Expert Systems with Applications 41 (2014) 1690–1700

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A novel hybrid intrusion detection method integrating anomaly detection with misuse detection

Gisung Kim [a], Seungmin Lee [b,*], Sehun Kim [a]

[a] Institute for IT Convergence, KAIST, Guseong-dong, Yuseong-gu, Daejeon 305-701, South Korea
[b] Future Research Creative Laboratory, ETRI 218 Gajeong-ro, Yuseong-gu, Daejeon 305-700, South Korea

ARTICLE INFO

Keywords:
Hybrid intrusion detection
One-class SVM
Anomaly detection
Decision tree

ABSTRACT

In this paper, a new hybrid intrusion detection method that hierarchically integrates a misuse detection model and an anomaly detection model in a decomposition structure is proposed. First, a misuse detection model is built based on the C4.5 decision tree algorithm and then the normal training data is decomposed into smaller subsets using the model. Next, multiple one-class SVM models are created for the decomposed subsets. As a result, each anomaly detection model does not only use the known attack information indirectly, but also builds the profiles of normal behavior very precisely. The proposed hybrid intrusion detection method was evaluated by conducting experiments with the NSL-KDD data set, which is a modified version of well-known KDD Cup 99 data set. The experimental results demonstrate that the proposed method is better than the conventional methods in terms of the detection rate for both unknown and known attacks while it maintains a low false positive rate. In addition, the proposed method significantly reduces the high time complexity of the training and testing processes. Experimentally, the training and testing time of the anomaly detection model is shown to be only 50% and 60%, respectively, of the time required for the conventional models.

1. Introduction

An intrusion detection system (IDS) has been developed that is capable of detecting all types of network attacks in the available environments. The IDS is placed inside the network that it protects and it collects network packets promiscuously in the same manner as a network sniffer. The IDS detects malicious network activities by analyzing the collected packets, alarms to system administrator, and blocks attack connections in order to prevent further damage from attacks. It also connects with the firewall as a fundamental technology for network security.

Generally, intrusion detection algorithms are categorized into two methods: misuse detection and anomaly detection (Depren, Topallar, Anarim, & Ciliz, 2005). Misuse detection algorithms detect attacks based on the known attack signatures. They are effective in detecting known attacks with low errors. However, they cannot detect newly created attacks that do not have similar properties to the known attacks. In contrast, anomaly detection algorithms analyze normal traffic and profile normal traffic patterns. The anomaly detection method is based on the hypothesis that the attacker behavior differs to that of a normal user. They classify traffic as an attack if the characteristics of the traffic are far from those of normal traffic patterns. Anomaly detection algorithms can be useful for new attack patterns, but they are not as effective as misuse detection models in the detection rate for known attacks and false positive rates, which is a ratio of misclassified normal traffic.

In order to resolve the disadvantages of these two conventional intrusion detection methods, hybrid intrusion detection methods that combine the misuse detection method and the anomaly detection method have also been proposed (Depren et al., 2005). Because none of the misuse and anomaly detection methods are better than any other, a hybrid intrusion detection system uses both the misuse detection method and anomaly detection method. The detection performance of the hybrid intrusion detection system depends on the combination of these two different detection methods. Most hybrid detection systems independently train a misuse detection model and an anomaly detection model, and then simply aggregate the results of the detection models. For example, hybrid intrusion detection systems regard a traffic connection as an attack if at least one of the two models classifies the traffic connection as an attack. In this case, the detection rate will be improved but the IDS will still have a high false positive rate. In contrast, if the hybrid method regards a traffic connection as an attack only if both models classify the connection as an attack, false alarms will be reduced but it may overlook many attack connections.

* Corresponding author. Tel.: +82 42 860 1775; fax: +82 42 860 6504.
E-mail addresses: todtom@etri.re.kr, brightdad@gmail.com (S. Lee).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.08.066

In this research, a new hybrid intrusion detection method is proposed that hierarchically integrates a misuse detection model and an anomaly detection model, rather than just combining the their results as in previous hybrid methods (Depren et al., 2005; Zhang & Zulkernine, 2006). In the proposed method, the anomaly detection model can indirectly use the known attack information throughout the integration in order to enhance its ability to build profiles of normal behavior. Generally, only the misuse detection method uses the known attack information to build a classifier and the anomaly detection method builds a classifier only based on normal traffic information. The proposed hybrid method also follows this general principle, but it is proposed that the normal training data is decomposed into disjoint subsets using the misuse detection model and then an anomaly detection model is built for each disjoint normal training data subset.

The entire normal data set has various types of normal connections, so an anomaly detection model cannot profile it precisely, which leads performance deterioration (Song, Takakura, Okabe, & Kwon, 2009). In the proposed hybrid intrusion detection process, each area for the decomposed normal data set does not have known attacks and includes less variety of connection patterns than the entire normal data set. An anomaly detection model for each normal training data subset can profile more innocent and concentrated data so that this decomposition method can improve the profiling performances of the normal traffic behaviors.

In this paper, the C4.5 decision tree (DT) is used to create the misuse detection model and the one-class support vector machine (1-class SVM) is used to create multiple anomaly detection models. In order to implement the concept described above, the DT model is first trained based on a training data set consisting of normal traffic and known attack traffic information, and then a 1-class SVM model is trained for each normal training data subset decomposed by the DT model. The proposed hybrid intrusion detection method was evaluated by conducting experiments using the NSL-KDD data set, which is a modified version of the famous KDD Cup 99 data set (Tavallaee et al., 2009). The experiment results demonstrate that the proposed method is better in terms of detection rates for unknown and known attacks than the conventional methods that independently train the DT model and 1-class SVM model without the proposed decomposition technique.

In addition to improving the detection rates, through training the anomaly detection model in each decomposed data set, the proposed method can reduce the high time complexity of the training and testing processes. Time consumption is the cost for updating the detection model. In particular, the testing time should be minimized in order to reduce the overhead of the detection algorithm in order to operate the detection model in real time. As the training data set is decomposed into smaller subsets, the training and testing times are significantly reduced. The experiments in this research demonstrate that the training and testing times of the anomaly detection model are only 50% and 60%, respectively, of that of the conventional models.

The remainder of this paper is organized as follows. In Section 2, the existing hybrid intrusion detection methods are reviewed. The detailed process of the proposed method is presented and the properties of the method are discussed in Section 3. In Section 4, the performance of the proposed method is evaluated in terms of the detection rate and time required for training and testing the anomaly detection model. The study concludes in Section 5 with a summary of the research undertaken and plans for future research.

2. Related works

There has been much research on hybrid intrusion detection methods in attempts to overcome the limits of the anomaly detection and misuse detection methods. The research has used three different methods to combine the anomaly detection model and misuse detection model: anomaly detection followed by misuse detection, parallel use of anomaly detection and misuse detection, and misuse detection followed by anomaly detection.

Barbara, Couto, Jajodia, Popyack, and Wu (2001) proposed the Audit Data Analysis and Mining (ADAM) method where the anomaly detection is followed by the misuse detection. ADAM uses a combination of association rule mining and a classification method to detect attacks. First, the anomaly detection model that uses association rule mining locates suspicious traffic connections and passes the connections to the misuse detection model. Then, the misuse detection model classifies the suspicious connections as normal (false alarm of the anomaly detection model), known attacks connections, and unknown attack connections. It is unusual that ADAM uses a misuse detection method to detect unknown attack connections. In the ADAM method, connections that cannot be confidently classified as normal or known attacks are classified as unknown attacks. In order to use the anomaly detection followed by the misuse detection, the anomaly detection model should have a high detection rate and the misuse detection model should remove the false alarms of the anomaly detection model by distinguishing the normal and unknown attacks. However, most misuse detection methods are not suitable for reducing false alarms.

Parallel hybrid approaches use an anomaly detection model and a misuse detection model in parallel. Depren et al. (2005) suggested an intelligent hybrid intrusion detection system that consists of an anomaly detection model, a misuse detection model, and a decision support system. They modeled the anomaly detection model with a self-organization map (SOM) and the misuse detection model with a decision tree. Each model is trained independently, and then the decision support system simply combines the classification results of both models.

Anderson, Frivold, and Valdes (1995) developed the Next Generation Intrusion Detection Expert System (NIDES) and it uses a rule-based analysis model and statistical analysis model. The rule-based misuse detection model employs expert rules to define the known attacks and the statistical analysis anomaly detection model detects connections that depart from the established patterns of normal behavior. In this method, the detection rate of known attacks and unknown attacks is enhanced. However, the high false positive property of the anomaly detection method remains because it regards an incoming connection as an attack if any detection model classifies the connection as an attack. Also, there is a detection overhead problem because every connection should be checked using both the anomaly detection model and the misuse detection model, and this can cause increases in the detection overhead.

Zhang and Zulkernine (2006) and Hwang, Chen, and Qin (2007) used the misuse detection method followed by the anomaly detection method to design a hybrid intrusion detection system. Because the misuse detection model can detect known attacks with a low false positive rate and can operate faster than the anomaly detection model, the misuse detection model is used first to detect the known attacks and then the anomaly detection model is only applied to the remaining uncertain connections. The anomaly detection model detects outliers that depart from the normal data patterns and classifies them as unknown attacks. Zhang and Zulkernine (2006) also developed a weighted signature generation scheme that extracts the attack signatures from the attack connections detected by the anomaly detection model and adds those signatures to the misuse detection model for fast and accurate operation. However, as with the parallel hybrid method, the anomaly detection model and the misuse detection model are trained independently, which nevertheless results in a high false positive rate.

The critical technique of the anomaly detection method is the construction of the normal profiles. If the profiles are too broad, it fails to detect some attacks, which leads to a low detection rate. If the profiles are too narrow, then it can detect almost all attacks but many normal connections are also classified as attacks. This is an inherit trade off property of the anomaly detection method, but none of the previous research has attempted to resolve this problem. The current research focuses on reducing the false positive rate of the anomaly detection model by suggesting a new hybrid intrusion detection method. While the previous research only combines the results of both detection models, in the proposed method these detection models are hierarchically integrated in a decomposition structure. This enables the anomaly detection model to enhance its normal profiling ability by using each misuse detection model. This assists in alleviating the technical trade off properties of the anomaly detection method. The details of the proposed method are presented in the next section.

3. Proposed hybrid intrusion detection method

In this section, the DT and 1-class SVM algorithms that are required in order to build the misuse detection model and anomaly detection model, respectively, are briefly introduced. Then, the integration of these models is explained and the properties of the proposed hybrid intrusion detection method are discussed.

3.1. Decision tree and C4.5

A decision tree (DT) is one of the most widely used classification algorithms in data mining. It operates in a divide and conquer manner, which recursively partitions the training data set based on its attributes until the stopping conditions are satisfied. The DT consists of nodes, edges, and leaves. A DT node has its corresponding data set; this specifies the attribute to best divide the data set into its classes. Each node has several edges that specify possible values or value ranges of the selected attributes on the node. The data set of the node is divided into subsets according to the specifications of the edges, and the DT creates a child node for each data subset and repeats the dividing process. When the node satisfies the stopping rules because it contains homogeneous data sets or no future distinguishing attributes can be determined, the DT terminates the dividing process and the node is labeled as following the class label of the data set. This labeled node is called a leaf node. In this way, the DT recursively partitions the training data set, which creates a tree-like structure.

Quinlan popularized the decision tree approach (Quinlan, 1996). The latest public domain implementation of Quinlan's model is C4.5. The primary issue of the decision tree algorithms is to locate the attribute that best divides the data into their corresponding classes. C4.5 builds decision trees from training data sets using the concept of information entropy. That is, it is based on the highest gain of each attribute. The gain is calculated using the following formula:

$$\mathrm{Gain}(S,A) = \mathrm{Entropy}(S) - \sum_{i=1}^{n} f_s(A_i) \times \mathrm{Entropy}(S_{A_i}), \tag{1}$$

where $\mathrm{Gain}(S,A)$ is the gain of set S after a split over the A attribute; $\mathrm{Entropy}(S)$ is the information entropy of set S; n is the number of different values of attribute A in S; A is the proportion of items possessing A_i as the value for A in S; A_i is the ith possible value of A; and S_{A_i} is a subset of S containing all items where the value of A is A_i.

Here, the entropy is obtained as follows:

$$\mathrm{Entropy}(S) = -\sum_{j=1}^{m} f_S(j) \times \log_2 f_S(j), \tag{2}$$

where m is the number of different values of the attribute in S (entropy is computed for one chosen attribute) and $f_S(j)$ is the proportion of the value j in the set S.

After the tree is created by maximizing the gain, the C4.5 model decomposes the data space such that certain decomposed regions become homogeneous. Then, C4.5 performs the final pruning step. This step reduces the classification errors caused by specializations in the training set; thus, it makes the tree more general.

In this study, the C4.5 is used to train the misuse detection model in the hybrid intrusion detection system. Both normal and attack data are used to train the model: C4.5 divides the data into decomposed regions and labels the regions as the classes of major data belonging to each decomposed region.

3.2. One-class support vector machine (1-class SVM)

A one-class support vector machine (1-class SVM) is a popular outlier detection algorithm in various applications such as document classification (Manevitz & Yousef, 2002), machine fault detection (Shin, Eom, & Kim, 2005), and so on. The 1-class SVM was proposed by Schölkopf, Platt, Shawe-Taylor, Smola, and Williamson (2001) and was inspired by the general SVM classifier. It is formulated to locate a hyper plane that separates a desired fraction of the training one-class instances from the origin in the feature space (F). Then, this hyper plane is used to detect outliers of a testing instance by determining to which class the instance belongs.

Let $x_1, x_2, \ldots, x_l \in \chi$ be the training data instances belonging to original space χ and l be the number of instances. The 1-class SVM may be viewed as a regular binary SVM where all training data lies in the first class and the origin belongs to the second class. It finds the maximal margin hyper plane that best separates the training data from the origin (Schölkopf et al., 2001). Because it is usually difficult to locate a hyper plane that creates training data patterns separable from the origin in the original space χ, the SVM uses a feature map (Φ: $\chi \to F$), which non-linearly transforms the data from the original space to the feature space in order to locate the hyper plane in the feature space. The 1-class SVM also considers a trade off between maximizing the distance of the hyper plane from the origin and the fraction of data instances contained in the separated region (Perdisci, Gu, & Lee, 2006). This is controlled by the parameter v, which represents the fraction of training instances that can be rejected. The 1-class SVM is formulated as the following quadratic program:

$$\begin{aligned} \min_{w,\xi,\rho} \quad & \frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_{i=1}^{l}\xi_i - \rho \\ \text{subject to} \quad & (w \cdot \Phi(x_i)) \geqslant \rho - \xi_i, \\ & \xi_i \geqslant 0, \quad i = 1, \ldots, l \end{aligned} \tag{3}$$

where w is the vector orthogonal to the hyper plane, $\xi = [\xi_1, \ldots, \xi_l]$ is the vector of slack variables used to penalize the rejected instances, and ρ represents the margin, i.e. the distance of the hyper plane from the origin.

Because computing in the feature space is difficult due to the curse of dimensionality (Manevitz & Yousef, 2002; Shin et al., 2005), the SVM utilizes the kernel theory: the inner product in the feature space can be computed using a simple kernel function $k(x,y) = \Phi(x) \cdot \Phi(y)$, such as the Gaussian kernel, $k(x,y) = e^{-\gamma\|x-y\|^2}$. By applying the kernel theory and Lagrangian multiplier (α_i) to the original quadratic program, the solution of Eq. (3) creates a decision function. For a generic test instance (z), it is formulated as follows:

$$f(z) = \mathrm{sgn}\sum\nolimits_{i=1}^{l}(\alpha_i k x_i(z-\rho)) \tag{4}$$

The test instance (z) is accepted when $f(z)$ is positive and it is rejected when $f(z)$ is negative. An acceptance indicates that the test instance (z) is considered to be similar to the training data set, and a rejection indicates that it departs from the training data and is considered as an outlier.

In a 1-class SVM model, there are some parameters that affect the characterization of the decision boundary of the training data set. The parameter v controls the fraction of training instances that are allowed to be rejected, which means that the decision hyper plane only contains $(1 - v) \times 100\%$ of training instances. If v is high, the 1-class SVM model only focuses on the most frequent training patterns. In contrast, if v is very low, the decision hyper plane contains most training instances, including noisy data. The feature mapping also has a significant effect on the decision boundary. The degree of polynomial kernel and the width parameter of the Gaussian kernel control the flexibility of the resulting decision boundary. When using the Gaussian kernel ($k(x, y) = e^{-\gamma\|x-y\|^2}$), if the width of parameter γ is small, the SVM model loses non-linear power and the decision boundary tends to be smooth. If γ is large, the decision boundary of the SVM model tends to be highly sensitive to the training data, which lacks regularization. Therefore, it is important to determine appropriate parameters that consider the classification performance.

When applying the 1-class SVM to the anomaly detection, the anomaly detection model is trained using the normal training traffic dataset from 1-class SVM. Throughout the training procedure, the model locates decision boundaries that separate the normal data from the origin. When it inspects the incoming traffic connections, it detects outliers using the decision function of the model and it classifies the outliers as attack connections.

3.3. Proposed hybrid intrusion detection method

The proposed hybrid intrusion detection method is as follows. First, a DT model based on the training data set is built. It is well known that a misuse detection model can detect known attacks with a small false positive rate, while it cannot detect unknown attacks well. Because the false positive rate of the DT model is low, the results of the attack detections of the DT model are followed.

Then, a 1-class SVM model is trained for each normal training data set, which is decomposed by the DT model. The primary reason for decomposing the normal training data set is that the anomaly detection models in the previous hybrid intrusion detection methods have attempted to profile the normal connection patterns using one outlier detection model. However, in reality, there are various normal patterns according to the protocol type (TCP, UDP, ICMP, etc.), service type (HTTP, FTP, SNMP, etc.), and so on. Although a 1-class SVM model is appropriate for creating a non-linear decision boundary, the 1-class SVM model can be very sensitive to the training data set and can increase the false positive rate. In order to alleviate this problem, the normal data set is decomposed into smaller subsets and then multiple 1-class SVM models are built for the decomposed subsets. Because the data patterns of each decomposed subset are less complex than those of the whole data set, multiple models for each decomposed data pattern can be less flexible than a single model for the whole data pattern. The training process is described in Table 1.

As mentioned above, the decision boundaries of 1-class SVM models using a Gaussian kernel are controlled by the parameter γ. Figs. 1 and 2 show the changes in the decision boundaries in the proposed method (DT-SVM) and the conventional one-class SVM (SVM), with γ parameters varying from 0.0001 to 0.001. When γ is 0.0001, the conventional SVM model almost loses its power to detect attacks because it profiles the normal data too broadly. As γ increases, the decision boundary becomes more flexible. When γ is 0.001, the conventional SVM model can detect unknown attacks, but it may have a high false positive rate because it appears to profile the normal data too narrowly. Thus, it appears appropriate to set γ to 0.003 or 0.004 considering both the detection rate and the false positive rate. In the conventional SVM, the decision boundary varies significantly so that it appears to be sensitive to γ. The decision boundary of the DT-SVM model also changes to be flexible as γ increases, but the DT-SVM model focuses more on the normal data in the smaller region because each 1-class SVM model of the decomposed region profiles only its corresponding normal patterns. Hence, it is expected that the DT-SVM model can detect attacks with a lower false positive rate compared with the conventional 1-class SVM model.

The training time and testing time of the anomaly detection model are also improved using the tree decomposition. The time complexity of training a 1-class SVM model is in the order of n^2, where n is the number of training instances (Chang, Guo, Lin, & Lu, 2010). If each decomposed problem handles σ training instances, then the complexity of solving the entire problem is in the order of $(n/\sigma) \times \sigma^2 = n\sigma$, which is significantly smaller than n^2. Although the distribution of the number of data instances in each decomposed region is not uniform, the decomposition can reduce the training time. When it is considered that the training time is a cost for updating the detection model, reducing the training time provides more opportunities to update the detection model. The testing process and diagram of the proposed method are described in Table 2 and Fig. 3, respectively.

The testing time is directly related to the intrusion detection performance. Anomaly detection systems are often designed for offline analyses due to the expensive processing and memory overheads. Hence, the time and memory complexity of the anomaly detection model for testing should be minimized in order to operate the detection model in real time. In this paper, the decomposition concept can contribute to reducing the testing time of the anomaly detection model in real time operations. When a 1-class SVM model classifies the test instances, the most time consuming work is the computation of a decision function (Chang et al., 2010). The complexity of the decision function is measured using the number of encountered support vectors because the number of support vectors dominates the complexity of the computation. In the proposed method, the test instances are classified by one of the 1-class SVM models in a decomposed region. Because the number of support vectors of each SVM model in the decomposed region is less than those of the conventional SVM model, the decomposition can reduce the testing time when performing the anomaly detection.

4. Experiments

In this section, the effectiveness of the proposed method is evaluated carefully through experiments using the NSL-KDD data set, which is a modified version of well-known KDD'99 data set. Because the KDD'99 data set has an inherent problem of a number of redundant instances existing in the training and testing data set, NSL-KDD data set was proposed by removing all redundant instances and reconstituting the data set, which makes it more efficient to have an accurate evaluation of different learning techniques (Tavallaee, Bagheri, Lu, & Ghorbani, 2009). Weka 3.6 (Hall et al., 2009) and LibSVM (from MATLAB) (Chang & Lin, 2011) are used to evaluate the performance of the proposed method.

In this paper, the evaluation data set is organized by modifying the KDDTrain+.TXT and KDDTest+.TXT documents in the NSL-KDD data set. Each of these documents consists of traffic records including information of the traffic features and a connection label. Because the connection label specifies the attack type and KDDTest+.TXT contains some attack types that do not exist in

Table 1
Training process of the proposed hybrid intrusion detection method.

Training process of the proposed hybrid intrusion detection method	
Step 1	Prepare a training data set consisting of normal data and known attack data
Step 2	Build a misuse detection model using a decision tree algorithm based on the training data set
Step 3	Decompose the normal training data into subsets according to the decision tree structure Data in the same leaf belong to the same subset
Step 4	For each normal leaf of the decision tree, build an anomaly detection model using the 1-class SVM algorithm based on a normal data subset for the leaf

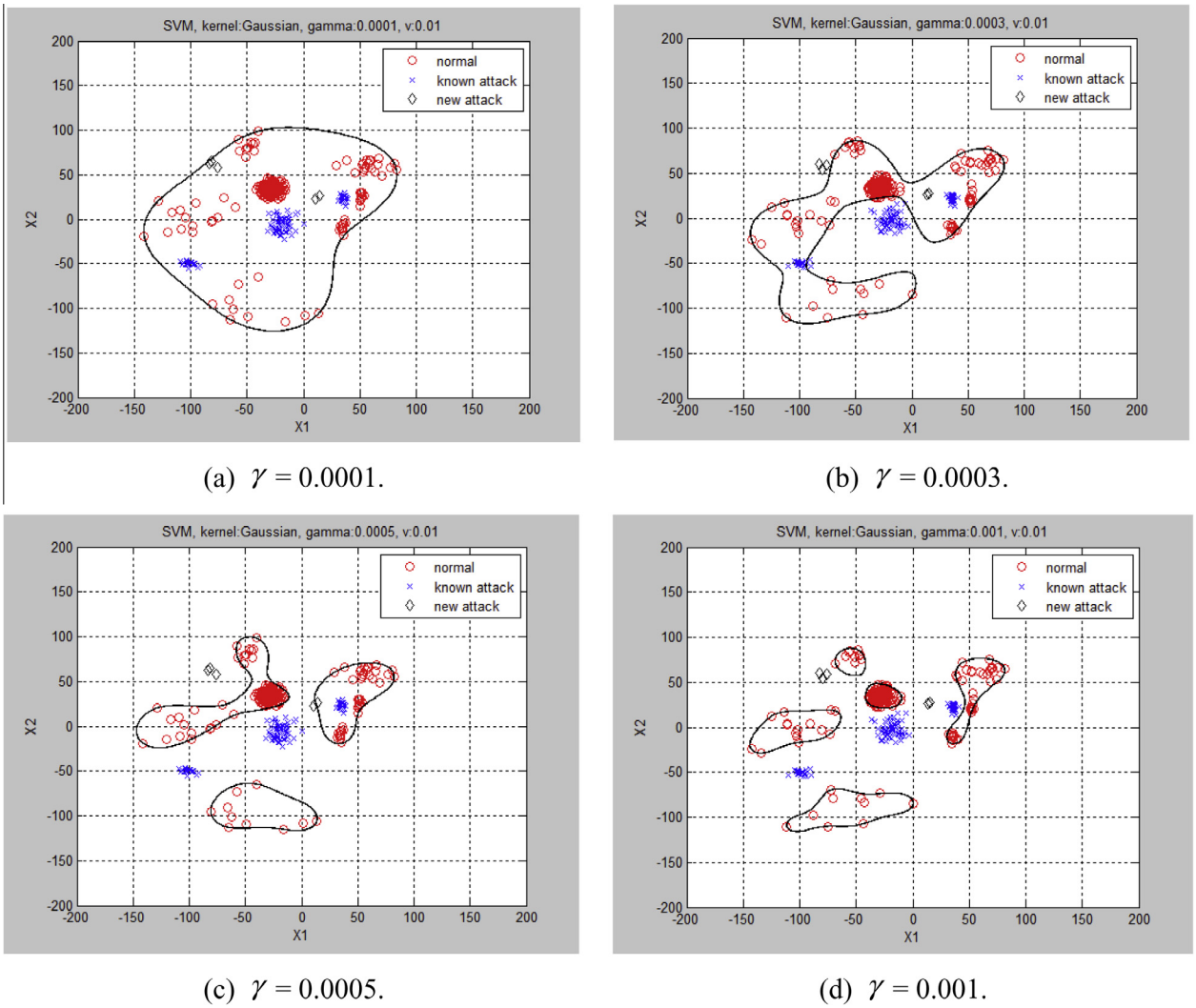


Fig. 1. Decision boundaries of the conventional 1-class SVM model.

KDDTrain+.TXT, it is possible to categorize the attack records in KDDTest+.TXT into known attacks and unknown attacks. However, even though the attack labels are the same, the traffic characteristics between KDDTrain+.TXT and KDDTest+.TXT are not sufficiently similar to satisfy the requirement for known attacks in the proposed context. Hence, in order to clearly distinguish between the known attacks and unknown attacks, the training data set and test data set were organized as follows.

The traffic data in KDDTest+.TXT was divided into two different sets based on whether the type of connection is known in KDDTrain+.TXT or not. The first set consists of connections that are known in KDDTrain+.TXT, and the second set consists of the unknown attack connections. Then, the KDDTrain+.TXT was combined with the first set and the combined data set was evenly divided based on the type of connection into the training data set and testing data set. Now, the testing set consists of normal and known attack connections. A second set of KDDTest+.TXT was added to the testing set and then the testing set organization was completed.

First, the detection performance of the proposed method was evaluated. The detection performance was compared with the misuse detection method (decision tree), anomaly detection method (1-class SVM), and their conventional hybrid approach. The minimum number of instances per leaf was set to approximately 0.1% and subtrees were allowed to raise the confidence factor to 0.1 in order to avoid the DT from over-fitting to the training data set.

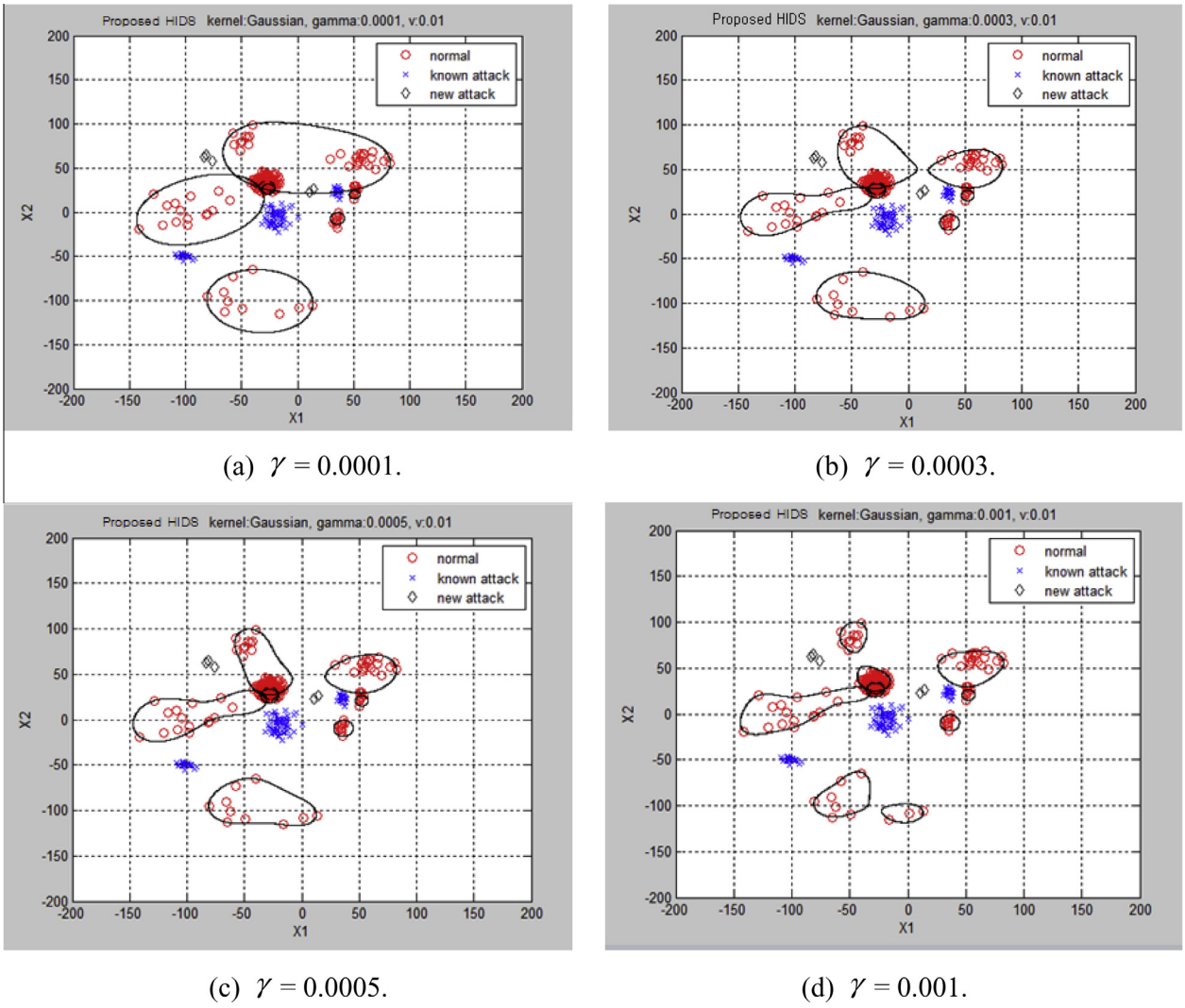


Fig. 2. Decision boundaries of the anomaly detection model in the proposed method.

Table 2
Testing process of the proposed hybrid intrusion detection method.

Testing process of the proposed hybrid intrusion detection method	
Step 1	Sense an incoming connection and extract the traffic features from the connection
Step 2	Check the connection with a trained decision tree using the extracted traffic features and determine whether the connection is a known attack
Step 3	If the decision tree classifies the connection as a known attack, then block the connection and go to Step 6; else, check the leaf of the decision tree where the connection belongs to and go to Step 4
Step 4	Check the connection with a trained 1-class SVM for the corresponding leaf, using the extracted traffic features, in order to verify if the connection is an unknown attack
Step 5	If the 1-class SVM detects the connection as an outlier, then block the connection and send an alarm to the security officer; else, allow the connection to the protected network
Step 6	Wait for the next incoming connection

For 1-class SVM, a Gaussian kernel was used; the parameter v was varied from 0.01 to 0.5; and the parameter γ was varied from 0.01 to 1.

The detection rate of the DT is 99.1% for known attacks and 30.5% for unknown attacks when the false positive rate is 1.2%. The detection performance for known attacks is desirable but, as is well known, the DT is not suitable for detecting unknown attacks. Because the objective of the proposed method is to improve the detection performance of the anomaly detection model, the detection performance of the proposed method was investigated in terms of the anomaly detection in detail. Fig. 4 presents the receiver operating characteristic (ROC) curves of the unknown attacks for the proposed method and its comparisons, with variations in parameter γ from 0.01 to 1. The detection rate of the proposed method is higher than those of the conventional methods considering the false positive rate. As each 1-class SVM model can focus on its corresponding decomposed region, the decision boundaries of the anomaly detection model in the proposed

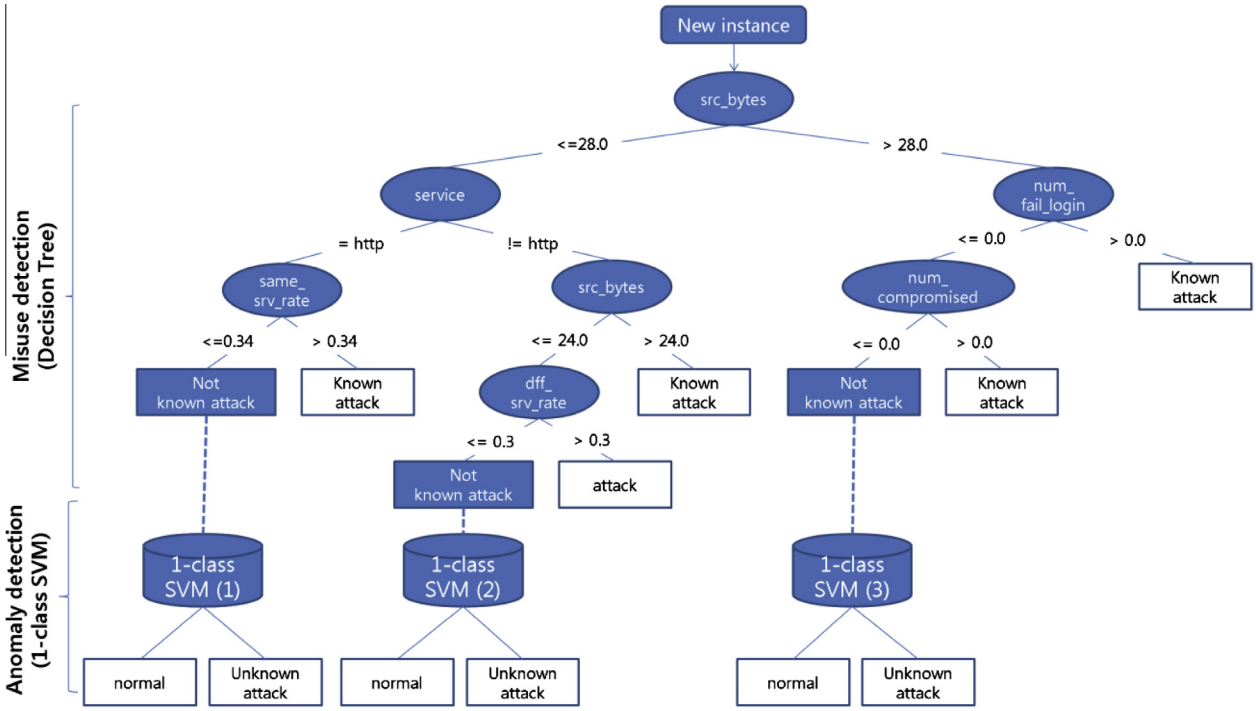


Fig. 3. Diagram of the decision making process of the proposed method. (a)ROC curve for unknown attacks when parameter $\gamma = 0.01$. (b) ROC curve for unknown attacks when parameter $\gamma = 0.1$.

method can describe the normal behavior better than those in the conventional methods. When the false positive rate is below 10%, which is desirable, the detection rate of the proposed method is approximately 10% higher than those of the conventional methods. It was observed that the detection rates of the conventional method are better when the false positive rate is approximately 50%. It is thought that this results from the proposed method building a 1-class SVM model for each decomposed region. When the false positive is very large (e.g. 50%), it would better to focus on highly concentrated regions rather than to focus on each decomposed region. Because an IDS operator must maintain a very low false positive rate, it can be concluded that the detection performance of the proposed method is superior to that of the conventional methods for unknown attacks.

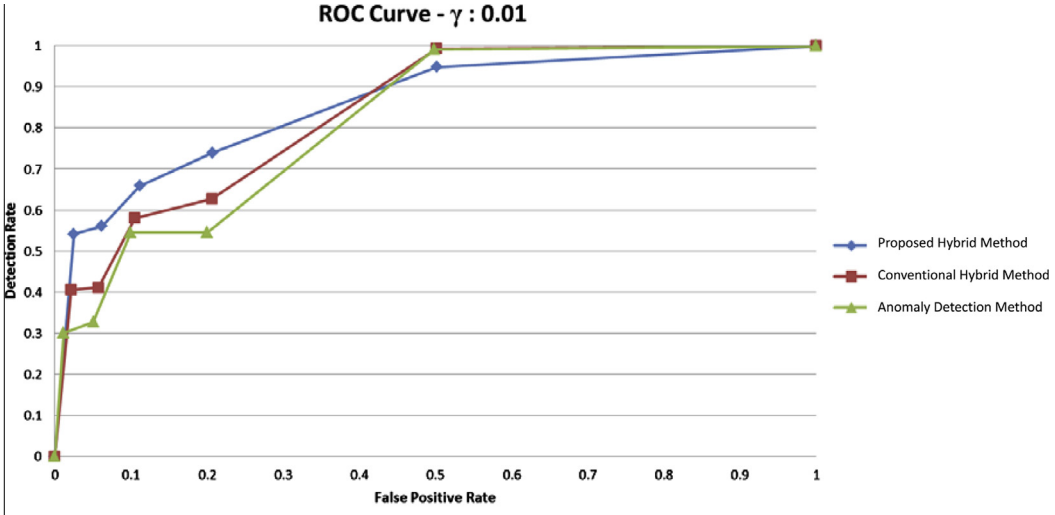
As the parameter γ increases, the detection rate of the 1-class SVM model converges to that of the conventional hybrid method. This indicates that the DT cannot affect the detection of unknown attacks as the performance of the 1-class SVM increases. However, it significantly affects the performance of the hybrid intrusion detection models that detect known attacks. Fig. 5 shows the ROC curves of known attacks for the proposed method and its comparisons varying the parameter γ from 0.01 to 1. As seen in the figure, the detection rates of both the hybrid detection methods are more than 99% without considering the performance of the 1-class SVM model. The detection rate of the hybrid intrusion detection model when γ is set to 1 is slightly higher than that of the DT, but its effectiveness is insignificant. As γ increases, the detection rate of the 1-class SVM model increases as in the unknown attack case. However, in this case, the misuse detection method using the DT is superior to the anomaly detection method using the 1-class SVM model.

Next, the training and testing times of the proposed method and its comparisons were measured. The training time was measured using Weka 3.6 and the testing time was measured using MATLAB. Table 3 presents the results.

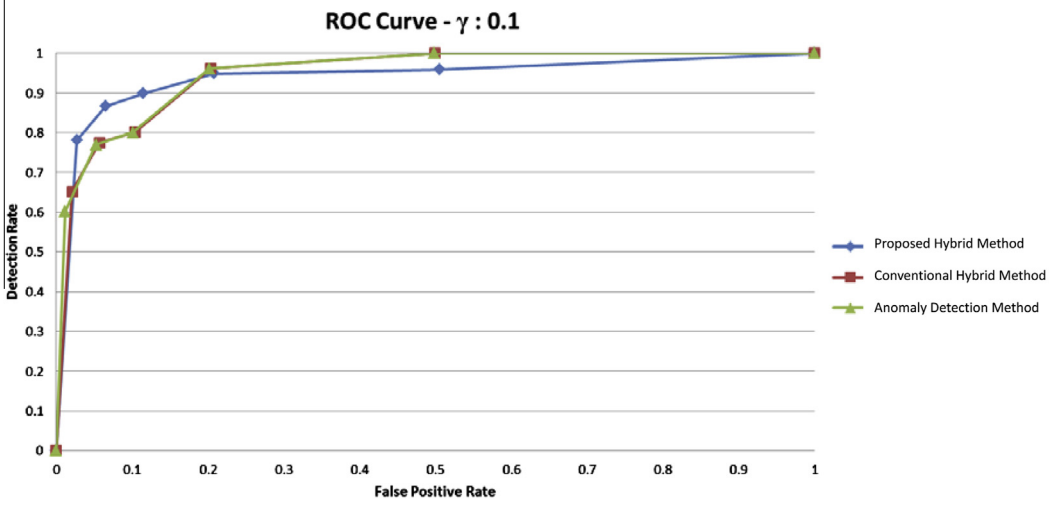
The training time of the proposed method is 56.58 s. This is shorter than that of the conventional methods, which are 76.63 s. The conventional methods combine a misuse detection model and an anomaly detection model independently. Thus, the time required to train the hybrid methods is calculated by adding the time required to train the misuse detection model (35.21 s) and time required to train the anomaly detection model (41.42 s). However, because the training time of the anomaly detection model in the proposed method can be improved through the tree decomposition, the training time can be shortened by approximately 26.2%. The testing time of the proposed method is 11.20s, which is also shorter than that of the conventional hybrid methods, as expected. The testing time can be reduced by 28.2% compared with the conventional methods. It should be noted that the testing process of the DT is very fast and thus the testing time of the hybrid intrusion detection depends on the testing time of the 1-class SVM as mentioned previously.

The details of the detection time are reported in Table 4. In this experiment, the normal data in the training data was divided into 12 subsets. In Table 4, the training time of the anomaly detection model of the proposed method is 21.37 s and that of both conventional methods is 41.42 s. As a result, the training time of the proposed method was reduced by 48% compared with the conventional methods.

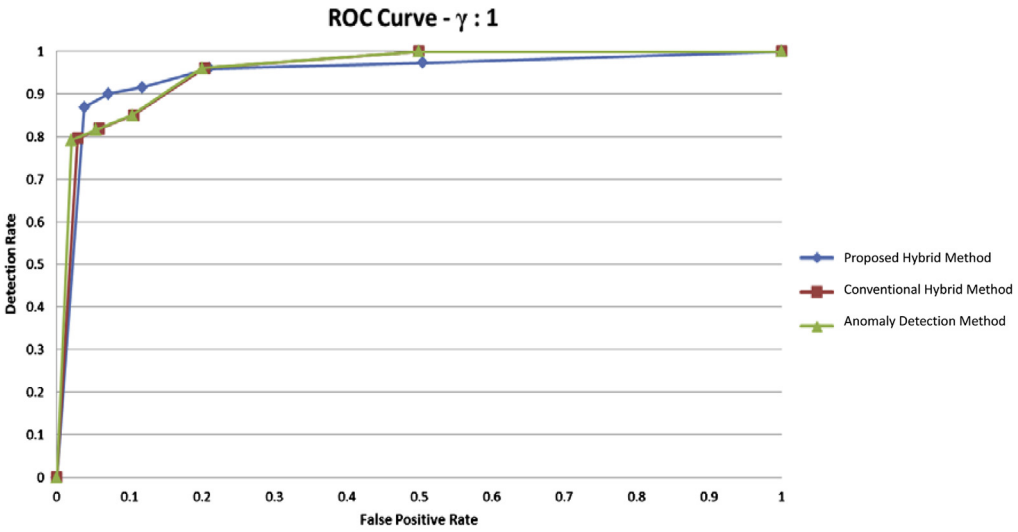
The testing time of the anomaly detection model in the proposed method is 10.13 s and that in the conventional method (Serial) is 14.55 s. In order to understand the reason for this difference, the average number of encountered support vectors (avg_SVs) were calculated by dividing the total number of encountered SVs by the total number of testing instances. In this experiment, the avg_SVs (403.27) is less than the number of SVs in the conventional methods (664). Because the number of SVs affecting the complexity of the testing computation can be reduced, it is possible to reduce the testing time. The conventional method (Parallel) requires 30.17 s. The Parallel method requires more time than the



(a) ROC curve for unknown attacks when parameter $\gamma = 0.01$.

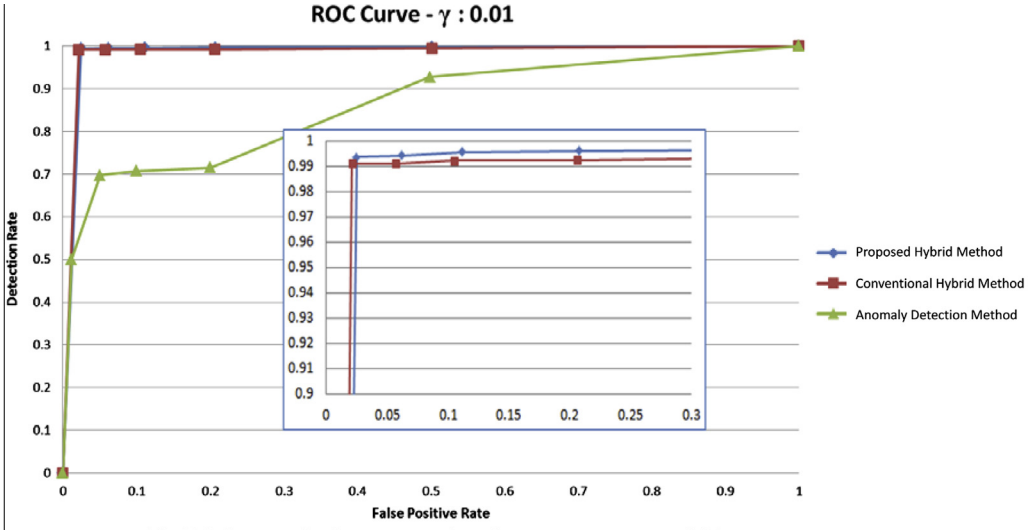


(b) ROC curve for unknown attacks when parameter $\gamma = 0.1$.

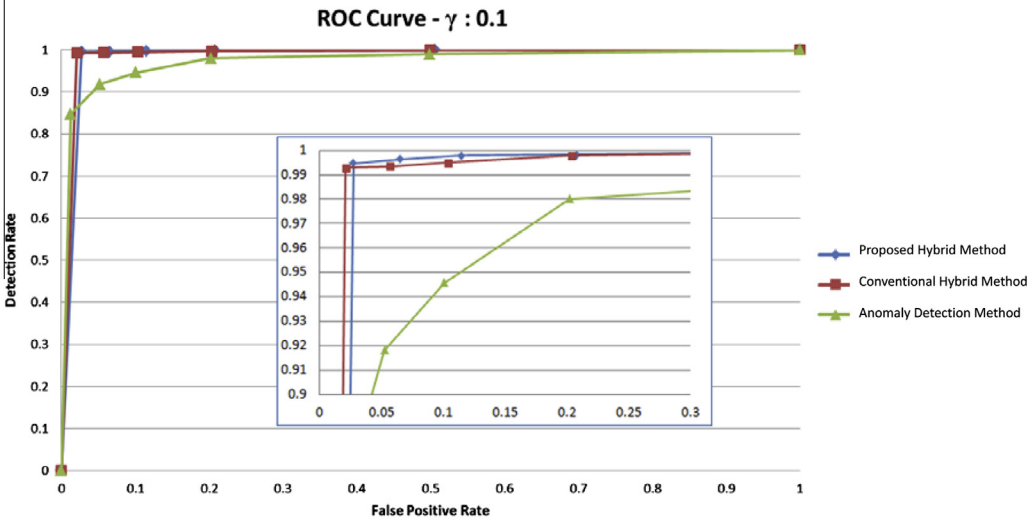


(c) ROC curve for unknown attacks when parameter $\gamma = 1$.

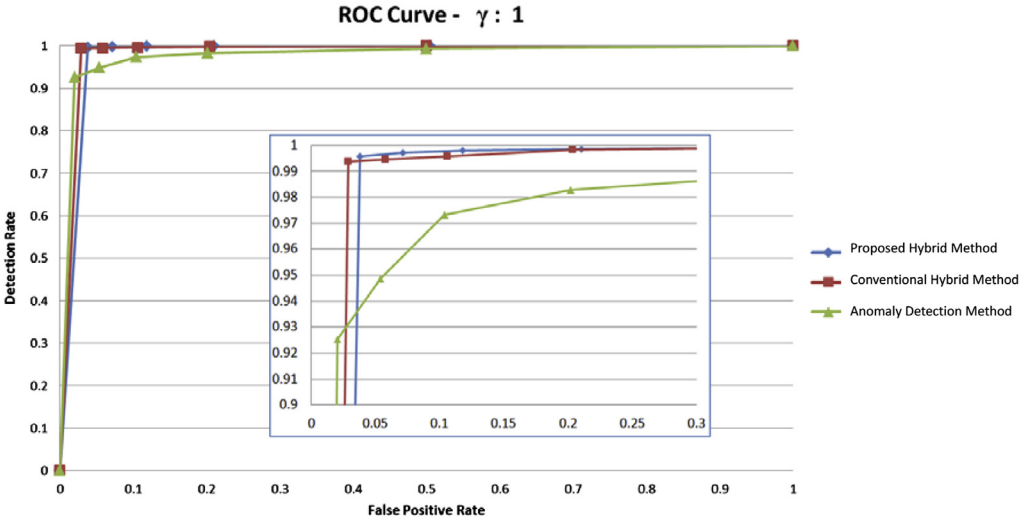
Fig. 4. Detection performance comparison using ROC curves for unknown attacks. (a) ROC curve for known attacks when parameter $\gamma = 0.01$. (b) ROC curve for known attacks when parameter $\gamma = 0.1$. (c) ROC curve for known attacks when parameter $\gamma = 1$.



(a) ROC curve for known attacks when parameter $\gamma = 0.01$.



(b) ROC curve for known attacks when parameter $\gamma = 0.1$.



(c) ROC curve for known attacks when parameter $\gamma = 1$.

Fig. 5. Detection performance comparison using ROC curves for known attacks.

Table 3
Training time and testing time comparison ($\nu = 0.01$; $\gamma = 1$).

Methods	Training time (s)	Testing time (s)
Proposed hybrid method	**56.58**	**11.20**
Conventional hybrid method (Serial)	76.63	15.62
Conventional hybrid method (Parallel)	76.63	30.17
Misuse detection method (DT)	35.21	1.07
Anomaly detection method (1-class SVM)	41.42	29.10

Table 4
Detection time comparisons between the proposed method and conventional methods.

Decomposed subset	# of training data	Training time (s)	# of testing data	Testing time (s)	# of support vectors
1	1491	0.43	1779	0.11	87
2	253	0.16	244	0.02	41
3	483	0.22	1485	0.11	88
4	112	0.06	95	0.01	10
5	91	0.07	93	0.01	20
6	833	0.28	1342	0.05	25
7	33,511	19.36	34,527	9.73	467
8	67	0.07	109	0.01	18
9	496	0.22	482	0.03	70
10	128	0.10	162	0.01	46
11	452	0.21	470	0.02	30
12	172	0.19	186	0.02	55
Proposed method	38,089	21.37	40,974	10.13	403.27
Conventional method (serial)	38,557	41.42	40,974	14.55	664
Conventional method (parallel)	38,557	41.42	76,133	29.10	664

other hybrid methods because its anomaly detection model verifies all incoming connections while the other hybrid methods do not verify connections that are detected as known attacks by the misuse detection model.

This experiment demonstrates that the proposed hybrid intrusion detection method is better than the conventional methods in terms of detection performance, training time, and testing time. It should be noted that the proposed method does not require an additional overhead in order to integrate the detection models. It is suggested that the training data set is decomposed before performing the anomaly detection. This decomposition is appropriate when using multi-core/parallel/distributed computing for further increases in speed. In future research, a specific decision tree algorithm that is more appropriate to the proposed hybrid intrusion detection method will be developed. Because the original C4.5 decision tree does not consider clusters in the normal data set, it can divide a well-formed normal cluster during the decomposition processes. This can hinder the well-formed normal cluster that belongs to a single decision boundary in the 1-class SVM model, which can degrade the profiling ability. In addition, the uneven distribution of data instances impedes the reduction of the training time and testing time. Note that approximately 87% of the training data belonged to subset 7 in Table 4. Therefore, the time reducing ability of the proposed method is not as good as expected. Hence, future research will focus on modifying the C4.5 decision tree algorithm to improve the supplement points as mentioned above without losing its ability to detect known attacks.

5. Conclusion

In this research, a new hybrid intrusion detection method that hierarchically integrates a misuse detection model and an anomaly detection model in a decomposition structure was proposed. First, the C4.5 decision tree (DT) was used to create the misuse detection model that is used to decompose the normal training data into smaller subsets. Then, the one-class support vector machine (1-class SVM) was used to create an anomaly detection model in each decomposed region. Throughout the integration, the anomaly detection model can indirectly use the known attack information to enhance its ability when building profiles of normal behavior. The experiments demonstrated that the proposed hybrid intrusion detection method could improve the IDS in terms of detection performance for unknown attacks and detection speed. This is the first attempt to use the misuse detection model to enhance the ability of anomaly detection model. This attempt to improve the anomaly detection model with known attack information in the hybrid intrusion detection method merits further research and investigation. Hence, future work on this research issue will focus on improving the C4.5 algorithm in order to decompose the normal data evenly into each subset without degrading the misuse detection performance, which will be expected to significantly improve the performance.

Acknowledgements

This research was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

References

Anderson, D., Frivold, T., & Valdes, A. (1995). *Next-generation intrusion detection expert system (NIDES), software users manual, beta-update release*. SRI International, Computer Science Laboratory.

Barbara, D., Couto, J., Jajodia, S., Popyack, L., & Wu, N. (2001). ADAM: Detecting intrusions by data mining. In Proceedings of the IEEE workshop on information assurance and security (pp. 11–16).

Chang, F., Guo, C., Lin, X., & Lu, C. (2010). Tree decomposition for large-scale SVM problems. *Journal of Machine Learning Research, 10*, 2935–2972.

Chang, C., & Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*(3), 27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

Depren, O., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system for anomaly and misuse detection in computer networks. *Expert Systems with Applications, 29*(4), 713–722.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter, 11*(1), 10–18.

Hwang, K., Chen, Y., & Qin, M. (2007). Hybrid intrusion detection with weighted signature generation over anomalous Internet episodes. *IEEE Transactions on Dependable and Secure Computing, 4*(1), 41–55.

Manevitz, L. M., & Yousef, M. (2002). One-class SVMs for document classification. *Journal of Machine Learning Research, 2*, 139–154.

Perdisci, R., Gu, G., & Lee, W. (2006). Using an ensemble of one-class SVM classifiers to harden payload-based anomaly detection systems. In Proceedings of the 6th international conference on data mining (pp. 488–498).

Quinlan, J. (1996). Learning decision tree classifier. *ACM Computing Surveys (CSUR), 28*(1), 71–72.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*(7), 1443–1471.

Shin, H. J., Eom, D. H., & Kim, S. S. (2005). One-class support vector machines: An application in machine fault detection and classification. *Computers & Industrial Engineering, 48*(2), 395–408.

Song, J., Takakura, H., Okabe, Y., & Kwon, Y. (2009). Unsupervised anomaly detection based on clustering and multiple one-class SVM. *IEICE Transactions on Communications, E92-B*(6), 1982–1990.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceeding of the IEEE symposium on computational intelligence for security and defense applications (pp. 53–58).

Zhang, J., & Zulkernine, M. (2006). A hybrid network intrusion detection technique using random forests. In Proceedings of the first international conference on availability, reliability and security (pp. 262–269).