Proceedings of the 3rd Asia Pacific International Conference on Industrial Engineering and Operations Management, Johor Bahru, Malaysia, September 13-15, 2022

Malware Detection using Portable Executable Header and Gradient Boosting Classification Algorithm

Lulu Sabila Paza
Informatics Department
Faculty of Science and Informatics
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
lulusabilap18@if.unjani.ac.id

Tacbir Hendro Pudjiantoro
Informatics Department
Faculty of Science and Informatics
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
thp@if.unjani.ac.id

Asep Id Hadiana
Informatics Department
Faculty of Science and Informatics
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
asep.hadiana@lecture.unjani.ac.id

Abstract

The ever-increasing development of technology has created many loopholes for software attackers to commit harmful crimes. Computer system users are vulnerable to malware attacks, the spread of malware (malicious software) is one of the most common computer security problems because it causes various kinds of losses. The technique used to detect the type of malware is malware classification. The method used in malware detection is the gradient boosting classification algorithm. The increasing development of malware makes technology users have to be careful in doing something. In this case, computer system security is very important. With the creation of malware detection, system security and information security can be better maintained. The Gradient Boosting Classification algorithm is an algorithm for classification that is effective and efficient in performing malware detection. Several other studies have proven this. The Gradient Boosting Classification algorithm can classify malicious software quickly and accurately. This research also uses the Portable Executable Header which contains metadata from the executable file. This research uses data taken from Kaggle with data sources from certain sites such as total virus and taken from the extraction of executable folders in windows. The results of the data set are divided into test data and training data with 20% testing of the total data and then sorting some important features from the portable executable, resulting accuracy of the gradient boosting classification algorithm of 99.20%. Further research can be suggested to show more details of the detected malware in the malware detection system.

Keywords
Malware, Portable Executable Header, Gradient Boosting Classification Algorithm.

1. Introduction

The information technology used at this time makes computer system users vulnerable to malware attacks. The spread of malware (malicious software) is one of the most common computer security problems because it causes various kinds of losses (Setiawan, Ijtihadie, and Studiawan 2017). Examples of malware losses include damaging computer systems and causing data or information theft. Malware is programmed to disrupt operations or exploit data stored on any computer system (Sarah et al. 2021). Malware can persistently harm any product such as a PC, servers, customers, or PC organizations (Choudhary 2020).

The development of malware is increasing rapidly day by day, resulting in many new types of malware that continue to emerge. Malware classification using machine learning is one of the techniques that can be used to detect malware. Malware detection is believed to be able to solve problems that make a lot of losses, as has been proven in previous research Malware Detection & Classification using Machine Learning (Choudhary 2020), Malware Detection Using Gradient Boosting Decision Trees with Customized Log Loss Function(Gao et al. 2021), Implementation of Malware Detection Service On Android (Muhammad Habibi, Setia Juli Ismail 2017).

The increasing development of malware makes technology users always have to be careful in doing something. In this case, computer system security is very important. With the creation of malware detection, system security and information security can be better maintained.

According to previous research, the Gradient-Boosting Classification algorithm is an algorithm for classification that is effective and efficient in performing malware detection. The Gradient-Boosting Classification algorithm can classify malicious software quickly and accurately (Gao et al. 2021). In previous research, static detection of ransomware extracted features to classify whether it was ransomware, malware, or benign before being executed on the system using the gradient-boosting classification algorithm. Detection of ransomware using the gradient-boosting classification method and produces an accuracy of 98% (Mary et al. 2020).

In line with the development of technology, the growth of new malware increases and continues to grow every day, the existence of a malware detection system is important for computer system security and information security.

From the above problems, this research is able to provide information from software and can produce an effective and efficient malware detection system with Portable Executable Header and Gradient Boosting Classification Algorithm as a method for malware classification.

1.1 Objectives

The system to be created here uses more attributes than previous research to compare which is better if using more attributes in terms of accuracy. The system to be made here also does not need to install software so as not to burden the memory on the computer and efficiently can be done anywhere (online). In this security process, it is carried out by computer users by uploading executable files contained on the computer and then the user only sends the file so that it can be detected that the file contains malware or not.

2. Literature Review

According to previous research in malware detection journals using machine learning algorithms, in performing malware detection using classification algorithms by evaluating 5 algorithms, namely the naive bayes algorithm, decision tree algorithm, random forest algorithm, AdaBoost algorithm and Gradient Boosting algorithm. The Gradient Boosting algorithm is effective in classifying PE Header files. The results of testing the classification algorithm using gradient boosting with a total dataset of 138,047 are 98.790293% (Bahtiar, Widiyasono, and Aldya 2018).

In the research of an efficient approach to malware detection using PE Headers, the research identifies malware programs, features are extracted based on headers and the PE file structure is used to train several machine learning models. The proposed method identifies malware programs with 95.59% accuracy (Rezaei and Hamze 2020).

In the research entitled Malware Detection in Web Application Environments with Document Categorization using machine learning methods. This research describes the implementation of an application that applies document categorization techniques to detect malware or malicious code. Categorization technique to detect malware or malicious code especially web shell types with document categorization technique. The document categorization

process includes preprocessing and tokenization of source code. Source code, creating the Multinomial Naive Bayes and Decision Tree classifier models, and classifying the Bayes and Decision Tree classifier models, and document classification using the created classifiers. Tests conducted on 718 PHP source code files resulted in precision rates from 72% to 83% and recall 83% to 97% (Setiawan, Ijtihadie, and Studiawan 2017).

The research on Android Malware Detection Based on System Call using the Support Vector Machine Algorithm aims to detect Android malware types dynamically and classify malware types using the Support Vector Machine (SVM) algorithm. The method in this research is divided into several stages, namely research data collection, retrieval of information from system calls, data pre-processing, feature selection, data training, data processing in testing, and classification testing. This research shows the accuracy results with a dataset without feature selection of 75.5556% and a dataset with the Wrapper Subset Val feature selection method of 73.3333%. It can be concluded from this accuracy that system calls can be used as a development of Android malware type classification in the future. But the accuracy results obtained have not achieved maximum results. The accuracy results are not much different between datasets that use the feature selection method and datasets that do not use the feature selection method. Feature selection can reduce the number of features to be used for classification purposes, but does not significantly improve accuracy results (Herlambang et al. 2018).

Based on the detection results by testing three models, it can be concluded that the application of the Naïve Bayes method with pre-processing using discretization techniques can improve the results of malware detection accuracy compared to the classification process without using binning techniques (discretization). The application of Naïve Bayes on Malware Detection with Discretization This variable produces Experimental results show that the application of Naïve Bayes on data classification that has not gone through the discretization stage produces an accuracy rate of 69.72% with 63.53% malware prediction while on data that has passed the discretization stage is able to provide up to 79.97% accuracy with 81.29% malware prediction (Anggraini and Kunang 2020).

In this malware detection research using reverse engineering techniques by analyzing portable executables. The research aims to detect malware by analyzing malware using malware samples to better understand how they can infect computers and devices, the level of threat they pose, and how to protect devices against them. Based on the analysis of malware using reverse engineering techniques that have been carried out in this study, it explains that reverse engineering techniques are appropriate and provide good results (Megira, Pangesti, and Wibowo 2018).

In research on forensic malware identification using the naive bayes method explains that the research makes a malware detection system by classifying the data used using the naive bayes algorithm method to identify whether the software detected has malware or not. Malware identification by classification using the naive bayes algorithm produces an accuracy of 93% for the detection process using static characteristics and 85% for detection through dynamic methods. In this research, portable executables and API Call Sequence are used (Ramadhan, Purwanto, and Ruriawan 2020).

Android Malware Detection Analysis using Support Vector Machine and Random Forest methods is a research by analyzing malware datasets that aims to compare the results of matrix performance between the two methods and compare the results of matrix performance with previous research. This research uses a machine learning approach in the classification process.
Testing of existing scenarios found that the Random Forest method is superior to the SVM method. With Random Forest accuracy results up to 98.99% while the SVM method gets accuracy results up to 96.23%. For the performance results in this study, the accuracy is 1.22% superior to SVM, 1.45%(Sitorus, Sukarno, and Mandala 2021).

3. Methods

The research method contains the steps that will be taken in performing malware detection using the Portable Executable Header and the gradient bosting classification algorithm starting with data collection, pre-processing, implementation of the Gradient-Boosting Classification algorithm, implementation of malware detection system with PE Header, and then testing. Figure 1 shows the steps of the research method. (Figure 1)

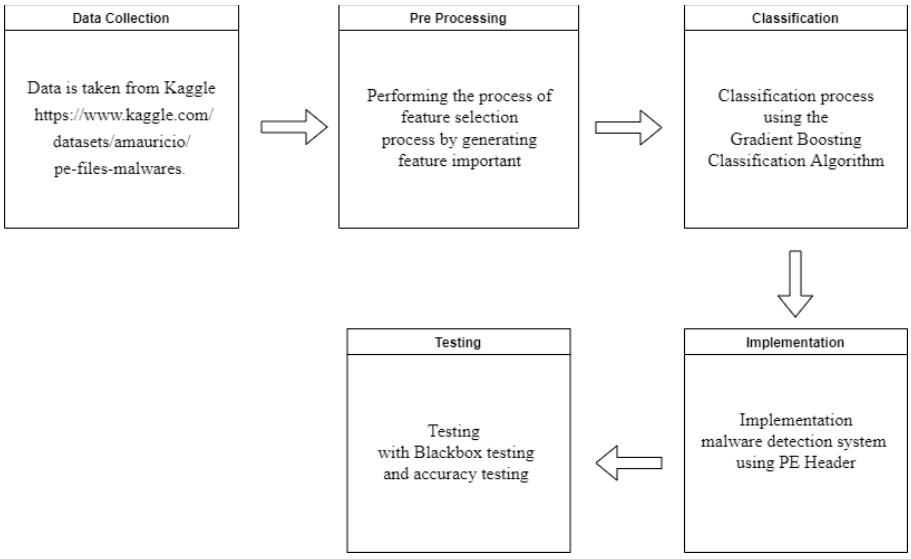

Figure 1. Research Methods

4. Data Collection

The methods used in data collection are:
Literature study is used to collect various kinds of information from previous research related to malware detection using Portable Executable Header and Gradient Boosting Classification algorithm.

Internet Searching is a data collection technique using technology in the form of search engines on the internet where all information is available. At this stage, it is determined the data to be processed and search for available data.

The data used is data that has been taken from https://www.kaggle.com/datasets/amauricio/pe-files-malwares. The data is downloaded from certain sites such as virustotal.com and some data is taken from the extraction of folders in windows. There are 79 attributes in this data.

Data collection includes malware and benign software. The software used has an executable format on Windows. The software that has been obtained will be used in the malware detection system for training and testing. Table 1 is the attributes used for the malware detection system.

Table 1. Attributes

Name	e_magic	e_cblp	e_cp	e_crlc	e_cparhdr	e_minalloc	e_maxalloc	e_ss	e_sp
e_csum	e_ip	e_cs	e_lfarlc	e_ovno	e_oemid	e_oeminfo	e_lfanew	Machine	NumberOfSections
TimeDateStamp	PointerToSymbolTable	NumberOfSymbols	SizeOfOptionalHeader	Characteristics	Magic	MajorLinkerVersion	MinorLinkerVersion	SizeOfCode	SizeOfInitializedData
SizeOfUninitializedData	AddressOfEntryPoint	BaseOfCode	ImageBase	SectionAlignment	FileAlignment	MajorOperatingSystemVersion	MinorOperatingSystemVersion	MajorImageVersion	MinorImageVersion
MajorSubsystemVersion	MinorSubsystemVersion	SizeOfHeaders	CheckSum	SizeOfImage	Subsystem	DllCharacteristics	SizeOfStackReserve	SizeOfStackCommit	SizeOfHeapReserve

SizeOfHeapCommit	LoaderFlags	NumberOfRvaAndSizes	Malware	SuspiciousImportFunctions	SuspiciousNameSection	SectionsLength	SectionMinEntropy	SectionMaxEntropy	SectionMinRawsize
SectionMaxRawsize	SectionMinVirtualsize	SectionMaxVirtualsize	SectionMaxPhysical	SectionMinPhysical	SectionMaxVirtual	SectionMinVirtual	SectionMaxPointerData	SectionMinPointerData	SectionMaxChar
SectionMainChar	DirectoryEntryImport	DirectoryEntryImportSize	DirectoryEntryExport	ImageDirectoryEntryExport	ImageDirectoryEntryImport	ImageDirectoryEntryResource	ImageDirectoryEntryException	ImageDirectoryEntrySecurity	

4.1 Pre-Processing

At the pre-processing stage, data is separated into training data and testing data. The results of this stage are training data and variables selected in feature selection as input for gradient-boosting classification. Feature selection refers to a technique that selects the subset of features (columns) that are most relevant to the data set. Fewer features can allow machine learning algorithms to run more efficiently (with less space or time complexity) and be more effective (Sarah et al. 2021)**.** At this stage, there are problems arising from the data processing process so that pre-processing is needed (Agarwal 2014). The feature selection process is carried out by generating important features.

4.2 Implementation of the Gradient-Boosting Classification Algorithm

At the implementation stage of the Gradient-Boosting Classification algorithm, the implementation process of classifying malware data on the malware detection system can produce information received from the malware detection system. The Gradient-Boosting Classification method is used in regression modeling and classification which iteratively converts weak-learners into strong-learners in prediction modeling (Sagar, Gupta, and Kaushal 2016)**.**

The Gradient Boosting Classification algorithm is performed using the following steps:
The first step is to build a basic model to predict the observed data. Take the average of the target column and consider it as the predicted value.
$F_0(x) = arg_\gamma min \sum_{i=1}^{n} L(y_i, \gamma)$ ….. (1)
Argmin means having to find the predicted/gamma value for which the loss function is minimum.
$L = \frac{1}{n}\sum_{i=0}^{n}(y_i - \gamma_i)^2$ …… (2)
Find the minimum value of gamma such that this loss function is minimum.
$\frac{dL}{d\gamma} = \frac{2}{2}(\sum_{i=0}^{n}(y_i - \gamma_i)) = \sum_{i=0}^{n}(y_i - \gamma_i)$ ….. (3)
The next step is
$\gamma_m = arg_\gamma min \sum_{i=1}^{n} L\ (y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$ ….. (4)
The last step updates the prediction of the previous model.
$F_m(x) = F_{m-1}(x) + v_m h_m(x)$ ….. (5)

4.3 Implementation of Malware Detection System using PE Header

At the implementation stage of the Malware Detection System, namely the process of creating a system using python on Google Colab using Flask and Ngrok. Implementation of a malware detection system using Portable Executable Header.

4.4 Testing

In this testing stage, testing is carried out using Blackbox testing to test the malware detection system. Then accuracy testing is carried out to calculate the accuracy of the gradient boosting classification algorithm.

5. Results and Discussion

The creation of a malware detection system is made by classifying with the gradient boosting classification algorithm method using data that is already available on Kaggle by taking Portable Executable Header characteristics.

The interface implementation displays all the views contained in the Malware Detection system. The interface implementation in Figure 2 is a look at the malware detection system.

In the main view of the malware detection system, it can upload executable files and then the file delivery process is carried out so that the system displays the detection results with two categories, namely detected malware or not detected malware.

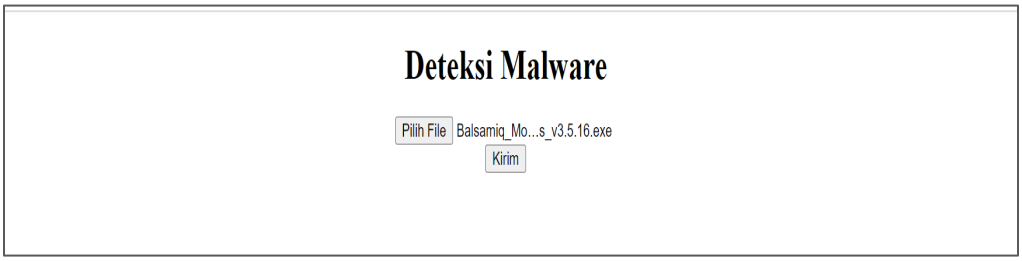

Figure 2. Malware Detection System

Figure 3 shows the implementation of the malware detection system with the detection result that the software is not detected malware or can be called a secure software.

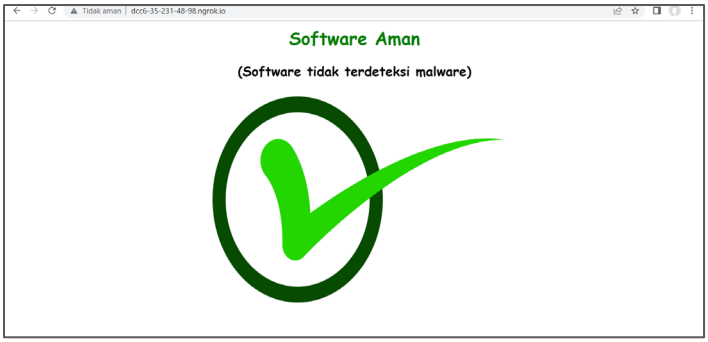

Figure 3. Software result not detected malware

Figure 4 shows the implementation of the malware detection system with the detection result that the software detected malware or can be called an unsafe software.

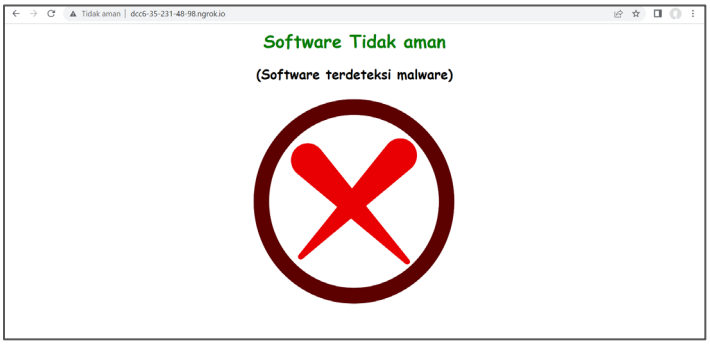

Figure 4. Software result detected malware

After implementation, testing is carried out using Black Box testing and gradient boosting classification accuracy testing. Based on the results of Black Box testing, it has been fulfilled and in accordance with the expected objectives. This shows that the design of the malware detection system has met the requirements set at the analysis stage and system requirements.

All test cases on the system can run as expected by producing a calculation of the percentage of system suitability of 100%.

Accuracy testing of the data used, using 100% for training data and 20% for test data. In testing the gradient boosting classification algorithm method, 5 trials were conducted. Table 2 is a table of confusion matrix calculation results.

Table 2. Confusion Matrix

Test to	True Negative	True Positive	False Negative	False Positive	Accuracy
1.	99.863004%	96.149242%	0.136996%	3.850758%	0.9920978842722407
2.	99.787657%	96.089385%	0.212343%	3.910615%	0.9900586286005608
3.	99.842455%	96.009577%	0.157545%	3.990423%	0.9892939077236809
4.	99.787657%	95.929769%	0.212343%	4.070231%	0.9908233494774408
5.	99.787657%	96.308859%	0.212343%	3.691141%	0.987254652052001

False positive and false negative results in a value smaller than 5% which means that this method model is very good.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

From the calculation of the gradient boosting classification algorithm, it produces the highest accuracy with 5 trials of 99.20%. The Figure 5 below is a confusion matrix image that displays performance measurements for machine learning classification problems that produce outputs in the form of safe and unsafe software.

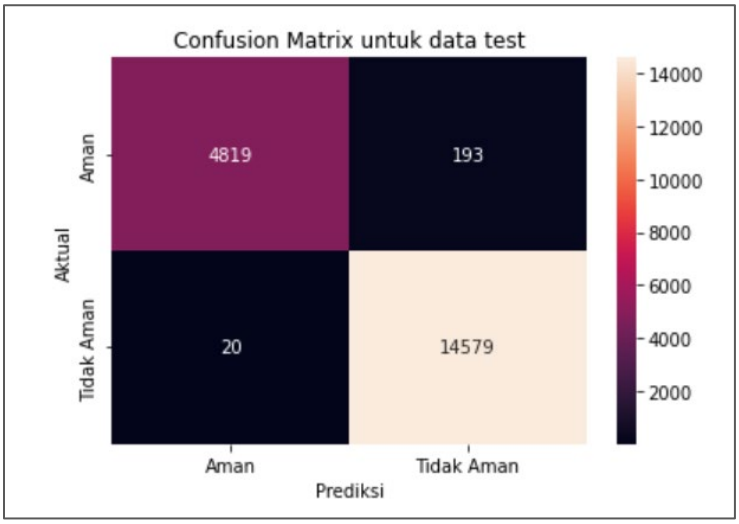

Figure 5. Confusion Matrix

6. Conclusion

The system created in this study uses 79 attributes. The Gradient Boosting Classification Algorithm method is used because the method is a gradual gradient-based approach method to increase the effectiveness of the classifier. The Gradient Boosting Classification algorithm is an algorithm for classification that is effective and efficient in performing malware detection. The system made here also does not need to install the software so as not to burden the memory on the computer and efficiently can be done anywhere (online). With the creation of malware detection, system security and information security can be better maintained. This research uses data taken from Kaggle with

data sources from certain sites such as total virus and taken from the extraction of executable folders in windows. The results of the data set are divided into test data and training data with 20% testing of the total data, then sorting some important features of the portable executable, resulting in an accuracy of the gradient boosting classification algorithm of 99.20%. In previous research, malware detection using the gradient boosting classification algorithm using 57 attributes resulted in an accuracy of 98%, it can be concluded that more attributes can affect accuracy results so that it becomes better.

Future research is suggested to be able to display more details of the types of malware detected in the malware detection system and can add more data.

References

Agarwal, Shivam. Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013 *Data Mining: Data Mining Concepts and Techniques*.

Anggraini, Inda, and Yesi Novaria Kunang. "Penerapan Naive Bayes Pada Detection Malware Dengan Diskritisasi Variabel." *Telematika* 13(1): 11–21. 2020.

Bahtiar, Fikri, Nur Widiyasono, and Aldy Putra Aldya. "Memory Volatile Forensik Untuk Deteksi Malware 2018.Menggunakan Algoritma Machine Learning." *Jurnal Teknik Informatika dan Sistem Informasi* 4: 242–53.

Choudhary, Sunita. "Malware Detection & Classification Using Machine Learning." : 20–23. 2020.

Gao, Yun, Hirokazu Hasegawa, Yukiko Yamaguchi, and Hajime Shimada. "Malware Detection Using Gradient Boosting Decision Trees with Customized Log Loss Function." *International Conference on Information Networking (ICOIN)*: 273–78. 2021.

Herlambang, Sendi, Setio Basuki, Denar Regata Akbi, and Zamah Sari. "Deteksi Malware Android Berdasarkan System Call Menggunakan Algortima Support Vector Machine." *Seminar Nasional Teknologi dan Rekayasa (SENTRA)* V: 157–65. 2018.

Mary, Martina Jose, Usharani, Manju Bala, and S G Sandhya. "Detection of Ransomware in Static Analysis by Using Gradient Tree Boosting Algorithm." *Scikit-learn*. 2020. https://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting.

Megira, S., A. R. Pangesti, and F. W. Wibowo. "Malware Analysis and Detection Using Reverse Engineering Technique." *Journal of Physics: Conference Series* 1140(1). 2018.

Muhammad Habibi, Setia Juli Ismail, Anang Sularsa. "Implementation of Malware Detection Service on Android." 3(3): 1839–47. 2017.

Ramadhan, Beno, Yudha Purwanto, and Muhammad Faris Ruriawan. "Forensic Malware Identification Using Naive Bayes Method." *2020 International Conference on Information Technology Systems and Innovation, ICITSI 2020 - Proceedings*: 1–7. 2020.

Rezaei, Tina, and Ali Hamze. "An Efficient Approach for Malware Detection Using PE Header Specifications." *2020 6th International Conference on Web Research, ICWR 2020*: 234–39.

Sagar, Medha, Arushi Gupta, and Rishabh Kaushal. "Performance Prediction and Behavioral Analysis of Student Programming Ability." 2020. *International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*: 1039–45. 2020.

Sarah, Neamat Al, Fahmida Yasmin Rifat, Md Shohrab Hossain, and Husnu S. Narman. "An Efficient Android Malware Prediction Using Ensemble Machine Learning Algorithms." *Procedia Computer Science* 191(2019): 184–91. https://doi.org/10.1016/j.procs.2021.07.023.

Setiawan, Fransiskus Gusti Ngurah Dwika, Royyana Muslim Ijtihadie, and Hudan Studiawan. "Pendeteksian Malware Pada Lingkungan Aplikasi Web Dengan Kategorisasi Dokumen." *Jurnal Teknik ITS* 6(1). 2017.

Sitorus, Yitshak Wanli, Parman Sukarno, and Satria Mandala. "Analisis Deteksi Malware Android Menggunakan Metode Support Vector Machine & Random Forest." 8(6): 12500–518. 2021.

Biographies

Lulu Sabila Paza is a final year undergraduate student in the Department of Informatics, Faculty of Science and Informatics, Jenderal Achmad Yani University, Cimahi, Indonesia. Her primary interests are systems analysis, security, data and software engineering.

Tacbir Hendro Pudjiantoro is an Associate Professor. Doctoral Candidate from the Indonesian University of Education. Researcher in the field of Knowledge Management and handles several Information Systems projects.

Asep Id Hadiana received his Master's degree in Enterprise Information System from Indonesian Computer Univerity and a Doctor of Philosophy from Technical University of Malaysia Melaka. He is a lecturer in the Informatics Department, Faculty of Science and Informatics, Universitas Jenderal Achmad Yani. Amongst his research interest are Cyber Security, Data Mining, Spatial Analysis, Location Based Services and Geographic Information Systems.