Improving Intrusion Detection System Detection Accuracy and Reducing Learning Time by Combining Selected Features Selection and Parameters Optimization

Bisyron Wahyudi Masduki*, Kalamullah Ramli*
*Department of Electrical Engineering
Universitas Indonesia
Jakarta, Indonesia
bisyron.wahyudi@ui.ac.id
kalamullah.ramli@ui.ac.id

***Abstract*—IDS capability in detecting an attacks is highly dependent on the accuracy of attack detection which usually is represented by the least number of false alarms. In this work we simplify the large network dataset by selecting only the most important and influential features in the dataset to increase the IDS performance and accuracy. The creation of smaller dataset is aimed to decrease time for training the SVM machine learning in detecting attacks. This work designed and built a prototype of IDS equipped with machine learning models to improve accuracy in detecting DoS and R2L attacks. Machine-learning algorithms is added to recognize specific characteristics of the attack at the national Internet network. New methods and techniques developed by combining feature selection and parameter optimation algorithm are then implemented in the Internet monitoring system. Through experiment and analysis, we find out that for DOS attacks the proposed approach improved accuracy for the detection and increased in speed on training and testing phase. Even though limited and appropriate selection of parameters slightly decrease the accuracy in the detection of R2L attacks but our approach significantly increases the speed of the training and testing process**

Keywords—intrusion detection system, machine-learning, support vector machine, threat, attack.

I. INTRODUCTION

There have been many IDSs developed to detect network attacks. A problem that often arises in the IDS is to overcome the problem of false detection, either false positives or false negatives. False is a mistake in recognizing IDS signature attacks that in turn confuses network administrators to take decisions in dealing with the situation. Therefore, there is need to develop our own IDS system that is able to increase the level of accuracy in recognizing the attack signature.

One approach that is widely used to improve the effectiveness of the IDS-related characteristics and adaptation is the machine-learning techniques. The IDS learns in recognizing signatures and profiles automatically based on the training dataset. With the application of machine learning techniques IDS is expected to be more accurate in recognizing attacks and easily adapt the changing network environment. Several machine learning methods are already widely known. Among others are Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), Artificial Neural Network (ANN).

In this paper the SVM method with selected features of the network dataset and optimized parameters value is proposed and implemented in IDS to detect DoS, and R2L attacks. We expect improvement in accuracy and decrease of the training time in detecting attacks.

II. RELATED WORKS

IDS is a system which oversees the entire flow of data in a network to determine the possibility of an attack. By knowing the possibility of attacks further IDS notify the system administrator so that it can be taken precaution whenever there are intruders who try to enter the network illegally. One type is a signature-based IDS that determines the presence of a malicious attack by their signatures. The IDS system will issue an alarm whenever there is a flow of data in the network corresponding to a specific attack signature. [1].

Akhmad Alimudin [2] used machine learning algorithm KNN, SVM and Dempster Shafer theory. Firstly KNN and SVM is used to classify attack data and then combine the output of those two different methods with Dempster Shafer Theory and used 8 feature from KDDCUP as training data. The performance of the classification process is good in overall, but the results are not optimal to detect R2L and U2R attacks categories. Mukkamala [4] used SVM and Neural Network methods with KDDCUP 1999 data as the training data. The result of this research shows that the SVM method is more accurate than Neural Network method.

P. Nettasan [3] coined Multi Stage Filter Using Enhanced Adaboost for Network Intrusion Detection that divide intrusion detection in three stage, namely DOS detection, Probe detection, and R2L and U2R detection. If there is no attack is detected by one of these detections than the packet data is considered as normal. Every stage of detection has different feature in quantity and kind.

Agarwal and Joshi [5] studied classifier model on a training dataset which has very different class distribution. They proposed PNrule, a two stage general-to-specific framework of learning a rule-based model. Using KDD test dataset which is separated from traning dataset and contains a lot of new R2L attack data the method is tested to find out its tradeoff. The results indicated that this method can detect only

978-1-5090-1178-0/16 /$31.00 ©2016 IEEE

10.7% of attacks in the attack class R2L despite a lot of false alarms generated. A real shortage of this method is that rule is determined automatically which makes it dependent on dataset as well as ignoring the generalization ability.

Levin [6] used Kelner Miner on KDD data set. The resulted optimal dataset of local decision trees, known as forest decision, were then subsequently used to determine the optimal subset, termed as sub-forest, to predict new cases. Multi-class categorization approach was used to detect variation of attacks on KDD dataset. The final tree detected all classes of attacks including R2L with high accuracy throughout the training dataset. However the proposed classifier achieved only 7.32% detection and 2.5% false alarm rates for the R2L attacks.

Yeung and Chow [7] proposed intrusion detection system using normal data only by taking a nonparametric density estimation approach based on Parzen-window estimators with Gaussian kernels. This detection system uses normal dataset to record 30,000 randomly selected from KDD training dataset as a sample to estimate the density of the model used. Threshold is then set to randomly obtain 30,000 normal data records from the sample KDD training dataset. The model detects whether a record is intrusive or not. This study indicated that 31.17% of R2L attacks recorded in the testing dataset KDD is detected as intrusive pattern. But there was no explanation on the false alarm rate. This model also failed to detect R2L attacks with a high detection rate.

Data mining technique to collect KDD features from DARPA 1998 dataset was proposed by Lee and Stolfo[8]. RIPPER rules were created using data mining techniques to detect R2L attacks. The attack detection accuracy of R2L attacks for this approach is 20% with false alarm rate of 0.01.

III. State of The Art

The Intrusion Detection System is a software or hardware tools that is used to detect unauthorized access of computer systems or network[9]. Some researchers have conducted studies to categorize/classify attacks data in the IDS with a specific classification method to improve the detection acuracy. As known in the previous studies, the types of attack is divided into the following four groups: DoS, Probe, R2L, and U2L. Classification method in previous research has high accuracy in detecting all group of attack, except for detecting R2L attack.

We proposed a new algorithm of SVM implementation on IDS by combining features selection and parameter optimization. The algorithm for selection of feature is used to select the best and the most influential feature of the dataset to improve the accuracy detection and to decrease the training time in classification. The parameter optimization algorithm is used to get the best value of the SVM parameters.

Feature selection algorithm is built to automatically search for the best feature subset in our dataset. The notion of "best" is relative to the problem we are trying to solve. That is, the highest accuracy in detecting attack type from our network dataset. The algorithm is used to:

- Improve the machine learning accuracy
- Intensify performance of model with high dimensional datasets
- Improve interpretability of model
- Prevent overfitting

The network dataset used in this research is GureKDDCup [11]. GureKDDCup dataset added payload as the 42th feature of KDD dataset. Payload is essential data in the form sequence of ASCII characters in the content area of network packets. In this study we propose the payload data have an important role in improving accuracy in detecting this type of R2L attack.

To achieve high-quality results on SVM machine learning algorithms needed the most appropriate parameter settings. This requires a very long time to perform parameter setting and seek the optimal parameter values manually based on learning and experience. To save time in setting a good parameter for the SVM machine learning then used parameter optimization algorithm. This parameter optimization algorithm applied to the GureKDDCUP dataset with features that have been selected to increase attack detection accuracy.

IV. Research Design

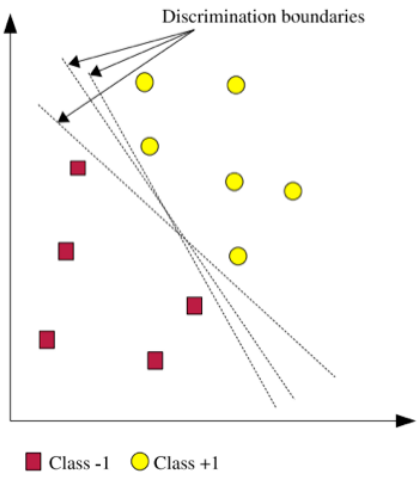
A. GureKddCup Dataset

GureKddcup is a dataset that is based on network connections kddcup99 dataset (UCI database repository), and added a new feature: the payload (the contents of network packets) in each data connection. In this way it can extracts information from any connection payload directly that will be used in the process of machine learning[12].

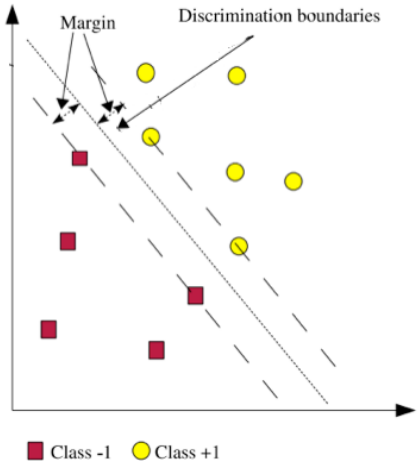
Tcpdump of the data that has been obtained is then extracted to determine the connection data and the results are presented in table format dataset with the UCI repository. 41 features are obtained for each connection feature and class. This feature is grouped into three main parts: intrinsic features extracted from the header of a packet network, content features mined from the contents of the packet network, and the traffic feature.

B. Support Vector Machine (SVM) Classifications

Basic idea of SVM is how to maximize the hyperplane as illustrated in Figure 1. Depicted in Figure 1(a) are some hyperplane choices for data set, and Figure 1(b) shows the maximal margin. Hyperplane with maximal margin gives better generalization for classification methods.



(a)



(b)

Fig. 1. Concept of SVM

SVM classification used a method for searching the best hyperplane function that separate two classes in space input. Figure 1 shows some pattern that is the member of two class data: Class +1 and Class -1. The data in Class -1 are square symbol whereas data in Class +1 are circle.

Margin is a distance between hyperplane with the closest data from each class. This closest data is termed as support vector. Solid line in Figure 1(b) shows the best hyperplane, that is located exactly in the middle of two classes. Two circle and two square that cross trough the boundary margin, i.e the dashed line, is a support vector. The method to find this hyperplane location is the core of SVM learning process.

C. *System Design*

The design of the process is shown in Figure 2 below. The process is divided into two main phases. The first phase is to choose the best features of the dataset. The second is utilizing parameter optimization algorithm to set the best value of the SVM parameters σ, ε and C.

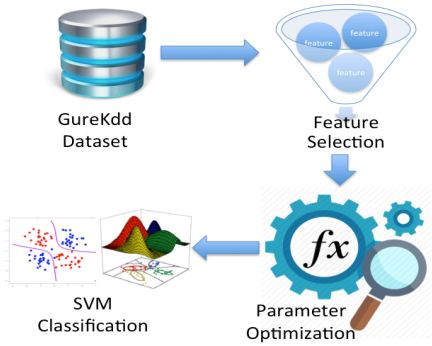


Fig. 2. System Design.

At selection phase experiment is conducted to find the most influential features in attack detection with or without payloads. Parameter optimization algorithm is implemented by using the best feature set to expect more accurate approach in detecting DoS and R2L. Getting the best features and the optimal parameter values the data is then used for training and testing.

The best and the most influential features obtained is described in the Table I for DoS attack and Table II for R2L attack detection. The following Table I below presents 27 features the most influential in determining the type of DoS attack from a total of 43 features that exist in the GureKddCup database. [11]

TABLE I. PROPOSED FEATURES FOR DOS ATTACK

No	Feature Name	Description
1	Duration	The connection length (seconds)
2	protocol_type	Protocol type used. (TCP, UDP etc.)
3	Service	Destination network service (HTTP, Telnet, etc.)
4	Flag	The network connection status (normal or error).
5	src_bytes	Source to destination data bytes
6	dst_bytes	Destination to source data bytes
7	land	1: source and destination IP addresses and port numbers are equal, else 0
8	wrong_fragment	connection bad checksum packets quantity
12	logged-in	1: logged in successfully, else 0
23	count	connections to the same destination IP address quantity
24	srv_count	connections to the same destination port number quantity
25	serror_rate	The connections percentage that have enabled the flag (4): s0, s1, s2 or s3, between the connections accumulated in count (23)
26	srv_serror_rate	The connections percentage that have enabled the flag (4): s0, s1, s2 or s3, between the connections accumulated in srv_count (24)

27	rerror_rate	The connections percentage that have enabled the flag (4): REJ, between the connections accumulated in count (23)
29	same_srv_rate	The connections percentage to the same service, between the connections accumulated in count (23)
30	diff_srv_rate	The connections percentage to different services, between the connections accumulated in count (23)
31	srv_diff_host_rate	The connections percentage to different destination ma- chines between the connections accumulated in srv_count (24)
32	dst_host_count	The connections to the same destination IP address
33	dst_host_srv_count	The connections to the same destination port number
34	dst_host_same_srv_rate	The connections percentage to the same service, between the connections accumulated in dst_host_count (32)
35	dst_host_diff_srv_rate	the connections percentage that were to different services, between the connections accumulated in dst_host_count (32)
36	dst_host_same_src_port_rate	The connections percentage to the same source port, between the connections accumulated in dst_host_srv_count (33)
37	dst_host_srv_diff_host_rate	The connections percentage that were to different destination ma- chines, between the connections accumulated in dst_host_srv_count (33)
38	dst_host_serror_rate	The connections percentage that have enabled the flag (4) s0, s1, s2 or s3, between the connections accumulated in dst_host_count (32)
39	dst_host_srv_serror_rate	The connections percentage that have enabled the flag (4) s0, s1, s2 or s3, between the connections accumulated in dst_host_srv_count (33)
40	dst_host_rerror_rate	The connections percentage that have enabled the flag (4) REJ, between the connections accumulated in dst_host_count (32)
43	payload	ASCII characters set of the content area of network packets in the context of message protocols.

Table content and description is referred to GureKddCup database description [11].

The following Table II below presents 27 features the most influential in determining the type of R2L attack from a total of 43 features that exist in the GureKddCup database. [11]

TABLE II. PROPOSED FEATURES FOR R2L ATTACK

No	Feature Name	Description
1	Duration	The connection length (seconds)
2	protocol_type	Protocol type used. (TCP, UDP etc.)
3	Service	Destination network service (HTTP, Telnet, etc.)
4	Flag	The network connection status (normal or error).
5	src_bytes	Source to destination data bytes
6	dst_bytes	Destination to source data bytes
8	wrong_fragment	1: source and destination IP addresses and port numbers are equal, else 0
12	logged-in	1: logged in successfully, else 0
21	is_hot_login	1: the user is accessing as root or admin, else 0
22	is_guest_login	1: the user is accessing as guest, anonymous or visitor
23	count	connections to the same destination IP address quantity
24	srv_count	connections to the same destination port number quantity
25	serror_rate	The connections percentage that have enabled the flag (4): s0, s1, s2 or s3, between the connections accumulated in count (23)
26	srv_serror_rate	The connections percentage that have enabled the flag (4): s0, s1, s2 or s3, between the connections accumulated in srv_count (24)
29	same_srv_rate	The connections percentage to the same service, between the connections accumulated in count (23)
30	diff_srv_rate	The connections percentage to different services, between the connections accumulated in count (23)
31	srv_diff_host_rate	The connections percentage to different destination ma- chines between the connections accumulated in srv_count (24)
32	dst_host_count	The connections to the same destination IP address
33	dst_host_srv_count	The connections to the same destination port number
34	dst_host_same_srv_rate	The connections percentage to the same service, between the connections accumulated in dst_host_count (32)
35	dst_host_diff_srv_rate	the connections percentage that were to different services, between the connections accumulated in dst_host_count (32)
36	dst_host_same_src_port_rate	The connections percentage to the same source port, between the connections accumulated in dst_host_srv_count (33)
37	dst_host_srv_diff_host_rate	The connections percentage that were to different destination ma- chines, between the connections accumulated in dst_host_srv_count (33)
38	dst_host_serror_rate	The connections percentage that have enabled the flag (4) s0, s1, s2 or s3, between the connections accumulated in dst_host_count (32)
39	dst_host_srv_serror_rate	The connections percentage that have enabled the flag (4) s0, s1,

		s2 or s3, between the connections accumulated in dst_host_srv_count (33)
40	dst_host_rerror_rate	The connections percentage that have enabled the flag (4) REJ, between the connections accumulated in dst_host_count (32)
43	payload	ASCII characters set of the content area of network packets in the context of message protocols.

Table content and description refer to GureKddCup database description [11].

Based on the dataset from the first phase parameter optimization algorithm is applied to get the best parameter. to detect DoS and R2L attacks. The result is then forwarded for training and testing process.

For ease and efficiency of machine learning in the learning process we did not use all the data GureKDDCUP which is large enough that it requires long processing time. This practice is supported by previous studies that so far typically used only 10% of the overall data. We work on a small portion of gureKddcup6percent data containing 159.321 connections data records, in which 90% of dataset is allocated for training process and the rest 10% is for testing process.

V. Analysis

The training and test results are presented in Table III.

TABLE III. Training and Testing Result Without Parameter Optimization

Data Classified Properly			158172		99.858 %
Data Classified Improperly			250		0.142 %
Processing time			54 minutes 11 seconds		
True Positive Rate	**False Positive Rate**	**Accuracy**	**F-Measure**	**MCC**	**Class**
1,000	0,220	0,999	0,999	0,881	normal
0,780	0,000	0,996	0,875	0,881	DoS
Data Classified Properly			159406		99.7216 %
Data Classified Improperly			445		0.2784 %
Processing time			31 minutes 28 seconds		
True Positive Rate	**False Positive Rate**	**Accuracy**	**F-Measure**	**MCC**	**Class**
1,000	0,181	0,997	0,999	0,903	normal
0,819	0,000	0,999	0,900	0,903	R2L

The optimized parameter value set resulting from phase two are as follows:

DoS: ***C= 0.2, S= 0, K= 2, D= 3, G= 0.0, R= 0.0, N= 0.5, M= 40.0, E= 0.001, P= 0.1***

R2L: ***C= 0.5, S= 0, K= 2, D= 3, G= 0.0, R= 0.0, N= 0.5, M= 40.0, E= 0.001, P= 0.1***

The result of training process and testing is presented in the following Table IV.

TABLE IV. Training and Testing Result with Parameter Optimization

Data Classified Properly			**158355**		**99.9735 %**
Data Classified Improperly			42		0.0265 %
Processing time			36 seconds		
True Positive Rate	**False Positive Rate**	**Accuracy**	**F-Measure**	**MCC**	**Class**
1,000	0,041	1,000	1,000	0,979	normal
0,959	0,000	1,000	0,959	0,979	DoS
Data Classified Properly			**158300**		**99.0297 %**
Data Classified Improperly			**1551**		**0.9703 %**
Processing time			1 minutes 7 seconds		
True Positive Rate	**False Positive Rate**	**Accuracy**	**F-Measure**	**MCC**	**Class**
0,999	0,564	0,991	0,995	0,611	normal
0,436	0,001	0,868	0,580	0,611	R2L

In phase I DoS attack detection accuracy rate is 99.858%, with processing time 54 minutes 11 seconds. And the result of R2L attack detection accuracy rate is 99.7216%, with a processing time 31 minutes 28 seconds.

In phase II DoS attack detection accuracy rate is 99.9735%, with processing time 36 seconds. And the result of R2L attack detection accuracy rate is 99.0297%, with processing time 1 minutes 7 seconds.

Overall our experiment shows that by using combination of feature selection and parameter optimization improved accuracy in the detection of DOS is 0.1155% whereas increased in speed is 53 minutes 35 seconds. While the accuracy in the detection of R2L is decreased by −0.6919% the increased in speed is quite satisfactorily, that is, 30 minutes 21 seconds.

VI. Conclusion

In order to detect DoS attacks there are 27 features that are most influential as shown in Table I. The work also finds out that to detect R2L attacks there are 27 of the most influential features as shown in Table II. The appropriate selection of parameters improves the accuracy in the detection of DoS attacks as well as increase the speed significantly the training and testing process. Even though limited and appropriate selection of parameters slightly decrease the accuracy in the detection of R2L attacks but our approach significantly increases the speed of the training and testing process

The future works is to study the use of all four classes attack type, DoS, Probe, U2L, and R2L. We plan also to employ chaos optimization algorithm to set the best parameters for the SVM machine learning and use the real network for testing dataset.

Acknowledgment

This research is partly funded by PITTA Grand of Universitas Indonesia under contract number 2128/UN2.R12/NKP/2016 and supported by Id-SIRTII/CC

(Indonesia Security Incident Response Team on Internet Infrastructure/Coordination Center), in the joint development of National Cyber Security Situational Awareness Systems. We gratefully appreciate this support and would like to thank Universitas Indonesia and Id-SIRTII/CC.

References

[1] H. Dreger. A. Feldmann, V. Paxson. And R. Sommer. “Operational Experiences with High-Volume Network Intrusion Detection”. In Proceedings of ACM Conference on Computer and Communications Security (CCS ’04), Washington, D.C., October 2004.

[2] Alimudin Akhmad, Hariadi Mochammad (2012). "Integrasi IDS menggunakan SVM dan KNN Dempster-Shafer". ITS JAVA Press.

[3] Netasan P, Balasubramanie P. (2012). “Multi Stage Filter Using Enhanced Adaboost for Network Intrusion Detection”, International Journal of Network Security & Its Applications (IJNSA), Tamilnadu, India.

[4] Mukkamala Srinivas, Guadalupe, Sung Andrew. (2002). “Intrusion Detection Using Neural Network and Support Vector Machines", IEEE.

[5] R. Agarwal and M. V. Joshi, “PNrule: A New Framework for Learning Classifier Models in Data Mining (A Case-Study in Network Intrusion Detection)”, Technical Report TR 00-015, Department of Computer Science, University of Minnesota, 2000.

[6] I. Levin, “KDD-99 Classifier Learning Contest LLSoft’s Results Overview”, SIGKDD Explorations, ACM SIGKDD, January 2000, Vol. 1 (2), pp. 67-75.

[7] D. Y. Yeung, and C. Chow, “Parzen-window Network Intrusion Detectors”, In Proceedings of the Sixteenth International Conference on Pattern Recognition, Quebec City, Canada, August 2002, Vol. 4, pp. 385-388.

[8] W. Lee, and S. Stolfo, “A Framework for Constructing Features and Models for Intrusion Detection Systems”, ACM Transactions on Information and System Security, November 2000, Vol. 3 (4), pp. 227-261.

[9] Manoj Sharma, Keshav Jindal, Aishish Kumar, “Intrusion Detection System using Bayesian Approach for Wireless Network”, International Journal of Computer Applications(0975-888), Volume 48-No.5, June 2012.

[10] Maheshkumar Sabhnani, Gursel Serpen, “KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection”, Security and Management, page 310-316. CSREA Press, (2003)

[11] GureKddcup database description,
http://www.aldapa.eus/res/gureKddcup/README.pdf

[12] KDD Cup 1999 Data
http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html