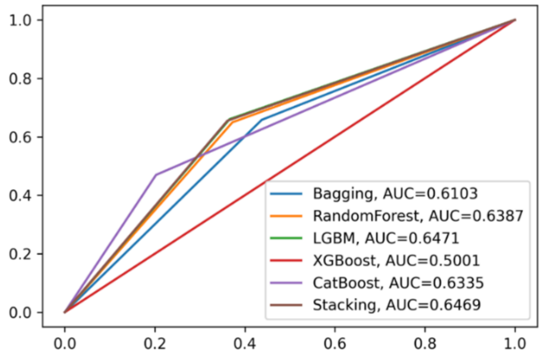

Figure 4. ROC curve and AUC value of each model.

The model then be tested with other 100,000 data that have not been recognized. The result of the training and testing for each ensemble algorithm can be seen in Figure 4, Bagging Classifier produces the lowest training value, while Extreme Gradient Boosting Classifier produces the lowest testing value. The highest accuracy of training result is owned by Category Boosting Classifier, while Light Gradient Boosting Machine Classifier owns the highest accuracy of model testing result. Because two different algorithms own the highest training and testing results values, the heterogeneous type (Stacking Classifier) of ensemble algorithm is need to be developed.

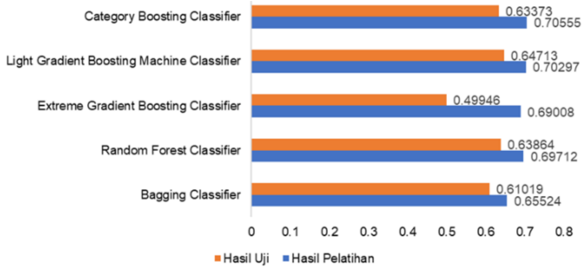

Figure 5. Comparison accuracy results training algorithm and model test performance.

The Stacking Classifier training result value is higher, compared to LGBM Classifier (0.5235%) and CatBoost Classifier (0.1559 %) values. While the Stacking Classifier testing result value is quite lower, compared to the LGBM Classifier (-0.0293%) value, but is higher than CatBoost Classifier (2.0845 %) value (Table 3). Therefore, Stacking Classifier has 0.5235% higher training accuracy value than the highest training value shown in Figure 5. Whereas, it has 0.0293% lower testing accuracy value than the highest testing value shown in Figure 5.

Table 3. Difference of training and test result.

Ensemble Algorithm	Training Result	Difference (%)	Test Result	Difference (%)
LGBM	0.70297	-	0.64713	-
CatBoost	0.70555	0.3670	0.63373	-2.0706
Stacking	0.70665	0.5235	0.64694	-0.0294

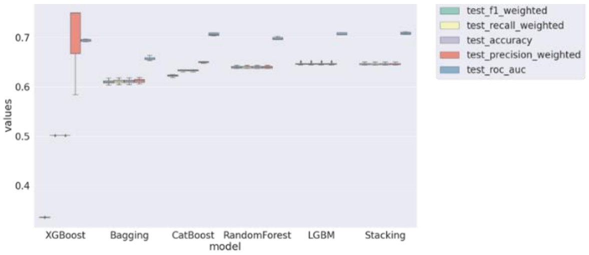

Figure 6. F1 values, recall, accuracy, precision, and ROC-AUC for each model.

A detailed evaluation from combining True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) scores can be developed into metric evaluation such as accuracy, recall, precision, F1, and ROC-AUC. Accuracy (Figure 5) and ROC-AUC of each model have been discussed before. The value of recall, precision, F1 for each model can be seen in Figure 6. If the result shows a value which is getting closer to the value of one, it means that the model performance is improving. LGBM and Stacking Classifier have the best performance.

The final evaluation is obtained from the training time (fit_time) and testing time (score_time) as seen in Figure 7. The smaller the number of times obtained, the faster the model can be trained and tested. The highest training time is owned by Stacking models (1,372.6 minutes) because there are two different ensemble algorithms and one single algorithm inside of it. And then the fastest time is owned by Bagging model (14.7 minutes). Meanwhile, the testing time results along all of the models do not have significant differences.

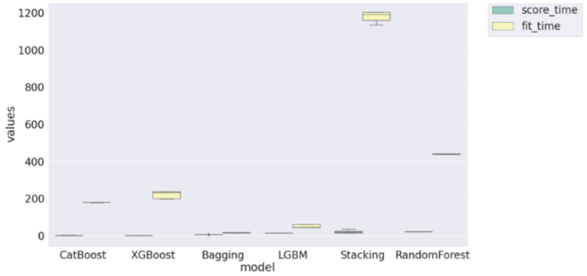

Figure 7. Length of training time (fit_time) and test time (score_time) for each model.

Overall, the best model is Stacking Classifier because it has the highest training results value. It also has the testing result value which is almost the same as LGBM value, and the difference is only 0.029%. Stacking Classifier is also prime in various metrics in Figure 6. However, this model needs the longest training time

DOI: https://doi.org/10.29207/resti.v7i3.4492

compared to other models, even though the testing time is not quite different from other models.

The important features is the most influential column (feature) to predict whether the computer is infected with malware or not. Several models can point out the important features in them. For example, Random Forest, LGBM, XGBoost, and CatBoost models have important features that can be seen in Figure 8. All models agree that the most important feature is a SmartScreen with higher value than other features. In addition, the superior model in Figure 5 are LGBM and CatBoost, so these two models will be used as other important feature's reference.

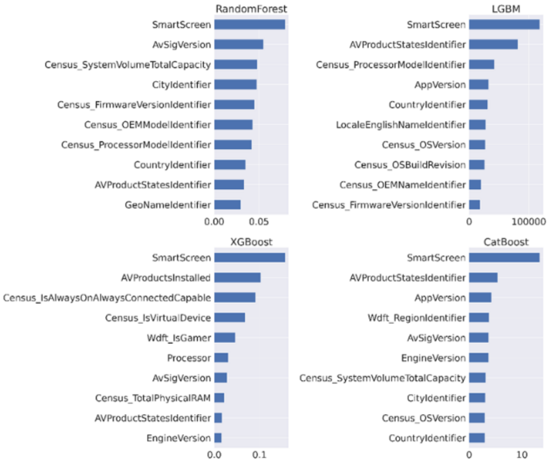

Figure 8. Ten feature important of the Random Forest, LGBM , XGBoost , and CatBoost models.

Important feature is useful as a reference for taking policy, so that the device users can avoid malware infection. Users can enable SmartScreen to prevent malware from the internet, because the internet connection existence on their device already becomes a general thing. SmartScreen referred to in the dataset feature is Microsoft Defender SmartScreen. This SmartScreen has a function to protect malware originating from websites or applications, especially when there is an action to download files. In other words, SmartScreen will detect malware originating from the internet when a website or application downloads files to the computer.

LGBM and CatBoost models consider AVProductStatesIdentifier to be a second important feature. AVProductStatesIdentifier is the identity number (ID) of the antivirus settings or configuration on the device. Different antivirus configurations will result in different AVProductStatesIdentifier values. Certain antivirus configurations will affect whether the device will be infected with malware or not. The best configuration is in the AVProductStatesIdentifier with a value of 53447. Computer users can customize the configuration by changing the AVProductStatesIdentifier value in the Windows Registry using the Registry Editor.

The Census_ProcessorModelIdentifier feature is the third important feature of the LGBM model. This feature contains an identification number (ID) related to the type of processor installed in the device. Meanwhile in the CatBoost model, the AppVersion feature occupies the third important position. AppVersion contains the version number of the antivirus installed on the device. Thus, the antivirus version and processor model installed will have an influence on the possibility of malware infection. The higher the number of cores on the processor, the better defense the device will have. This is because the antivirus service will run in the background automatically, thus requiring space for processing. Processors with a low number of cores such as 2 cores will have little difficulty in detecting malware, especially if users run multiple applications simultaneously (multi-tasking). And the other important features also can be use for being consideration to end users, like feature-based on location, hardware specification, operating system, and installed antivirus. Microsoft needs to strengthen user security on CountryIdentifier with codes 43, 29, and 141 because it is experiencing a lot of malware attacks. The strengthening can be done by providing support for updates and upgrades, especially related to the operating system version and the operating system build version. Users are recommended to turn on the Windows Update feature to receive each of these updates. Likewise with antivirus updates, users need to ensure that the antivirus signature and engine versions are kept up to date. The antivirus engine is the core of any antivirus program, which is aimed at finding malware code that has infiltrated the system. While the signature is a database that is used as a reference to identify malware that has infected. Due to these various updates, users need to provide free space on the storage so that the update process runs smoothly.

Malware infection research is mostly done when malware has infected a user's device. This study uses the opposite approach, which is to predict the potential for malware infection on the user's device before the infection occurs. Similar studies (Table 1) still use single algorithms, while this study uses ensemble algorithms that are more resistant to bias-variance trade-off. The ensemble algorithm used in the previous study was only LGBM, and it was proven to be able to produce a model that outperformed single algorithms. The results of this study indicate that the CatBoost algorithm is able to match the performance of LGBM, and the combination of the two algorithms (Stacking Classifier) is able to produce a model with better performance. In addition, the built model's performance can be upgraded by limiting the features used by the model, referring to important feature. Thereby, the noise possibility from the unimportant features can be removed.

4. Conclusion

In this study, the data is divided into 100,000 rows of data for training and 100,000 rows of data for testing. The data needs to go through the cleaning, filling empty value, and encoding process before further analysis. There is a correlation between feature and target (HasDetections column) which allows the model to be built from machine learning algorithms. This research use ensemble algorithm such as Bagging Classifier, Random Forest Classifier, Extreme Gradient Boosting (XGBoost) Classifier, Light Gradient Boosting Machine (LGBM) Classifier, and Category Boosting (CatBoost) Classifier. CatBoost produces the highest training value (0.70297) and LGBM produces the highest testing value (0.64713). These two algorithms are combined to be Stacking Classifier model with training and testing value of 0.70665 and 0.64694. In this research, it is found out that several features have important roles in predicting whether the computer is infected by malware or not. The features are SmartScreen, AVProductStateIdentifier, Census_ProcessorModelIdentifier, AppVersion , AvSigVersion , Census_SystemVolumeTotalCapacity , CountryIdentifier, CityIdentifier, and Census_FirmwareVersionIdentifier . Important feature is useful as a reference for taking policy, so that the device users can avoid malware infection.

References

[1] D. P. F. Möller, *Cybersecurity in Digital Transformation: Scope and Applications*. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-60570-4.

[2] I. Pihir, K. Pupek, and M. Furjan, ‘Digital Transformation Insights and Trends’, in *Proceedings of the Central European Conference on Information and Intelligent Systems*, Croatia, Sep. 2018, pp. 141–149.

[3] E. Indriasari, S. Supangkat, and R. Kosala, ‘Digital Transformation: IT Governance In The Agile Environment A Study Case Of Indonesia High Regulated Company’, *International Journal of Scientific and Technology Research*, vol. 9, no. 4, pp. 1557–1562, Apr. 2020.

[4] M. Stamp, M. Alazab, and A. Shalaginov, Eds., *Malware Analysis Using Artificial Intelligence and Deep Learning*. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-62582-5.

[5] K. Thakur and A.-S. K. Pathan, *Cybersecurity fundamentals: a real-world perspective*, First edition. Boca Raton, FL: CRC Press, 2020.

[6] R. Das, *Practical AI for cybersecurity*. Florida: Auerbach Publications, 2021.

[7] T. Rains, *Cybersecurity Threats, Malware Trends, and Strategies*. S.l.: Packt Publishing, 2020.

[8] N. R. Pokhrel, H. Rodrigo, and C. P. Tsokos, ‘Cybersecurity: Time Series Predictive Modeling of Vulnerabilities of Desktop Operating System Using Linear and Non-Linear Approach’, *JIS*, vol. 08, no. 04, pp. 362–382, 2017, doi: 10.4236/jis.2017.84023.

[9] D. Gibert, C. Mateu, and J. Planes, ‘The rise of machine learning for detection and classification of malware: Research developments, trends and challenges’, *Journal of Network and Computer Applications*, vol. 153, p. 102526, Mar. 2020, doi: 10.1016/j.jnca.2019.102526.

[10] R. McCann *et al.*, ‘Microsoft Malware Prediction’, Dec. 14, 2018. https://kaggle.com/c/microsoft-malware-prediction (accessed Nov. 13, 2021).

[11] S. W. Knox, *Machine learning: a concise introduction*. Hoboken, New Jersey: John Wiley & Sons, 2018.

[12] T. Amr, *Hands-on machine learning with scikit-learn and scientific Python toolkits: a practical guide to implementing supervised and unsupervised machine learning algorithms in Python*. Birmingham Mumbai: Packt, 2020.

[13] R. Kumar, *Machine Learning Quick Reference*. Birmingham, UK: Packt Publishing, 2019.

[14] M. Shahihi, R. Farhanian, and M. Ellis, ‘Machine Learning to Predict the Likelihood of a Personal Computer to Be Infected with Malware’, *SMU Data Science Review*, vol. 2, no. 2, p. 9, 2019.

[15] Q. Pan, W. Tang, and S. Yao, ‘The Application of LightGBM in Microsoft Malware Detection’, *J. Phys.: Conf. Ser.*, vol. 1684, p. 012041, Nov. 2020, doi: 10.1088/1742-6596/1684/1/012041.

[16] M. Sokolov and N. Herndon, ‘Predicting Malware Attacks using Machine Learning and AutoAI’:, in *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods*, 2021, pp. 295–301. doi: 10.5220/0010264902950301.

DOI: https://doi.org/10.29207/resti.v7i3.4492