DOI: 10.15849/IJASCA.220720.13

Int. J. Advance Soft Compu. Appl, Vol. 14, No. 2, July 2022
Print ISSN: 2710-1274, Online ISSN: 2074-8523
Copyright © Al-Zaytoonah University of Jordan (ZUJ)

Analyzing ANOVA *F*-test and Sequential Feature Selection for Intrusion Detection Systems

Muhammad Jaya Siraj, Tohari Ahmad, and Royyana Muslim Ijtihadie

Department of Informatics, Institut Teknologi Sepuluh Nopember
Kampus ITS, Surabaya, 60111, Indonesia
e-mail: sirajjaya@gmail.com
Department of Informatics, Institut Teknologi Sepuluh Nopember
Kampus ITS, Surabaya, 60111, Indonesia
e-mail: tohari@if.its.ac.id
Department of Informatics, Institut Teknologi Sepuluh Nopember
Kampus ITS, Surabaya, 60111, Indonesia
e-mail: roy@its.ac.id

Abstract

An Intrusion Detection System (IDS) helps the computer system notify an admin when an attack is coming to a network. However, some problems may delay this process, such as a long time caused by several features in the captured data to classify. One of the optimization approaches is to select those critical features. It is intended to increase performance and reduce computational time. This research evaluates feature selection methods using the ANOVA **F*****-test and Sequential Feature Selection (SFS), whose performance is measured using some metrics: accuracy, specificity, and sensitivity over NSL-KDD, Kyoto2006, and UNSW_NB15 datasets. Using that approach, the performance increases, on average, by more than 10% for multiclass; and about 5% for binary class. It can be inferred that an optimal number of features can be obtained, where the best features are selected by SFS. Nevertheless, this method still needs to be improved before being implemented in a real system.***

Keywords: *Network security, Network infrastructure, Intrusion Detection System, Data Security, Information Security.*

1 Introduction

Technological advances have made sharing resources through the internet more manageable. Because of this characteristic, the security of confidential data has been an essential aspect of computer networks. This issue can lead to severe damage if it is not handled correctly. One of the methods to overcome this problem is by implementing an Intrusion Detection System (IDS) to address malicious activities in a computer network [1].

The purpose of IDS is to notify users when an attack is detected by collecting information from various sources within the system and determining whether the activity is classified as an attack [2]. The IDS can be classified into signature or misuse-based, and anomaly-based detection systems. That first IDS type works well in recognizing known attacks [3]. Nevertheless, it depends on regular pattern updates and cannot detect unknown or new threats. The second type employs user behaviour to determine the activity [4]. This detection process can be automated using machine learning, but it cannot be easy since the user activity contains many attributes; therefore, getting the best decision may take a longer computational time [5].

Furthermore, too many attributes may raise more false alarms due to redundant or duplicate records. It is shown that reducing attributes increases the system's performance [4]. The performance of IDS can be optimized by using training data to adjust the parameters of a new model, which is then implemented for future prediction or to get some important information [6]. One of the schemes is feature selection, which takes some attributes representing the actual data. By selecting only the essential features, the data can be prevented from repetitive or unwanted attributes to increase overall performance.

This research focuses on feature selection, taking the Sequential Feature Selection (SFS) and ANOVA *F*-test. Some evaluation metrics: accuracy, sensitivity, and specificity are evaluated. The performance is compared with the previous methods to see the effect of the ANOVA *F*-test and SFS. This method can be a starting point for following research employing various algorithms for selecting features.

This study is constructed in five sections. The first is the background of the research. The second describes related research, followed by explaining the method. The fourth section is the experimental results, and the last section gives the conclusion.

2 Related Work

Some previous research in IDS implements data mining to increase overall performance. Ahmad and Azis [7] focus on feature selection using Correlation-based Feature Selection, which is optimized using Particle Swarm Optimization (PSO). Specifically, this method calculates the correlation of each feature and creates subsets of features which are then filtered by the PSO algorithm after normalization. The research has obtained acceptable performances on KDD Cup99, Kyoto206, and UNSWNB15 for most measurements. However, the false positive rate on the last two datasets is relatively high.

Another approach is made [4] using feature importance and Recursive Feature Elimination (RFE) implemented in the NSL-KDD. Each feature is ranked considering the Gini index to check the feature quality and then fed into RFE. This method causes the RFE to perform faster and more accurately because unimportant features are removed in the first phase. The result shows that by adding the feature importance step, the RFE execution time is reduced, and the accuracy increases. In spite of these advantages, selecting important features takes time, especially for high-dimensional datasets.

A similar approach was previously carried out by Nkiama et al. [8], which is the main inspiration for this method. In that study, the ANOVA *F*-test is implemented to obtain the score of each feature, carried out by considering the relation between features and labels. Next, the subset of the selected feature is used to perform RFE. The result shows an increase in accuracy and reduced execution time. Nevertheless, the experiment was only carried out on the NSL-KDD dataset. Therefore, more datasets are required for the following research.

3 Feature Selection Methods

In this research, a scheme is implemented by integrating the ANOVA *F*-test and SFS, which is inspired by Nkiama et al. [8] for using ANOVA *F*-test with Recursive Feature Elimination (RFE) in the NSL-KDD dataset and Yan et al. [9] for using Back-tracing SFS as a feature selection algorithm in Fault Detection and Diagnosis (FDD) in a heating ventilation air conditioning.

The method generally is divided into two steps. The first is the ANOVA *F*-test, where features are given scores by their relation to the label to be ranked. We get the number of features from this test, which is then evaluated in SFS to find the best feature combination. The flow of this study is illustrated in Figure 1.

The first step in the research is splitting the dataset into training and testing datasets. The training dataset is for creating the model, while the testing dataset evaluates that model. The next step is data transformation, where all categorical data are transformed into numerical data. The third step is to normalize all data to prevent more features from disproportionately impacting the others. The fourth step is the ANOVA *F*-test feature ranking, that the number of features is taken as a parameter for SFS. The next step is feature selection using SFS, where features to be used for the model are calculated. Finally, the selected features are created and evaluated using the testing dataset. The main focus of this research is feature selection, which is the fourth and fifth steps.

3.1 Data Transformation

The dataset attributes are divided into two categories, numerical and categorical. The numerical data are ready to use, but the categorical data should be firstly processed into numerical. This research takes the One-Hot-Encoder scheme, which distributes each categorical value into several columns filled in by binary dummy numbers [4].

3.2 Data Normalization

The numerical data have different value scales, resulting in decreasing performance

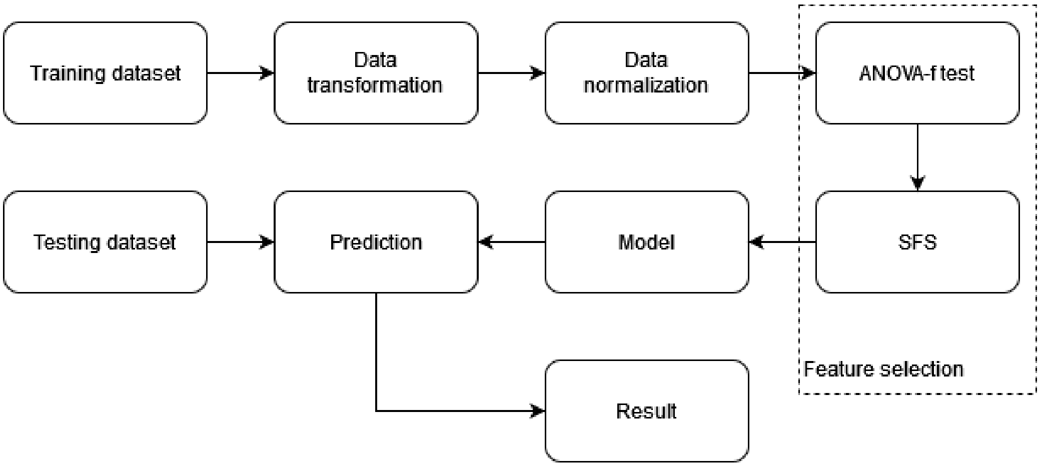

FIGURE 1. RESEARCH FLOW

because the algorithm favours a higher value, even though the value has minimal relation to the label. This can be handled by normalizing the data, wherein in this research, the *StandardScaler* from the *sklearn* kit is implemented to normalize the data. It makes the average and variance of each feature 0 and 1, respectively [4].

$$z = (x - u)/s \tag{1}$$

To have normalized data, (1) is implemented. Here, *z* is the standard score of sample *x*, *s* is the standard deviation, and *u* is the average of the training samples.

3.3 Rank Feature using ANOVA *F*-test

The ANOVA can determine the value of each feature to its label using the *F*-test to evaluate the means of different groups statistically. Each feature is then scored and ranked to see which one has the most relevant score with the label. From ANOVA, we get the number of features and an "*f* ratio". The higher the *f* ratio, the more separate the class is. The *f* ratio is calculated between class-to-class variance divided by within-class variance [10]. The score of each feature is calculated using (2), where n_i is the amount of class *i* comes up in the set, $\bar{x}_i$ is the mean of the class, $\bar{x}$ is the mean of the feature, and *k* is the number of classes.

$$\sigma_{cl}^2 = \frac{\Sigma(\bar{x}_i - \bar{x})^2 n_i}{(k-1)} \tag{2}$$

Finally, we get the score of that attribute by dividing the distance between classes by the distance within the class. The larger the value, the more relevant that feature is to the labels.

3.4 Feature Selection using SFS

SFS is an iterative feature selection, starting from an empty set and adding a feature that gives the best score for its model. After that, another feature is tested and added to the previous subset. Next, an analysis of the new subset is performed. This feature is appended if it maximizes the accuracy of the classification. This step is done sequentially until the feature set is obtained [11].

There are two main types of Sequential Feature Selection. The first is forward, which has been explained. The second is backward, which starts by taking all features and testing each removed feature. The feature with the lowest impact on the score, meaning it has no significant value, is discarded.

3.5 Classification

For classification, a Decision Tree is used as the classifier, similar to [8], partitioning the input space iteratively according to the attribute values [4]. We use the CART algorithm like [4] and evaluate the created model with the test data provided by each dataset. If there are no provided testing or training data, the dataset is split whose training and testing ratio is 2:8. The model is evaluated with various cross-validation values: 2, 5, 10, 15, 20, 30 and 50.

4 Result and Discussion

The method is evaluated using three datasets: NSL-KDD [12], Kyoto2006 [13], and UNSW_NB15 [14], considering that they represent different characteristics. It is helpful to evaluate the method from various aspects.

The NSL-KDD dataset comprises 125,973 and 22,544 training and testing data records, respectively, and 42 features, consisting of numerical and categorical feature types: 'service and flag' and 'protocol_type'. In this dataset, the attack types can be generalized into four categories: Probe, User to Root (U2R), Remote to Local (R2L), and Denial of Service (DoS). The distribution of the class can be seen in Table 1.

The Kyoto2006 dataset does not provide separate training and testing dataset. The "20151231.txt" dataset is used for this research, split into training and testing data with an 8:2 ratio. The training dataset contains 247,254 data, and the test dataset contains 61,814 data. There are 21 features, which have several categorical features, which are ‘flag’, ‘start_time’, ‘source_ip_address’, ‘protocol’, ‘service’, ‘destination_ip’, and ‘ids_detection’. The ‘start_time’, ‘source_ip_address’, and ‘destination_ip’ are dropped because they cause dimensional issue for the encoded dataset. This dataset has two labels indicating either attack or normal, whose distribution is given in Table 2. Differently, the UNSW_NB15 dataset provides 82,332 training and 175,341 testing data. This dataset consists of 44 features, three of which are categorical: 'proto', 'service', and 'state'.

This dataset has two labels; one is for determining whether it is an attack, and the other is for the attack type. This second label can be Analysis, Backdoor, Denial of Service (DoS), Fuzzers, Exploit, Generic, Reiconannce, Worm, and Shellcode. The distribution of the class can be seen in Table 3.

4.1 ANOVA *F* Feature Selection and SFS

First, ANOVA *F* feature selection is made to the dataset, where each categorical attribute is encoded using One-Hot-Encoding. This increases the number of features. In this step, *skelarn.feature_*selection is implemented, precisely the *SelectPercentile* method. It is a public programming class written in Python (https://scikit-learn.org/), which can be used for selecting features. As for the parameter *f_classif* function is used as the classifier. After processing all data with ANOVA *F*, we obtained several features sent to the SFS. In NSL-KDD, those are 12, 12, 13, and 13 for DoS, U2R, R2L, and Probe, respectively. In Kyoto2006, the optimal number is ten features, while UNSW_NB15 is 21 for binary and multiclass. The features are filtered after encoding,

Table 1: Distribution of NSL-KDD

Class	Train Data	Test Data
DoS	113,270	17,171
U2R	67,395	9,778
R2L	68,338	12,596
Probe	78,999	12,132

Table 2: Distribution of Kyoto2006

Class	Train Data	Test Data
Attack	228,851	57,156
Normal	18,403	4,658

and the detailed features are provided in Tables 4, 5, and 6.

4.2 Evaluation

The confusion matrix obtained from the experiment is provided in Tables 7, 8, and 9. Using the 13 selected features of the NSL-KDD dataset in the DoS class, the accuracy increases from 82.29% to 84.38%. This rise also applies to sensitivity, from 79.94% to 83.68%. On the contrary, the specificity decreases from 97.82% to 83.36%. In the probe class, its performance sharply climbs, where the accuracy, sensitivity, and specificity, increase from 36.52% to 89.96%, from 57.40% to 87.25%, and from 22.65% to 93.63%, respectively.

Table 3: Distribution of UNSW_NB15

Class	Training Data	Testing Data
Analysis	677	2,000
Backdoor	1,746	583
DoS	4,089	12,264
Exploit	11,132	33,393
Fuzzers	6,062	18,184
Generic	40,000	18,871
Reconnaissance	3,496	10,491
Shellcode	1,133	378
Worm	44	130

Table 4: Selected features of NSL-KDD

Class	Selected Features
Probe	'dst_host_rerror_rate', 'service_private', 'land', 'src_bytes', 'duration', 'rerror_rate', 'dst_bytes', 'dst_host_same_src_port_rate', 'dst_host_same_srv_rate', 'protocol_type_icmp', 'service_auth', 'service_pop_2', 'service_ftp_data'
U2R	'su_attempted', 'num_shells', 'num_file_creations', 'root_shell', 'service_ftp_data', 'serror_rate', 'num_access_files', 'src_bytes', 'dst_host_serror_rate', 'srv_count', 'dst_host_count', 'wrong_fragment'
R2L	'flag_OTH', 'src_bytes', 'service_ftp', 'service_urp_i', 'dst_bytes', 'num_file_creations', 'service_imap4', 'urgent', 'num_shells', 'num_access_files', 'service_ftp_data', 'srv_count', 'logged_in'
DoS	'count', 'protocol_type_icmp', 'src_bytes', 'dst_host_srv_serror_rate', 'dst_host_same_srv_rate', 'dst_host_rerror_rate', 'dst_host_serror_rate', 'service_private', 'service_domain_u', 'srv_diff_host_rate', 'dst_bytes', 'rerror_rate'

Table 5: Selected features of Kyoto2006

Class	Selected Features
Attack	'dst_host_serror_rate', 'dst_host_same_src_port_rate', 'destination_port', 'src_bytes', 'dst_bytes', 'dst_host_count', 'ids_detection_6-128-2(1)', 'ids_detection_19559-1-6(1),6-128-2(2)', 'ids_detection_1917-1-15(1)', 'dst_host_srv_serrir_rate'

Almost similar results can also be found in the R2L class. The accuracy increases from 78.54% to 79.16%, sensitivity is 53.19% to 56.23%, and specificity decreases from 99.98% to 96.70%. Lastly, in the U2R class, the accuracy increases from 99.26% to 99.45%, sensitivity is from 54.42% to 63.41%, and specificity is from 99.89% to 99.97%. From the results using the NSL-KDD dataset, it is found that removing the unnecessary features increases performance.

The following experiment uses the Kyoto2006 dataset, where we find an increase in performance when detecting an attack. The accuracy increases from 94.42% to 97.42%, sensitivity is from 74.98% to 76.64%, and specificity is from 52.10% to 77.20%. From the experimental result, we can see that, in general, there is an increase in performance using this method.

In the UNSW_NB15 dataset, it is also shown that there is improved performance for the binary and multiclass labels. There is an increase from 66.90% to 70.63% of accuracy for the binary class. The sensitivity also rises from 71.81% to 81.31%, and specificity from 85.39% to 95.31%. For the multiclass label, the accuracy goes up from 43.52% to 86.52%, sensitivity from 22.40% to 62.99%, and specificity from 31.67% to 74.96%. It is worth noting that the cross-validation value generally generates a low

Table 6: Selected features of UNSW_NB15

Class	Selected Features
Binary	'id', 'ct_dst_src_ltm', 'dpkts', 'sloss', 'dur', 'sttl', 'dload', 'dbytes', 'service_smtp', 'smean', 'state_ACC', 'dmean', 'state_RST', 'service_pop3', 'proto_udp', 'proto_arp', 'proto_sctp', 'proto_ax.25', 'ct_state_ttl', 'dtcpb', 'trans_depth'
Multi-class	'sbytes', 'sloss', 'smean', 'state_FIN', 'dbytes', 'dloss', 'dmean', 'service_-', 'state_INT', 'service_http', 'is_sm_ips_ports', 'label_bin', 'proto_ddp', 'proto_ipv6', 'proto_iplt', 'proto_tcp', 'proto_udp', 'ct_src_dport_ltm', 'sttl', 'service_ssl', 'dpkts'

Table 7: Confusion matrix using selected features of NSL-KDD

		PREDICTION							
		DoS		Probe		R2L		U2R	
		N*	A*	N*	A*	N*	A*	N*	A*
Actual	N*	9,679	32	9,650	61	9,504	207	9,683	28
	A*	21	7,439	63	2,358	126	2,759	20	47

**N=Normal, A= Attack*

Table 8: Confusion matrix using selected features of Kyoto2006

		PREDICTION	
		Normal	Attack
ACTUAL	Normal	4,526	132
	Attack	80	57,076

Table 9: Confusion matrix using selected features on UNSW_NB15

		PREDICTION	
		Normal	Binary
ACTUAL	Normal	50,589	5,411
	Binary	11,705	107,636

Table 10: The accuracy of NSL-KDD with different cross-validation values

Research	Method	Accuracy (%)			
		DoS	Probe	R2L	U2R
Nkiama et al. [8]	ANOVA *F*-test + RFE	99.90	99.80	99.88	99.90
Megantara and Ahmad [4]	Feature importance + RFE	88.98	91.18	81.29	99.42
Revathi and Malathi [19]	CFS + Random Forest	99.10	98.90	98.70	97.90
The method	ANOFA *F*-test + SFS	84.38	89.96	79.16	99.45

Table 11: The accuracy on Kyoto2006 with different cross-validation values

Research	Method	Accuracy (%)
Xu et al. [17]	CFS + Best First	99.52
Azis and Ahmad [6]	Cluster Analysis-Based	99.55
He et al. [18]	Information Gain	99.55
The method	ANOFA *F*-test + SFS	97.42

Table 12: The accuracy on UNSW-NB15 with different cross-validation values

Research	Method	Accuracy (%)
		Binary
Moustafa and Slay [15]	CP + ARM + EM	77.20
	CP + ARM + LR	77.20
	CP + ARM + NB	79.50
Kanimozhi and Jacob [16]	ANN	89.00
The method	ANOFA *F*-test + SFS	70.63

accuracy. This may be caused by the lack of training data provided in the dataset.

The comparison between this method and previous research is given in Tables 10, 11, and 12 for NSL-KDD, Kyoto2006, and UNSW_NB15 datasets, respectively. It is found that even though this feature selection can refine the performance, this method is not the best; some other methods have more remarkable improvements.

Overall, the experimental results improve the performance of all datasets. It shows that reducing the number of features and removing unnecessary features affect detection. However, depending on the hardware specification, this method takes longer to reduce the features.

5 Conclusion

In this research, we evaluate the effect of reducing the number of features on the IDS performance. Using ANOVA *F*, an optimal number of the feature is obtained, which is then used as a parameter of feature selection. Removing unnecessary features can improve the performance of the system. The accuracy rises by more than 10% and about 5% for multiclass and binary classes, respectively. Nevertheless, this feature selection takes time, which can be an overhead.

It is shown that this method still requires enhancements, considering some points of view, such as false detection rates. Despite its performance, this method has shown an alternative approach to secure networks. Moreover, it can also be the starting point of following studies. Some possible approaches can be considered in the future, for example, by taking appropriate parameters and classifiers. Furthermore, the training stage should be carried out using more data to get better values.

References

[1] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1).

[2] Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1).

[3] Akashdeep, Manzoor, I., & Kumar, N. (2017). A feature reduced intrusion detection system using ANN classifier. *Expert Systems with Applications*, (*88*).

[4] Megantara, A. A. & Ahmad, T. (2020, December). Feature Importance Ranking for Increasing Performance of Intrusion Detection System. In *3rd International Conference on Computer and Informatics Engineering*. IEEE.

[5] Liu, H., Dougherty, E. R., Dy, J. G., Torkkola, K., Tuv, E., Peng, H., Ding, C., Long, F., Berens, M., Parsons, L., Yu, L., Zhao, Z., & Forman, G. (2005). Evolving feature selection. *IEEE Intelligent Systems*, *20*(6).

[6] Aziz, M. N., & Ahmad, T. (2019). Cluster analysis-based approach features selection on machine learning for detecting intrusion. *International Journal of Intelligent Engineering and Systems*, *12*(4), 233-243.

[7] Ahmad, T., & Aziz, M. N. (2019). Data preprocessing and feature selection for machine learning intrusion detection systems. *ICIC Express Letters*, *13*(2).

[8] Nkiama, H., Zainudeen, S., & Saidu, M. (2016). A Subset Feature Elimination Mechanism for Intrusion Detection System. In *International Journal of Advanced Computer Science and Applications*, *7*(4).

[9] Yan, K., Ma, L., Dai, Y., Shen, W., Ji, Z., & Xie, D. (2018). Cost-sensitive and sequential feature selection for chiller fault detection and diagnosis. *International Journal of Refrigeration*, *86*.

[10] Johnson, K. J. & Synovec, R. E. (2002). Pattern recognition of jet fuels: Comprehensive GC × GC with ANOVA-based feature selection and principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *60*(1–2).

[11] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, *40*(1).

[12] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In *the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE.

[13] Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., & Nakao, K. (2011, April). Statistical Analysis of Honeypot Data and Building of Kyoto 2006+ Dataset for NIDS Evaluation. In *the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*. ACM.

[14] Moustafa, N., Slay, J. (2015, November). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *the 2015 Military Communications and Information Systems Conference (MilCIS)*. IEEE.

[15] Moustafa, N., & Slay, J. (2015, November). A hybrid feature selection for network intrusion detection systems: Central points. In *Australian Information Warfare and Security Conference* (pp. 5-13). ECU.

[16] Kanimozhi, V., & Jacob, P. (2019). UNSW-NB15 dataset feature selection and network intrusion detection using deep learning. *International Journal of Recent Technology and Engineering*, *7*(5).

[17] Xu, S., Zhang, Z., Wang, D., Hu, J., Duan, X., & Zhu, T. (2017, March). Cardiovascular risk prediction method based on CFS subset evaluation and random forest classification framework. In *IEEE 2nd International Conference on Big Data Analysis*. IEEE.

[18] He, F., Yang, H., Miao, Y., & Louis, R. (2016, December). A hybrid feature selection method based on genetic algorithm and information gain. In *the 5th International Conference on Computer Science and Network Technology*. IEEE.

[19] Revathi, S. & Malathi, A. (2013). A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. *International Journal of Engineering Research and Technology*, *2*(12).

Notes on contributors

Muhammad Jaya Siraj received his degree from Institut Teknologi Sepuluh Nopember, Indonesia. He has been a software developer for some years.

Tohari Ahmad is a Professor at the Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia. His research interests include network security, biometric template protection, and computer network.

Royyana Muslim Ijtihadie obtained his bachelor and master's degree from Institut Teknologi Sepuluh Nopember, Indonesia, and Ph.D from Kumamoto University, Japan. He has published some articles on computer network.