A Feature Selection Approach Based on Simulated Annealing for Detecting Various Denial of Service Attacks

In-Seon Jeong¹, Hong-Ki Kim², Tae-Hee Kim², Dong Hwi Lee²,
Kuinam J. Kim³ and Seung-Ho Kang⁴

¹*School of Electronics & Computer Engineering, Chonnam National University,
77 Yongbong-ro, Buk-gu, Gwangju, Republic of Korea 61186*

²*Department of Information Security, Dongshin University, 185 Geonjae-ro,
Naju, Jeonnam 58245, Republic of Korea*

³*Department of Convergence Security, Kyonggi University, 94-6 Yui-dong,
Yeongtong-gu, Suwon-si, Gyeonggi-do 16227, Republic of Korea*

⁴*Department of Information Security, Dongshin University, 185 Geonjae-ro,
Naju, Jeonnam 58245, Republic of Korea
E-mail: {jis0755; kinston}@gmail.com*

Received 25 February 2016; Accepted 27 March 2016;
Publication 16 April 2016

Abstract

Feature combinations affect network intrusion detection/prevention systems based on machine learning methods such as multi-layer perceptron (MLP) in terms of accuracy and efficiency. However, selecting the optimal feature subset from the list of possible feature sets to detect network intrusions requires extensive computing resources. In this paper, we propose an optimal feature selection algorithm based on the simulated annealing algorithm to determine six denial of service attacks (neptune, teardrop, smurf, pod, back, land). In order to evaluate the performance of our proposed algorithm, three well-known machine learning methods (multi-layer perceptron, Bayes classifier, and support vector machine) are used against the NSL-KDD data set.

Keywords: Network intrusion detection system, Machine learning, Feature selection, Simulated annealing algorithm, NSL-KDD trustworthy, cyber-physical, identity, locator.

1 Introduction

An incomparable amount of changes is taking place in the methods and volume of information distribution owing to the widespread application of computers and the exponential increase of wired and wireless networks. Especially, the application of information and computer technology to various industries definitely contributes to increased efficiency and productivity. However, the wide spread application of computers and the increase of networks also elevated the incidence of malicious activities such as information leakage and intrusions, resulting in economic losses and difficulty with spreading information technology. To tackle these problems, varying methods have been proposed especially in the area of network-based intrusion detection systems (IDS). Almost all of the IDS deployed in networks are signature-based IDS and use a set of simple rules. Despite some advantages such as high confidence in detection and low false positive rate, signature-based IDS have limits on detecting unknown attacks and the need for expert knowledge to create signatures. For this reason, machine learning based IDS has attracted many researchers as an alternative IDS approach [1–8].

One of the most important factors in developing IDS based on machine learning methods is finding feature sets to characterize and describe attacks in networks. Although various features have been extracted from network packets and system logs and many others have been proposed, a public feature data set for objective and fair comparison between proposed IDS is required by researchers. The KDD'99 data set was provided by MIT Lindon lab to fulfill this requirement [9]. Many researchers have used the KDD'99 data set to evaluate the performance of IDSs they proposed. However, owing to the disadvantages of the KDD'99 data set such as excessive data size, data redundancy and bias to certain attacks, the data set has limitations on using it without modification. To address the problems of the KDD'99 data set and provide a data set that can be used to carry out objective and fair performance comparisons, the NSL_KDD data set was proposed by Tavallaee et al. [10, 11].

Both the KDD'99 and NSL_KDD data sets use a total of 41 features to characterize and describe various attacks. However, a total of 41 features is not suitable as a descriptor for representing attacks and as the input

vector for machine learning methods such as multi-layer perceptron. Methods that use feature subsets relevant to specific attacks, therefore, have received considerable attention among many concerned researchers. In this respect, many methods based on the analysis of the correlations with attack classes such as the information gain [12], dependency ratio [13] and correlation [14] of individual features have been proposed. These methods eliminate features with lower ranks after ordering in terms of correlation measures. Although the correlation based methods guarantee efficiency, they cannot reflect the emergent effect of feature combinations, which is different from the naïve addition of individual features. The reason for almost all proposed feature selection methods dependent on the correlation analysis of individual features is that the number of possible feature subsets is too large to evaluate each feature subset through experimentation. For example, given a set of 41 features, the number of possible feature subsets is $2^{41}-1$.

A method based on the meta heuristic algorithm was proposed to tackle the optimal feature selection problem by Kang et al. [15]. The proposed optimal feature selection algorithm is based on a local search algorithm to provide a feature subset for multi-layer perceptron. The authors showed that the feature subsets selected by the approach guarantees above 95% accuracy and the average size of feature subsets is 21, half of 41 features. However, they encountered two class problems in determining whether denial of service (DoS) attacks occur from the selected feature subset without specifying the kind of DoS attacks. In view of the fact that the countermeasure should be different according to the kind of attack, the functionality to discern the kind of attack is an important ability that IDSs require. Therefore, the research has limitations for applications to network-based IDSs in practice.

In this paper, we propose an optimal feature selection algorithm to characterize the six kinds of DoS attacks (neptune, teardrop, smurf, pod, back, land) defined in the KDD'99 data set and the NSL_KDD data set in addition to normal traffic. The proposed method is based on the simulated annealing algorithm. In order to evaluate the performance of selected feature subsets in terms of accuracy and efficiency using the proposed algorithm, three well-known machine learning techniques, experiments using multi-layer perceptron (MLP), naïve Bayes classifier and support vector machine (SVM), are carried out against the NSL_KDD data set. Subsequently, we compare the performance of our proposed method with that of the feature selection method based on the local search algorithm.

The paper is arranged as follows. In the second section, the composition and properties of the NSL_KDD data set is described. A feature selection

algorithm based on the simulated annealing algorithm is proposed in Section 3. In Section 4, experiments using three machine learning methods are conducted to evaluate the performance of the selected feature subsets obtained by using the proposed feature selection method. A performance comparison is also carried out against the NSL_KDD data set in terms of accuracy and efficiency. Lastly, we conclude and present future research in Section 5.

2 Material

2.1 NSL_KDD Data Set

We used the NSL_KDD data set [11] to evaluate the usability of the proposed feature selection algorithm. The KDD'99 data set [9], which has been widely used to evaluate IDSs, is composed of training data containing about 5 million samples and test data containing about 300,000 samples. Attacks in the data set are categorized into 4 classes (denial of service attack, user to root attack, remote to local attack, probing attack) including normal traffic. Additionally, 41 features (refer to Table 1) are classified into 3 groups: basic

Table 1 The 41 features of NSL_KDD data set

No	Feature Name	No	Feature Name
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate'
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login		

feature, contents feature and traffic feature. Among 41 features, features like duration, protocol_type, and service are classified as basic features and they can be usually extracted from TCP/IP connections. Features such as num_failed_logins, logged_in, num_compromised and su_attempted are part of the contents feature. Contents features are relevant to the attributes which help to detect a suspicious behaviour such as login failure. Lastly, the traffic feature is computed through observing the network connection using a time window of 2 seconds and it is classified into 2 categories: same host features and same service features. While error_rate and rerror_rate are part of the same host feature, srv_error_rate and srv_rerror_rate are classified under the same service feature. The 41 features detailed in the NSL_KDD data set are presented in Table 1.

The fact that the complete KDD'99 data set is too large makes it difficult to use the data set to compare the performance of proposed methods without artificial manipulation such as arbitrary selection of part of the data set according to the author's subjective decision. In addition to the data size problem, the fact that the results of experiments could be biased towards the relatively abundant attack records has been revealed by many studies focusing on the KDD'99 data set itself.

To complement the disadvantages of the KDD'99 data set, M. Tavallaee et al. [10] proposed the NSL_KDD data set. While the NSL_KDD data set is basically a subset of KDD'99, it improves the KDD'99 data set as follows. Firstly, the NSL_KDD data set eliminates data redundancy such that it prevents the results of experiments from having a bias towards relatively redundant attack records. In addition, the NSL_KDD data set increases the objectivity of the performance comparisons by adjusting the difficulty levels between attack classes. Lastly, the NSL_KDD data set consists of a reasonable amount of records so that objective comparisons among different detection methods are possible while avoiding the arbitrariness that occurs when randomly selected parts of the data set are used.

Because the goal of this paper is to propose a feature selection method for determining which kind of DoS attack occurs in a network among six kinds of attacks, normal records and six kinds of DoS attack records were extracted from the complete NSL_KDD data set. The selected part of the data set is composed of training data and test data, containing 113271 records and 15452 records, respectively. The composition of the prepared data set for experiments including normal instances is shown in Table 2. From the table, we can observe large differences in the number of records for DoS attacks in the NSL_KDD data set.

Table 2 The composition of the data set

	Normal	Neptune	Teardrop	Smurf	Pod	Back	Land
Training data	67344	41214	892	2646	201	956	18
Test data	9711	4657	12	665	41	359	7

2.2 Data Preprocessing

NSL_KDD data set contains a range of feature types. Therefore, each feature needs to be normalized into a certain range of numeric values in order to be used as an input to the machine learning classifier. Data normalization was conducted following the method proposed in [2].

- 1) symbolic features like *protocol type* – integers from 0 to $N-1$, where N is the number of symbols, were assigned to each symbol and then each value was linearly scaled to the range of $[0, 1]$.
- 2) numeric features with large integer value ranges like *src_bytes* and *dst_bytes* – logarithmic scaling with base 10 was applied to the features.
- 3) boolean features – the corresponding value of 0 or 1 was used without any modification.
- 4) all other numeric features: linearly scaled to the range of $[0, 1]$.

After mapping to a certain range of numeric values, min-max normalization was applied to each scaled feature value. A feature value s is linearly transformed to a value in the range of $[0, 1]$ using (1)

$$\frac{s - \min(f_i)}{\max(f_i) - \min(f_i)}, \quad (1)$$

where, $\min(f_i)$ and $\max(f_i)$ denotes the minimum value and the maximum value of the i -th feature over the training and test data set ($1 \leq i \leq 41$).

3 Optimal Feature Selection Algorithm

3.1 Optimal Feature Subset Selection Problem

Kang et al. [15] defined the feature selection problem for IDS as a combinatorial optimization problem. The number of possible feature combinations from the feature set with 41 features is $2^{41}-1$. This number means that it is impossible to conduct performance evaluations for all feature combinations. The optimal feature subset selection problem is defined as follows:

Definition 1. Optimal feature subset selection problem

Given a feature set $f = \{f_1, f_2, f_3, \dots, f_n\}$ and a cost function $C: f \rightarrow q$ ($0 \leq q$), find feature subset (s) f' such that the value of the cost function is minimized.

3.2 Feature Selection Approach based on Simulated Annealing Algorithm

We designed a novel feature selection algorithm for IDS. The proposed feature selection algorithm is based on simulated annealing, which is a widely used method in combinatorial optimization. The simulated annealing algorithm can be considered one of the search algorithms. However, while naïve local search algorithms use a greedy approach to find the optimal solution, simulated annealing is a probabilistic technique that enables us to leave the local optima to find better solutions. For this reason, simulated annealing is known to behave better than the naïve local search algorithm most of the time.

3.2.1 Solutions

A solution used in the feature selection algorithm is represented by a binary vector f with a length of 41 as (2). While the value of 1 is assigned to the selected feature, 0 is assigned to the unselected feature.

$$f = \langle f_1, f_2, f_3, \dots, f_{41} \rangle, \text{ where } f_i \in \{0, 1\}, 0 \leq i \leq 41 \quad (2)$$

Most search algorithms used to handle the optimization problem need an initial solution. The simulated annealing approach needs an initial solution as well. We randomly selected a feasible solution and used it as an initial solution.

In the meantime, the neighboring solutions for a given solution are defined as binary vectors with one bit different from the given solution. For example, the neighboring solutions for a solution which uses all 41 features are composed of 41 binary vectors with only one bit of value 0. $\langle 0, 1, 1, 1, \dots, 1 \rangle$ is one of them.

3.2.2 Cost function

One of the important factors on which the performance of optimization heuristic algorithm such as simulated annealing depends is the cost function for evaluating individual solutions. In other words, the performance of an algorithm is largely dependent on how the cost function is defined. The cost function used in this paper is similar to the one suggested in [15]. The basic idea of the cost function is to use the accuracy of clustering by using features represented by the given solution. How accurately the training data is

partitioned into corresponding clusters when the clustering is conducted using only features presented by a solution serves as the cost for a given solution.

K-means clustering algorithm was adopted as a clustering algorithm for the cost function. The objective of the k -means clustering algorithm is to partition observation into k clusters such that the sum of the variation in the clusters is minimized. Because the paper dealt with classifying 6 kinds of DoS attacks and normal traffic, the feature selection problem belongs to the 7-class problem and the value of k is 7. Additionally, the cost φ for a given record x in the training data is computed by

$$\varphi(x) = \begin{cases} 1 & \text{if } p(x) = q(x) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$\varphi(x)$ is set to 1 if the class $p(x)$ determined by the clustering algorithm is equal to the original class $q(x)$ that a record x belongs to, otherwise it is set to 0. The cost $C(f)$ for a given solution f is calculated over the training data set (with a size of N) using (5).

$$C(v) = \frac{1}{\sum_{i=1}^N \varphi(x_i)} \quad (5)$$

3.2.3 Other parameters

Simulated annealing adopts a cooling scheme to find optimal solutions avoiding local optima while searching the solution space. In general, the cooling scheme refers to a schedule for how to search. Parameters such as an initial temperature, a temperature reduction function and a termination condition have to be specified.

The initial temperature T has to be large enough to allow sufficient transitions to be accepted. A value of 100,000 was assigned as the initial value T , which is larger than the size of the training data set. The temperature reduction function is defined as a simple iterative function which is the product of T multiplied by a constant r .

$$T \leftarrow r \times T \quad (6)$$

where the value of r is set to 0.9. Lastly, the termination condition is that if the value of T is less than 0.001 then the algorithm stops where 0.001 was determined after several experiments.

3.2.4 Procedure of algorithm

The feature selection algorithm based on simulated annealing proceeds as follows.

Firstly, an initial solution is selected randomly and it is assumed to be the optimal solution. Subsequently, the cost of the initial solution is computed using the cost function described previously. While temperature T does not satisfy the termination condition, a neighboring solution of the current optimal solution is selected and its cost is calculated. If the cost of the newly selected neighboring solution is less than or equal to that of the current optimal solution, the current optimal solution is replaced with a newly selected neighbor solution. If the cost of the neighboring solution is greater than the current optimal solution, a random value q is chosen in the range of $(0, 1)$. In this case, the replacement of the optimal solution is permitted only if a random value q is less than $e^{-\frac{Cost(v_n) - Cost(v_b)}{T}}$. After temperature T is reduced following (6), these processes are continued until T satisfies the termination condition.

A pseudo-code for the feature selection algorithm based on simulated annealing is suggested as follows.

Algorithm: Feature selection algorithm based on simulated annealing

Input: Training data set

Output: Combination of features: ν_b

1. $\nu_b \leftarrow \text{Null}$; // final solution
 2. $T \leftarrow 100000$;
 3. $r \leftarrow 0.9$;
 4. Generate an initial solution, ν_i ;
 5. $\nu_b \leftarrow \nu_i$;
 6. Calculate the cost of initial solution, $Cost(\nu_b)$;
 7. while ($T > 0.001$) do
 8. begin:
 9. Randomly select a neighbor solution, ν_n , of ν_b which have one bit different from ν_b ;
 10. if ($Cost(\nu_b) = Cost(\nu_n)$):
 11. $\nu_b \leftarrow \nu_n$;
 13. else:
 14. Generate a random number q uniformly in the range $(0, 1)$;
 15. if ($q < e^{-\frac{Cost(v_n) - Cost(v_b)}{T}}$)
 16. $\nu_b \leftarrow \nu_n$;
 17. $T \leftarrow r \times T$
 18. end // for while loop
-

4 Experiments and Results Analysis

We generated 20 feature subsets using the proposed feature selection algorithm for the performance evaluation of the selection algorithm.

4.1 Machine Learning Methods

In order to evaluate the selected feature subsets, three representative supervised machine learning methods, MLP, Bayes classifier and SVM, were used. A brief introduction about machine learning methods including the used parameters are presented in this sub section.

4.1.1 Multi-layer perceptron

Multi-layer perceptron, also known as an artificial neural network, is a supervised machine learning method that learns and recognizes objects by imitating the information process of the human brain. A 3-layer format containing input, hidden and output layers was adopted as the basic structure of MLP. The number of nodes at the input layer is equal to the size of the given feature subset and the number of nodes at the output layer is set to 7. The number of nodes at the hidden layer was set up with a number which showed the best performance through many experiments. The sigmoid function was used as a neuron function and MLP was learned using the backpropagation algorithm with various learning rates and momentums.

$$S(v) = \frac{1}{1 + e^{-v}} \quad (7)$$

The learning algorithm terminated when the improvement with respect to error rate was less than 0.1% between two contiguous epochs over the entire training data set.

4.1.2 Naïve Bayes classifier

The naïve Bayes classifier [16] is a supervised machine learning classifier based on the following Bayes theorem.

$$p(\omega_i|x) = \frac{p(x|\omega_i)p(\omega_i)}{p(x)} \quad (8)$$

where w_i indicates the i^{th} class and x denotes a given feature vector. In the naïve Bayes classifier, the class of the given feature is determined by the class with the greatest posterior probability $p(x|w_i)$. We assumed that the distribution of

$p(x|w_i)$ follows the Gaussian distribution and the maximum likelihood method was used to determine the parameters of the distribution.

4.1.3 Support vector machine

Support vector machine (SVM) is one of the supervised machine learning methods suggested by Vapnik [17]. Because it considers the margin between support vectors when finding the decision hyperplane, it is known to have generality compared with other machine learning methods.

We used a polynomial kernel function $K(x, y) = (x \cdot y + 1)^p$ for the non-linear support vector machine. Lagrange multipliers were obtained by a sequential minimal optimization (SMO) algorithm [18]. Originally, SVM was a 2-class classifier. The 2-class SVM for solving multi-class problems that deal with more than two classes had to be expanded. The pair-wise classification method that adopts voting to identify the given feature was used.

4.2 Performance Comparisons

In order to evaluate the performance of the feature subset computed by the proposed selection algorithm, the accuracy, which is generally evaluated in studies on IDS, was measured. The accuracy is the proportion of correct results in determinations by the machine against the test data set. In addition to the accuracy, the time taken to train and test and the size of the feature subset were measured as well. These measures are important evaluation factors for real time IDS/IPSSs.

Table 3 presents the average accuracy and standard deviation when three machine learning methods were applied to 20 feature subsets obtained by two feature selection algorithms, one based on the local search algorithm and the other based on simulated annealing. In the same table, the accuracy of all 41

Table 3 The results of experiments which show accuracy, time taken for training, time taken for testing and feature size

	Accuracy(%)		
	Multi-Layer Perceptron	Naïve Bayes Classifier	Support Vector Machine
Local search algorithm	96.77 (± 1.52)	81.30 (± 19.12)	97.15 (± 2.13)
Simulated annealing	96.83 (± 0.90)	86.64 (± 14.40)	97.48 (± 1.38)
All features	96.98	98.33	99.24

features when they were used in the three machine learning methods is also shown.

The accuracy of the feature subsets obtained by the proposed feature selection algorithm was 96.83% for MLP, 96.64% for the Bayes classifier and 97.48% for SVM. Although these accuracy values are lower than those obtained when all features were used, they are slightly higher than those obtained with the local search based feature selection algorithm. In the case of the naïve Bayes classifier, while the simulated annealing algorithm showed 5% higher accuracy than the local search algorithm, it was quite a bit lower than the accuracy when all 41 features were used. However, the best accuracy values achieved by the proposed algorithm for the three machine learning methods among 20 feature subsets were 98.74%, 98.69% and 99.24% respectively, which were higher than or equal to those achieved when 41 features were used. This means that if we seek many solutions with the proposed algorithm and select the solutions with high performance, we can achieve accuracy as high as that obtained when 41 features are used for machine learning techniques regardless of the types. Table 4 shows the feature subsets which achieved the best accuracy values among the 20 feature subsets used for three machine learning techniques with their accuracy values.

Table 5 shows the average lengths of feature subsets obtained by two feature selection algorithms using MLP. It also shows the time required for training and the time for testing. The average length of feature subsets produced by the proposed feature selection algorithm is 18 (± 3.74) and is lower than that of feature subsets obtained by the feature selection algorithm based on a local search algorithm. As we can expect, the time taken to train and test the proposed method is shorter than that of the local search based algorithm. The length of the feature vector and especially the time taken to determine whether an attack occurs or not, even though it is trivial, is as important as the accuracy in realtime IDS/IPSSs. The best solution in terms of

Table 4 A feature set with accuracy of 98.8%

	Feature Composition	Accuracy
MLP	0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0	98.74
Bayes	1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0	98.69
SVM	0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1	99.24

Table 5 The average lengths of feature subsets

	Length of Feature Vector	Time for Training	Time for Testing
Local search algorithm	19.05 (± 3.47)	315.99 (± 49.31)	0.71 (± 0.12)
Simulated annealing	18.00 (± 3.74)	296.79 (± 46.19)	0.64 (± 0.12)
All features	41	799.65	1.48

the time had a length of 14 and the time taken to train and test was 233.11 sec and 0.49 sec, respectively.

5 Conclusion

In this paper, we proposed an optimal feature selection algorithm for detecting six kinds of denial of service attacks against the NSL_KDD data set. The feature selection problem was defined as a combinatorial optimization problem. The proposed algorithm is based on the simulated annealing algorithm.

In order to evaluate the accuracy and efficiency of selected feature subsets obtained by the proposed feature selection algorithm, MLP, Bayes classifier and SVM were used against the NSL_KDD data set. A comparison between our proposed algorithm and other feature selection algorithms was conducted including a comparison with the results obtained when all 41 feature sets were used. From the experiment results, we confirmed that the feature subsets selected by the proposed algorithm have a higher accuracy and detection rate. In addition, the average length of the feature subsets obtained by the proposed algorithm was 18 and the algorithm was more efficient in both learning and identifying time. This indicates that the proposed feature selection algorithm is suitable for realtime IDS/IPS.

References

- [1] Paliwal, S., and Gupta R. (2012). Denial-of-service, probing & remote to user (R2L) attack detection using genetic algorithm. *Int. J. Comput. Appl.* 60, 57–62.
- [2] Sabhnani, M., and Serpen, G. (2003). Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. *Proc. Int. Conf. Mach. Learn. Model Technol. Appl.* 209–215.

- [3] Bankovic, Z., Stepanovic, D., Bojanic, S., and Nieto-Taladriz, O. (2007). Improving network security using genetic algorithm approach. *Comput. Electr. Eng.* 33, 438–451.
- [4] Azad, C., and Jha, V. K. (2013). Data mining in intrusion detection: a comparative study of methods, types and data sets. *Int. J. Inf. Technol. Comput. Sci.* 5, 75–90.
- [5] Balajinath, B., and Raghavan, S. V. (2001). Intrusion detection through learning behavior model. *Comput. Commun.* 24, 1202–1212.
- [6] Tsai, C. F., Hsu, Y. F., Lin, C. Y., and Lin, W. Y. (2009). Intrusion detection by machine learning. *Rev. Expert Syst. Appl.* 36, 11994–12000.
- [7] Wu, S. X., and Banzhaf, W. (2010). The use of computational intelligence in intrusion detection system. *Rev. Appl. Soft Comput.* 10, 1–35.
- [8] Kolias, C., Kambourakis, G., and Maragoudakis, M. (2011). Swarm intelligence in intrusion detection: a survey. *Comput. Secur.* 30, 625–642.
- [9] KDD Cup. (1999). Available at: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
- [10] Tavallae, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (1999). “A Detailed Analysis of the KDD CUP 99 Data Set,” in *CISDA’09 Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications*, Ottawa, ON (NJ, USA: IEEE Press Piscataway), 53–58.
- [11] NSL-KDD data set. Available at: <http://nsl.cs.unb.ca/NSL-KDD/>
- [12] Kayacik, H. G., Zincir-Heywood, A. N., and Heywood, M. I. (2005). “Selecting features for intrusion detection: a feature relevance analysis on kdd 99 intrusion detection datasets,” in *Thrid Annual Conference on Privacy, Security and Trust*.
- [13] Olusola, A. A., Oladele, A. S., and Abosede, D. O. (2010). “Analysis of KDD’99 intrusion detection dataset for selection of relevance features,” in *Proceedings of the World Congress on Engineering and Computer Science*, Vol. 1.
- [14] Parazad, S., Saboori, E., and Allahyar, A. (2012). “Fast feature reduction in intrusion detection datasets,” in *MIPRO, Proceedings of the 35th International Convention* pp.1023–1029.
- [15] Kang, S.-H., and Kim, K. J. (2015). A feature selection approach to find optimal feature subsets for the network intrusion detection system. *Cluster Comput.* doi: 10.1007/s10586-015-0527-8

- [16] John, G. H., and Langley, P. (1995). “Estimating continuous distributions in Bayesian classifiers,” in *UAI’95 Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*.
- [17] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* 2, 121–167.
- [18] Platt, J. C. (1998). *Sequential minimal optimization: a fast algorithm for training support vector machines*. Technical Report MSR-TR-98-14.

Biographies

I.-S. Jeong received her M.S. and Ph.D. degree in computer science from the Chonnam National University, Gwangju, Korea, in 2006 and 2011. During 2011–2015, she was a postdoctoral researcher in genomics division at Rural Development Administration, Korea. Her research interests include machine learning, data mining, algorithms in bioinformatics, and sensor networks.

H.-K. Kim is a professor of Information Security Department in the Dongshin University, Naju, Korea. He received his M.S. and Ph.D. in Computer Science from Chonnam National University, Gwangju, Korea, in 1986 and 1996, respectively. His research interests include information security, spatial data structure, and graphics.

T.-H. Kim received her M.S. and Ph.D. in Computer Science from Chonnam National University, Gwangju, Korea, in 1991 and 1999, respectively. During 1993–1997, she was a part time lecturer at Dongshin University, Naju, Korea. She joined Dongshin University, Naju, Korea, in 1998, where she works as an associate professor. Her research interests include information security, security programming, and database security.

D. H. Lee received the B.S. degree in Computer Science from Kyonggi University, Korea. He received M.S. and Ph.D degree in Information Security from Kyonggi University, Korea. and Research Scholar of University of Colorado Denver, USA, in 2011 and 2012. He is currently a assistant Professor in Information Security, Dongshin University, Korea.

His research areas include Information Security and Convergence security.

K. J. Kim is a professor of Information Security Department in the Kyonggi University, Korea. He received his Ph.D and MS in Industrial Engineering from Colorado State University in 1994. His B.S in Mathematics from the

University of Kansas. He is Executive General Chair of the Institute of Creative and Advanced Technology, Science, and Engineering. His research interests include cloud computing, wireless and mobile computing, digital forensics, video surveillance, and information security. He is a senior member of IEEE.

S.-H. Kang received his M.S. and Ph.D. in Computer Science from Chonnam National University, Gwangju, Korea, in 2003 and 2009, respectively. During 2010–2013, he was a researcher in the National Institute for Mathematical Science, Daejeon, Korea. He joined Dongshin University, Naju, Korea, in 2013, where he works as an assistant professor. His research interests include information security, wireless sensor networks, and algorithm.