Journal of Computer Science

Original Research Paper

Soft Marking Scheme of SVM Hierarchical Classifiers for Attack Classification

[1]Azizi Abdullah and [2]Warhamni Jani@Mokhtar

[1]Center for Artificial Intelligent, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia
[2]Department of Malaysia, Information Technology Management Division, Accountant General's, Putrajaya, Malaysia

Article history
Received: 29-07-2021
Revised: 30-07-2021
Accepted: 27-08-2021

Corresponding Author:
Azizi Abdullah
Center for Artificial Intelligent,
Faculty of Information Science
and Technology, Universiti
Kebangsaan Malaysia,
Malaysia
Email: azizia@ukm.edu.my

Abstract: Classification is a predictive modelling problem that involves assigning a class label to an instance correctly. There exist several strategies in machine learning to deal with the multi-class classification problems for attack detection. One of the popular strategies is the one-vs-one that decomposes the multi-class problem into multiple binary ones. The approach has been applied in many popular supervised learning algorithms, such as support vector machines. A possible problem of the standard multi-class classification problem is that it lacks correlation between different classes, which can increase overfitting problems and hinder generalization performance. Thus, a possible solution to the problem is to use a hierarchical classification that captures the relationship between classes by dividing the multi-class classification problem into a tree. However, one possible challenge in this approach is selecting parent and child nodes of the tree. The selected nodes should be informative to recognize and then classify different attack classes. One way is by looking at specific domain knowledge to train and build classifiers of the base learners for effective prediction. Thus, a soft marking scheme is introduced to assess a set of binary classifiers to ensure the best overall predictive base learners. Finally, we validate and compare the proposed approach to the standard NSL-KDD dataset. The results show that the proposed method outperforms the standard classifier on the intrusion attack classification.

Keywords: Support Vector Machine (SVM), Hierarchical Classifier, Attack Detection, Multi-Class SVM

Introduction

The growing number of security threats has prompted researchers to use various classifiers, especially in Intrusion Detection Systems (IDSs), such as support vector machines. In the IDS, signature-based and anomaly-based are the two major detection techniques (Nguyen *et al*., 2012; Cui *et al*., 2016; Lee and Stolfo, 2000). Signature-based approach detects attacks by analysing network traffic and comparing it to attack patterns stored in a database. It functions similarly to anti-virus software in that it detects suspicious activity by matching well-known patterns from databases. Although this technique may consistently and swiftly identify assaults, it has several limitations in detecting new or unknown malicious behaviors. On the other hand, anomaly-based bases on estimation and prediction techniques that employ a set of priori profile assaults or information gathered from network sensors. These profiles are used to train a machine-learning algorithm to learn what types of attacks need to be detected and classified. In this case, this approach has some advantages in identifying unknown attacks. But due to some complexities and limitations on learning algorithms, it is challenging to construct a complete model that can detect all possible attacks. Thus, the approach is one of the most widely researched area in attack recognition until now (Panetta, 2017).

In literature, the most commonly researched in attack recognition is the anomaly-based technique. The technique can be categorized into three main approaches, namely statistical based, knowledge based and machine learning based (Garcia-Teodoro *et al*., 2009). Among all these techniques, machine learning considers the most popular and reliable approach for attack recognition. It contains a set of instructions is used to create effective classification classifiers. The machine learning algorithms can be grouped into four main categories, namely (a) supervised learning (b) semi-supervised learning (c) unsupervised and (d) reinforcement learning. However, this study focuses on the supervised machine learning implementation for attack prediction. In this regard, a number of other overviews on the supervised learning intrusion detection have been published to date, see e.g., (Laskov *et al*., 2005; Gharibian and Ghorbani, 2007; Tavallaee *et al*., 2009; Belavagi and Muniyal, 2016). However, the purpose of this study is to improve the predictability of intrusion attacks in terms of the multi-class classification problem.

The most popular strategies to solve a multi-class problem into multiple binary ones are The One-Versus-All (OVA), the One-Vs-One (OVO) (Bishop, 2006). These approaches are mainly utilized in many popular supervised learning algorithms such as Support Vector Machines (SVMs) (Cortes and Vapnik, 2004), Neural Networks (NNs) (Hopfield, 1982) and logistic regression (Tolles and Meurer, 2016). Even though these multi-class classification problems are widely employed, but sometimes they failed to classify test attacks instances correctly (Ergen and Kozat, 2019). Furthermore, how one should deal with a multi-class recognition problem is still an open issue (Duin and Pekalska, 2005). One possible alternative approach is to use a hierarchical classification that divides the multi-class classification problem into a tree (Cesa-Bianchi *et al*., 2006; Ahmim and Zine, 2015; Ahmim *et al*., 2019). Each parent node is divided into two child nodes in this scheme and the process is continued until each child node represents only one class. However, one possible challenge in this approach is selecting parent (root) and child nodes of the tree. Furthermore, the selected nodes in the tree-based structure should be discriminative by considering the domain knowledge and the relationship between different classes of the base learners (Albashish *et al*., 2018; Zhang *et al*., 2018). Thus, in this study, we introduce a soft marking scheme to assess a set of binary classifiers to ensure the best overall predictive base learners to model the relationships of different attacks. This study starts with selecting the best features to describe the intrusion attacks from the NSL-KDD dataset (Tavallaee *et al*., 2009). After that, the 5-cross validation is used to construct classifier models and a ranking rule is used to select the sequence of the most discriminative-based learner classifiers. Then, we validate and compare the proposed approach on the standard NSL-KDD dataset.

The contribution of this study. Abdullah *et al*. (2009) we proposed a soft marking hierarchical multi-class scheme for attack classification. The method assigns a weight for each based classifier which represents a leaf in the decision tree. The standard SVM that trains on the standard dataset is not the best to describe attack features, but an efficient ranking of the based learner using hierarchy or decision tree structure of them can be. Ahmim *et al*. (2019) we demonstrate the effectiveness of a soft marking scheme with a hierarchical approach for intrusion detection. Ahmim and Zine (2015) we compare the most widely used standard SVM and the proposed soft marking Scheme Hierarchical SVM classifiers (SHSVM) for intrusion detection on the NSL-KDD dataset.

Related Work

In this section, we discuss some related works that are used to construct the hierarchical SVM learning for intrusion attack classification.

Multi-Class Classification

There are two types of classification problems in machine learning, namely binary classification and multi-class classification. The classifier model is produced from training with only two classes in a dataset in the binary classification case. The classifier is then tested on a given test sample to + 1 if it contains some properties that belong to the model. And it classifies -1 if the example doesn't belong to the model. In contrast, multi-class classification trains multiple classes in a dataset. In general, multi-class classification problems can be trained using the binary classification method. It can be done differently by the One-Versus-One (OVO) or One-Versus-All (OVA) approach. After that, the goal of classifiers is to determine which of the N classes the test sample belongs to.

In OVO, the classifier uses a max-win voting scheme to classify which test sample belongs. In this approach, there are $N \times (N-1)/2$ class models that need to be constructed, where N is the number of classes. Each one distinguishes only between samples of 2 classes. As a result, a model is trained with +1 for positive class samples and -1 for negative class samples that do not belong to the class. To determine which a test sample belongs to, all $N \times (N-1)/2$ models are tested and the class which most often wins against the other classes is considered as the winner.

In OVA it uses a winner-takes-all strategy. Thus, in this scheme, there are N class models constructed and one for each class. Then, each model receives training data +1 for samples belonging to that class and -1 for all examples belonging to one of the other classes. After that, all N models are trained and for testing, the test samples are given to all N class models and the model with the highest probability output is assumed to be the right model.

Support Vector Machines

The Support Vector Machine (SVM) is a supervised learning algorithm developed by Vapnik and others at AT and T Bell Laboratories (Cortes and Vapnik, 2004). SVM is one of popular machine learning algorithms for classification and has been extensively used with excellent empirical performance in computer security (Ariff *et al*., 2018; Zolfi *et al*., 2019; Kadis and Abdullah, 2017) and other research areas such as in computer vision (Albashish *et al*., 2018; Abdullah *et al*., 2009; Nashat *et al*., 2011). The method is intended for binary classification problems. It might, however, be extended to include multi-class classification.

The main goal of SVM is to locate the optimal separating hyperplane as the decision line which separates the +1 class from the -1 class by maximizing the most significant or largest margin between the classes' closest points. The hyperplane is calculated by determining the boundaries of the input data. The points lying on boundaries are called support vectors and the middle of the margin is the optimal separating hyperplane. Figure 1 shows the main components of the SVM algorithm.

In this study, the classification of intrusion attacks is divided into five different categories according to the NSL-KDD dataset. In order to extend it to multiclass classification is by considering the problem as a collection of binary classification problems. Suppose that given some input of attacks l in the training data corresponding to the two groups and each input is denoted by a sample $\vec{x}_i, (i = 1, ..., 1)$, which represents the selected attack features. Therefore, the training data can be represented by a set of sample-label pairs $(\vec{x}_i, y_i)$, where $x_i \in R^n$ in some n-dimensional space and $y \in +1, -1$ indicates the class label. The classification of attack samples can be considered as the task of determining a classification f function from the training data. After that, the constructed function model can be used to predict which the test sample belongs. If $f(\vec{x}_i) > 1$, the test sample is assigned to the class $y_i = +1$, otherwise $y_i = -1$. For the linear problem, the classification function f has the following form:

$$f(x) = sign\left(\vec{W}^T \vec{x} + b\right) \tag{1}$$

where, $\vec{W}$ is the slope type to the hyperplane and b is a bias term which satisfy the following condition:

$$y_i\left(\vec{W}^T \vec{x} + b\right) \geq 1 \tag{2}$$

Thus, given the training set in a vector format, the SVM attempts to find the optimal hyperplane that maximizes the margin between +1 and -1 samples. In the linear problem,

SVM attempts to maximize the margin is by minimizing $\frac{1}{2}\vec{x}\vec{x}$ that subject to constrain in Eq. (2). However, for the non-linear problem, the input vectors $\vec{x}_i$ are mapped into a high dimensional feature space by using a basic kernel function $K\left(\vec{x}_i, \vec{x}_j\right)$. In this study, the most popular kernel namely Radial Basis Function (RBF) is used for mapping as in Eq. (3):

$$K\left(\vec{x}_i, \vec{x}_j\right) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \gamma > 0 \tag{3}$$

where, γ is the RBF kernel parameter value that affects the partitioning outcome in the feature space. In terms of kernels, the following function form is used for classification as in Eq. (4):

$$f(x) = sign\left(\sum_{i=1}^{l} \alpha_i y_i K\left(\vec{x}_i, \vec{x}_j\right) + b\right) \tag{4}$$

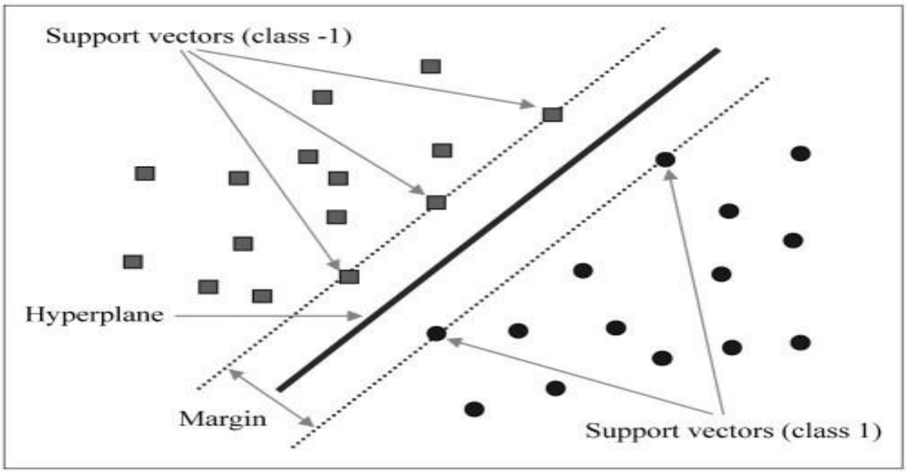

Fig. 1: Support vector machine and its important components

where, K is a kernel function, b is a bias term and α_i is the Lagrange multiplier coefficient with corresponding coefficients $\alpha_i > 0$. The coefficients α_i is obtained by maximizing the function as follows:

$$W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{i=1}^{l}y_i y_j K\left(\vec{x}_i,\vec{x}_j\right)\alpha_i\alpha_j \tag{5}$$

with subject to:

$$0\le\alpha_i\le C,\left(i=1,...l\right) and \sum_{i=1}^{l}y_i\alpha_i=0 \tag{6}$$

where, C is a non-negative regularization parameter and represents the cost of the penalty.

Grid Search

A Support Vector Machine (SVM) is a supervised machine learning algorithm that learns from training examples to classify the given set of attack records. SVM uses kernel functions to map the input data to higher dimensional feature space such as RBF, polynomial and Sigmoid (Hsu *et al*., 2008). However, the RBF kernel basis function is used in this study and it is susceptible to its learning parameters or hyper-parameters, namely C and γ values. Thus, the best hyperparameter values for the kernel need to be obtained. In this case, the grid-search algorithm (Hsu *et al*., 2008) is employed, which is the most basic hyperparameter optimization implementation. In this scheme, the user specifies a finite set of values for each hyperparameter and grid search evaluates the Cartesian product of these sets. We tried the set values {2−5, 2−3,...,215}and {2−15, 2−13,...,23} for *C* and γ, respectively. The parameter values which gave the best accuracy performance with n-fold cross-validation are picked and used to train on the training dataset.

Feature Normalization

Feature normalization is important in SVMs and it can be used to minimize domination of certain smaller number values in feature space (Hsu *et al*., 2008). One often gets better classification performance with normalized feature vectors by improving the numerical condition based on the statistical training data distribution. Besides, the normalization can avoid some numerical issues associated with inner product calculation in feature space. In this study, all feature vectors (x) in the dataset are normalized to the interval [-1, +1] by using the following feature normalization function as in Eq. (7):

$$x'=\frac{2\left(x-\min\right)}{\max-\min}-1 \tag{7}$$

where min and max values are determined in the training dataset.

Cross Validation

Cross-validation is a popular statistical method for choosing the best classifier model from a pool of candidates for reliability machine learning models. It can give a comprehensive measure of the model's performance throughout the whole dataset. In this method, we split the dataset into k number of subsets (known as folds), then we perform training on all the subsets but leave one subset for the evaluation of the trained model. In this method, we iterate k times with a different subgroup reserved for testing purposes each time. Then, the average accuracy performance of k times evaluations is used to select the best model. After that, the approach uses an SVM classifier on different C and γ values for evaluation. In this case, the 5-fold cross-validation on the train set to tune the learning parameters of SVM. Finally, we pick a classifier model that gives the best average classification performance on the dataset.

Soft Marking Based Hierarchical SVM Classifiers (SHSVM)

This section discusses the hierarchical SVM classifiers (SHSVM) for classifying attacks in the NSL-KDD dataset. The learning classifier is inspired by a Binary Tree Support Vector Machine (BTSVM) learning architecture for solving the multi-class classification problem (Wang *et al*., 2007). The main idea of BTSVM is to solve an N-class problem by decomposing it into N-1 sub-problems, each separated with SVM classifiers. SHSVM uses normal traffic and four different intrusion classes, namely, dos, probe, u2r and r2l, for attack detection. Then, these five class problems are decomposed into a series of binary classification sub-problems based on the classifier accuracy performance from the cross-validation results.

Algorithm 1 SHSVM

```
Input: The training dataset
Step1: Identify the best features for attack
    descriptions from the dataset
Step 2: Identify the best five binary classifications
    models from a set of One-vs-One models
Step 3: Identify the best five binary classifications
    from a set of One-vs-All models
Step 4: Identify the best five One-Vs-One multi-class
    classifications
Step 5: Compute points or scores for each attack
    category using the soft marking scheme approach
Output: Ranked classifiers for SHSVM
```

Hierarchical Classifier Algorithm

The SHSVM decomposes the binary class problem into a set of nodes. Each node represents a classifier model using the training examples from two classes chosen out of N classes. Two critical issues in SHSVM learning are (a) identifying the most discriminative class for classification tasks and (b) decomposing sub-problems into smaller ones. Thus, one possible solution is to use the statistical k-fold cross-validation method. Some experiments have been proposed in this technique to identify the best discriminative class for SHSVM. We introduced four different experiments and each experiment employs the standard multi-class classification problems, i.e., OVA and OVO. The argument is that the output of a single step is not the best to choose the best model, but the combination of a series of steps can be. Algorithm 1 depicts the complete SHSVM process. Therefore, in this study, we develop a hierarchical classifier using various binary classification strategies to identify the best base learners and can be explained as follows:

Step 1: The main goal is to identify the best features for attack description. The distinctive features are essential for constructing reliable classifier models. We have conducted several experiments to ensure the selected features are capable of providing better intrusion attack descriptions (Kang and Kim, 2016). We used some earlier findings in this case, such as in (Staudemeyer and Omlin, 2014), which suggested using 13 and 15 features for host and network attacks, respectively. Besides, we use all 41 features provided in the dataset. After that, these three types of feature categories will then go through some experiments using OVO multi-class classification by employing SVM with RBF kernel and 5-cross validation accuracy to select the best feature category. In this step, we found that 41 features gave the best performance for attack description. Thus, these 41 features are used in the next steps of the algorithm

Step 2: The main goal is to identify the best five binary classification models from a set of OVO models. In this step, the dataset is decomposed into five main classes and a number of instances (in bracket), namely Normal (5000 samples), DoS (5000 samples), Probe (5000 samples), U2R (52 samples) and R2L (995 samples). After that, each class will be paired with other classes resulting in 20 different pairs using OVO. Next, the SVM models on training samples with RBF kernel are proposed. We used the grid-search algorithm to find the best C and γ values for each model. The 5-cross validation accuracy is used to validate the training process of SVM by testing a different number of pattern subsets (or folds). The accuracy classification results from the cross-validation are used to identify the best model. We found that the DoS vs. Probe and Probe vs. R2L gave the best accuracy performance of 100% each. The worse model is U2R vs. R2L, with an accuracy of 97.42%. Table 1 shows the overall results in step 2

Step 3: The main goal is to identify the best five binary classification models from a set of One-vs-One models. The dataset is separated into five different classes in this experiment and OVA is used to train the multi-class classifier of SVM to generate a prediction model. The SVM with RBF kernel is used to build five different models in this method. We used the grid-search algorithm to find the best C and γ values to construct SVM models. Each SVM model is trained with all samples in the class as positive labels and the other samples as negative labels. For example, to train DoS, all samples from DoS are labelled as a positive label and other samples (Normal, U2R, R2L and Probe) as a negative label. The 5-cross validation accuracy is used to validate the training process of SVM by testing a different number of pattern subsets (or folds). The results showed that DoS gives the best accuracy performance of 99.99%. Probe, R2L and U2R are the second, third and fourth and last, respectively, on the list. Table 2 shows the overall results in this step

Step 4: The main goal is to identify the best five OVO multi-class classifications. In this step, the dataset is divided into five different classes. OVO approach is used to train the multi-class classifier of SVM with RBF kernel for obtaining a prediction model. OVO constructs all possible pairwise classifiers in this approach, where each classifier is constructed using the training samples from two classes chosen out of five attack categories. We used the grid-search algorithm to find the best C and γ values for SVM models. The 5-cross validation accuracy is used to validate the training process of SVM by testing a different number of pattern subsets (or folds). For testing the models, all samples are classified by all classifiers. Each classifier gives one vote for the class that in-favours the test sample and the process continues for the other 19 classifiers. When finish testing for all models, the class with majority votes is used to assign the correct label. The results showed that DoS gives the best accuracy performance of 100%. Followed by Probe for the second and U2R gives the worse detection result. Table 3 shows the overall results in step 4

Step 5: To rank each classifier model using the soft marking scheme as discussed in the next section

Table 1: Binary classification strategy with 5-cross validation results for all single models

Intrusion attack	Accuracy (%)	Rank
Normal vs. Dos	99.95	3
U2R vs. DoS	99.98	2
R2L vs. Probe	100	1
DoS vs. Probe	100	1
Probe vs. R2L	100	1

Table 2: OVA binary classification strategy with 5-cross validation results for all single model

Intrusion attack	Accuracy (%)	Rank
Normal vs. {U2R, R2L, DoS, Probe}	99.58	5
U2R vs. {Normal, R2L, DoS, Probe}	99.83	4
R2L vs. {Normal, U2R, DoS, Probe}	99.91	3
DoS vs. {Normal, R2L, U2R, Probe}	99.99	1
Probe vs. {Normal, R2L, U2R, DoS}	99.91	2

Table 3: OVO Multi-class Classification Strategy with 5-Cross Validation Results

Intrusion attack	Accuracy (%)	Rank
Normal	99.60	3
U2R	78.85	5
R2L	98.59	4
DoS	100	1
Probe	99.92	2

Table 4: The hierarchical order of the NSL-KDD dataset

Attack class	Total scores	Hierarchical rank
Normal	1.4	4
U2R	1.4	4
R2L	2.0	3
DoS	3.0	1
Probe	2.6	2

Soft Marking Scheme for Hierarchical Classifier

Based on the experimental results for each step, the next problem is to determine the rank order of the importance of selected classes used in the hierarchical SVM classifiers. Following this, a simple soft marking scheme is used to arrange the order according to the highest accuracy at first, followed by subsequent other model accuracies for the next level. In this method, each winner is given a separate weight or probability value. The value is based on the following equation in Eq. 8:

$$score(rank) = \begin{cases} 1.0 \; rank = 1 \\ \dfrac{rank}{M} \; otherwise \end{cases} \tag{8}$$

The rank order of SVM classifiers for attack detection is shown in Table 4 from strongest to weakest. The DoS class is ranked as number one in this table due to the highest scores in all these three steps, i.e., steps 2, 3 and 4. As a result, it can be chosen as the most potent class and used in the proposed hierarchical SVM classifiers' initial node (root node). The second-place robust classifier is Probe, followed by and R2L, U2R and Normal for the third, fourth and last nodes.

The first step in demonstrating the notion of hierarchical SVM classifiers is to rank each class into strong and weak models as shown in Table 4. The next step is to construct a set of binary classifiers that address the five-class classification problems in intrusion attack detection. In this case, the root node of SHSVM includes the most discriminative classifier, namely DoS vs. Others (Probe, R2L, U2R and Normal). The first class has a positive label, whereas the remaining classes have a negative label. The next level belongs to the less discriminative binary tasks, Probe vs. others (R2L, U2R and Normal). In this level, the left node, which is already used for classification (level one), will be removed from binary sub-problems. The process of eliminating, separating and merging continues until only one group of nodes remains. Figure 2 shows the structure of SHSVM for solving the five-class classification of the NSL-KDD dataset. The main difference between SHSVM and the existing OVA and OVO approaches is the connection between interclass relationships and correlation among different classes. It starts from the most potent class at first (root node) until the weakest class for classification. Furthermore, the number of samples used to train SVM models is decreased as SHSVM levels increase, reducing the imbalanced dataset for training and reducing the problem of classifier overfitting.

Results

A well-known dataset is required for a reliable experiment. Therefore, to demonstrate the performance of our SHSVM algorithm, the standard dataset, namely NSL-KDD, is used (Tavallaee *et al*., 2009). In this study, the machine that is used to run experiments is Intel ^{R} Core^{TM} i7-8550U CPU with 4GB memory and 500GB 5400 rpm SATA HDD for experiments. For evaluation, two different experiments are performed. The first experiment is to evaluate the SHSVM classifier using 5-cross validation on the training samples of size 16047 and the second experiment is on the test samples of size 22544 as mentioned in Table 5. In this study, the lib SVM library (Hsu *et al*., 2008) is used throughout all of the experiments.

NSL-KDD Dataset

The NSL-KDD is one of the popular and widely used datasets to demonstrate the performance of intrusion detection analysis (Tavallaee *et al*., 2009). The NSL-KDD is an improved version of the KDD Cup 99 dataset (Cup, 2007), which contains a vast number of redundant attack samples (Tavallaee *et al*., 2009). It consists of normal traffic and four attack categories depicting the real-world network intrusion attacks data, namely (a) User-to-Local (U2L), (b) Root-To-Local attacks (R2L), (c) Denial-of-Service (DoS), (d) Probe attacks. The dataset contains 39 attacks, each of which is categorized into one of the four categories, namely U2R, R2L, U2R, DoS and probe. However, some of the attacks are not available in the training set. In other words, it is available only in the testing set. Thus, this gives a challenging task to classify the test samples correctly. List of intrusion attack software tools used in NSL-KDD as follows:

- User-to-local (U2R): Rootkit, perl, load module, xterm, ps, sqlattack, buffer overflow, Http tuneel, Xterm, Ps, SQL attack
- Root-to-local attacks (R2L): Warezmaster, warezclient, spy, snmpgetattack, named, xlock, xsnoop, send mail, http tunnel, worm, snmp guess, multihop, phf, multihop, imap, guess passwd, ftp write, Named, Sendmail, SnmpGetAttack, SnmpGuess, Worm, Xsnoop, Xlock
- Denial-of-Service (DoS): Mail bomb, teardrop, smurf, pod, neptune, land, back, Apache 2, udpstrom, processtable, Apache 2, Mailbomb, Processtable, UD Pstorm
- Probe attacks: Satan, portsweep, nmap, ipsweep, Saint, Mscan

In this study, 16047 intrusion samples are used to train models and 22554 intrusion samples for testing. Note that all the training samples are taken randomly from the dataset. The number of NSL-KDD samples used for training and testing is shown in Table 5. One of the challenging tasks to train SVM models is from a few samples, i.e., U2R and R2L attacks. One reason it is easy to overfit the SVM models due to the imbalanced distribution of intrusion attack data. As a result, the original training data clearly tend towards the negative label.

Attack Features

The number of attack features used in the experiment is 41 features. These features can be categorized as follows (Latah and Toker, 2018):

- Basic features (10 features): Duration, protocol type, service, flag, source bytes, destination bytes, land, wrong fragment, urgent and hot
- Content features (12 features): Number failed logins, logged in, num compromised, root shell, su attempted, num root, num file creations, num shells, num access files, num outbound cmds, is host login and is guest login
- Time-based features (9 features): Count, srv count, error rate, serve error rate, error rate, srv error rate, same srv rate, diff srv rate and srv diff host rate
- Host-based features (10 features): Dst host count, dst host srv count, dst host name srv rate, dst host diff srv rate, dst host same src port rate, dst host srv diff host rate, dst host serror rate, dst host srv serror rate, dst host error rate and dst host srv error rate

All these feature values in the training and testing dataset are normalized to the interval [-1,+1] as in Eq. (7). In this case, normalization eliminates numerical issues during the calculation and ensures that the most significant numbers do not overwhelm the less significant ones.

Before the experiments can be done, the first is to make the dataset available for SVM data processing as follows:

- Data transforming - NSL-KDD contains both numeric and character data values. For example, the character values are TCP, ICMP, UDP, etc., for protocol type, AOL, AUTH, BGP, etc., for services and OTH, REJ, RSTR, etc. flag. These data, of course, can't be processed by SVM that required numeric data. Thus, we need to convert the character data to numeric data

- Data normalization - Data normalization is one of the essential steps in SVM. The main advantage of scaling or normalization is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation. Because kernel values usually depend on the inner products of feature vectors, e.g., the linear kernel and the RBF kernel, large attribute values might cause numerical problems (Hsu *et al*., 2008). In this study, all feature vectors are normalized or scaled to the range of [-1, +1] using the Eq. (7)
- Class labelling - NSL-KDD contains normal records (normal) and records of intrusion attacks i.e. U2R, R2L, DoS, Probe. To train a binary classifier, we regard the label of +1 and 1. And for the multi-class classifier, we regard the label of 1,2,3,4 and 5. These class labels have fulfilled the requirement of the lib SVM software (Hsu *et al*., 2008)

Experimental Set-up

In addition, attack models are trained using the SVM with RBF non-linear kernel. We chose the SVM because it is a state-of-the-art machine learning algorithm and gives good results as reported in many classification benchmark datasets. However, the RBF kernel is sensitive to the hyperparameters, i.e., C and γ. Thus the grid-search algorithm is employed to determine these values.

Evaluation Metric

The classifier performance is evaluated using the standard accuracy metric as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

where, *TP* (*true positives*) specifies the attack correctly predicted, *TN* (*true negatives*) indicates the attack is identified as correct, *FP* (*false positives*) denotes the attack wrongly assumed as other attacks and *FN* (*false negatives*) specifies the abnormal performance that is misdirected as normal.

Results on NSL-KDD

Table 6 shows the average classification accuracy results of the different multi-class classification problems to classify intrusion attacks correctly using the SVM with RBF kernel on the training dataset. The overall predictive accuracy of the 5-fold cross-validation is increased from 95.39 to 99.80%. The performance is attributed to the hierarchical structure capable of retaining the interaction between the classes in the proposed SHSVM.

Based on the previous experiment results (Table 6), we extend the experiments on the whole intrusion test samples. However, one challenge in this experiment is that some new intrusion test samples are introduced for the testing (not available in the training dataset) as follows:

- U2R-Httptuneel, Xterm, Ps, SQ Lattack
- R2L-Named, Sendmail, Snmp Get Attack, Snmp Guess, Worm, Xsnoop, Xlock
- Dos-Apache 2, Mailbomb, Processtable, UDP storm
- Probe-Saint, Mscan

Table 7 shows the overall results of the test samples. The experiment results show that SHSVM gives an accuracy of 90.98% higher than standard SVM (Pervez and Farid, 2014). These results indicate that using hierarchical learning is beneficial towards improving multi-class classification problems. The complexity of the classification stage can be reduced by ranking SVM classifiers. Moreover, it reduces the class of decision functions and the number of intrusion samples for training. Thus, it may give some advantages to classify intrusion attacks by starting at the well-known binary classifier (DoS vs. others) at first. It is then divided into separate strong and weak sub-tasks based on the distinctive features. As a result, this motivates the more robust classifier to produce correct results since they have rich information about the considered task. It also shows that cross-validation can be used as an approach in ranking to measure the importance of the best model in the SHSVM architecture. Thus, using the multi-class hierarchical relationship from the strong sequence dependency can better generalize attack classification performance.

Table 5: NSL-KDD - Number of training and test samples used for experiments

Attack class	Train	Test
Normal	5000	9711
U2R	52	67
R2L	995	2887
DoS	5000	7458
Probe	5000	2421
Total	16047	22544

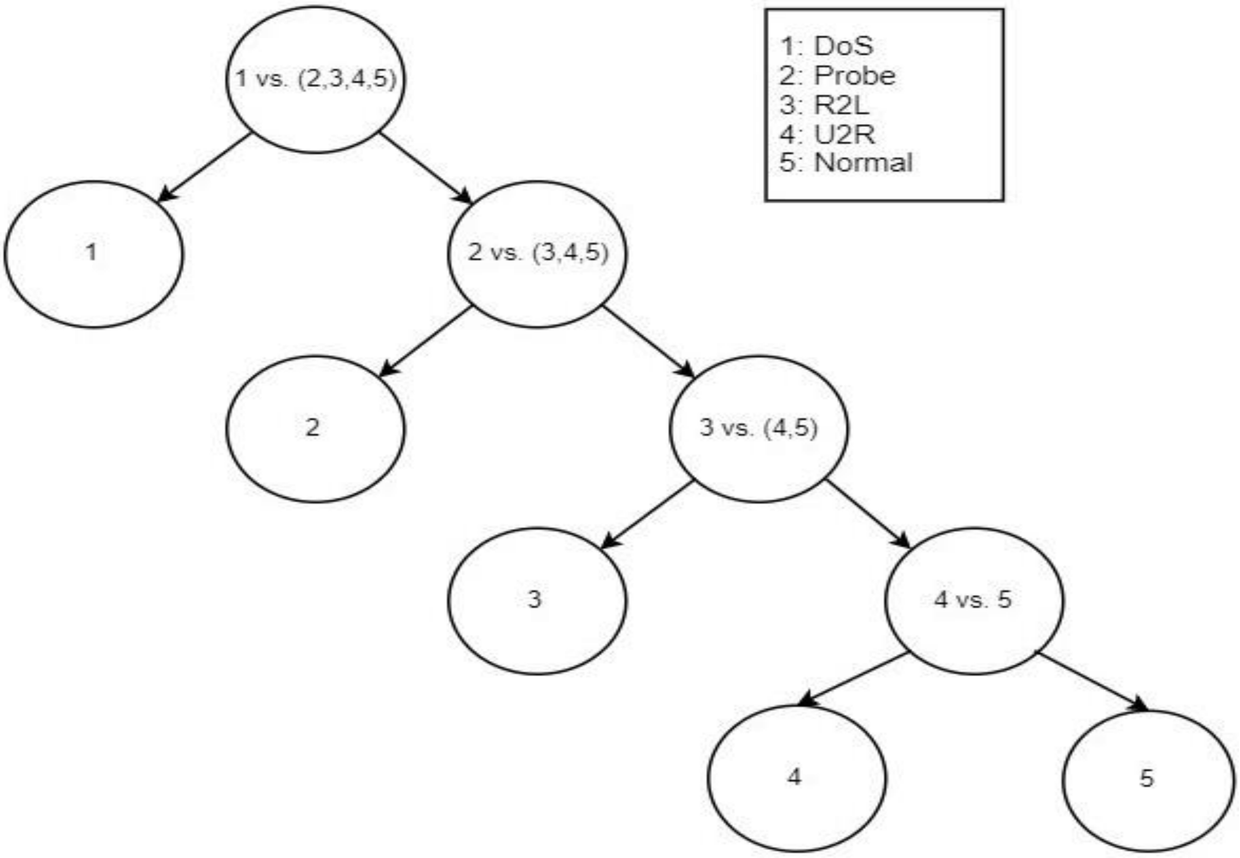

Fig. 2: The proposed hierarchical SVM classifiers for intrusion detection. 1 = DoS, 2 = Probe, 3 = R2L, 4 = U2R and 5 = Normal

Table 6: Classification results on NSL-KDD training dataset

	Standard OVO (%)	SHSVM (%)
Accuracy	95.39	99.80

Table 7: Classification results on NSL-KDD Test dataset

	Standard OVO (%)	SHSVM (%)
Accuracy	83.50	90.98

Table 8: Performance comparison of different intrusion detection methods on NSL-KDD test dataset

Methods	Accuracy (%)
Naive Bayes (Deshmukh *et al.*, 2014)	90.10
SVC (De La Hoz *et al.*, 2013)	89.70
SVM-IG (Mugabo, E. *et al.*, 2020)	86.76
FA-SVM (Al-Yaseen, 2019)	83.70
SVM (Nguyen *et al.*, 2012)	88.32
NNRw (Raza *et al.*, 2017)	84.12
Deep Learning (Nguyen *et al.*, 2018)	90.99

In contrast, the results showed that the standard OVO multi-class training strategy of SVMs is easy to overfit. For example, train SVMs using the OVO with imbalanced data such as U2R (52 attack samples) and R2L (995 attack samples) allows the SVM models to memorize specific data points and cause overfitting and poor generalization to the test of intrusion samples (Boyle, 2019). In addition, the OVO strategy is also sensitive to unknown intrusion attack categories that available only in the testing set. Table 8 shows the result of other approaches using the same experiment setup of the NSL-KDD dataset. It demonstrates that the proposed method (SHSVM) is more accurate than previous approaches and produces comparable results to the state-of-the-art techniques.

Conclusion

We have introduced a method based on hierarchical SVM classifiers to classify attack records in the NSL-KDD dataset. The algorithm starts at the root or first level with the most robust model and works its way down with the weaker models until only one class remains at the node. Finally, using the 5-cross validation procedure on various experiments, the most resilient models are picked and ranked. The proposed approach aims to obtain distinctive models for attack classification by starting with ordered levels, namely DoS, Probe, R2L, U2R and Normal. The proposed algorithm is tested on the standard NSL-KDD dataset. The results show that the soft marking scheme approach gives the highest correct classifications than the standard SVM using NSL-KDD.

Acknowledgment

This study has been supported by the Malaysia’s Ministry of Higher Education Fundamental Research Grant FRGS/1/2019/ICT02/UKM/02/8.

Author’s Contributions

Azizi Abdullah: Supervised the problem definition, design method and the content of the paper.

Warhamni Jani@Mokhtar: Designed the scenario of experiments, organized the research project and the content of the paper.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Abdullah, A., Veltkamp, R. C., & Wiering, M. A. (2009, June). Spatial pyramids and two-layer stacking SVM classifiers for image categorization: A comparative study. In 2009 International Joint Conference on Neural Networks (pp. 5-12). IEEE. doi.org/10.1109/IJCNN.2009.5178743

Ahmim, A., & Zine, N. G. (2015). A new hierarchical intrusion detection system based on a binary tree of classifiers. Information and Computer Security. doi.org/10.1108/ICS-04-2013-0031

Ahmim, A., Maglaras, L., Ferrag, M. A., Derdour, M., & Janicke, H. (2019, May). A novel hierarchical intrusion detection system based on decision tree and rules-based models. In 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 228-233). IEEE. doi.org/10.1109/DCOSS.2019.00059

Albashish, D., Sahran, S., Abdullah, A., Adam, A., & Alweshah, M. (2018). A hierarchical classifier for multiclass prostate histopathology image gleason grading. Journal of Information and Communication Technology, 17(2), 323-346. doi.org/10.32890/jict2018.17.2.7

Al-Yaseen, W. L. (2019). Improving intrusion detection system by developing feature selection model based on firefly algorithm and support vector machine. IAENG International Journal of Computer Science, 46(4), 534-540.

Ariff, N. A. M., Abdullah, A., & Nasrudin, M. F. (2018). Experimental Approach Based on Ensemble and Frequent Itemsets Mining for Image Spam Filtering. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-5), 121-126. https://web.archive.org/web/20180413220626id_/http://journal.utem.edu.my/index.php/jtec/article/viewFile/3642/2636

Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Computer Science, 89, 117–123. doi.org/10.1016/j.procs.2016.06.016

Bishop, C. M. (2006). Pattern recognition. Machine learning, 128(9). https://www.academia.edu/download/30428242/bg0137.pdf

Boyle, T. (2019). Dealing with imbalanced data - a guide to effectively handling imbalanced datasets in python. https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18 [Online; posted 4-February-2019].

Cesa-Bianchi, N., Gentile, C., & Zaniboni, L. (2006, June). Hierarchical classification: Combining bayes with svm. In Proceedings of the 23rd international conference on Machine learning (pp. 177-184). doi.org/10.1145/1143844.1143867

Cortes, C., & Vapnik, V. (2004). Support-vector networks. Machine Learning, 20, 273–297. doi.org/10.1007/BF00994018

Cui, Y., Yan, L., Li, S., Xing, H., Pan, W., Zhu, J., & Zheng, X. (2016). SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. Journal of Network and Computer Applications, 68, 65-79. doi.org/10.1016/j.jnca.2016.04.005

Cup, K. (2007). Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

De La Hoz, E., Ortiz, A., Ortega, J., & De La Hoz, E. (2013, September). Network anomaly classification by support vector classifiers ensemble and non-linear projection techniques. In International Conference on Hybrid Artificial Intelligence Systems (pp. 103-111). Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-642-40846-5_11

Deshmukh, D. H., Ghorpade, T., & Padiya, P. (2014, February). Intrusion detection system by improved preprocessing methods and Naïve Bayes classifier using NSL-KDD 99 Dataset. In 2014 International Conference on Electronics and Communication Systems (ICECS) (pp. 1-7). IEEE. doi.org/10.1109/ECS.2014.6892542

Duin, R. P., & Pekalska, E. (2005). Open issues in pattern recognition. In Computer Recognition Systems (pp. 27-42). Springer, Berlin, Heidelberg. doi.org/10.1007/3-540-32390-2_3

Ergen, T., & Kozat, S. S. (2019). Unsupervised anomaly detection with LSTM neural networks. IEEE transactions on neural networks and learning systems, 31(8), 3127-3141. doi.org/10.1109/TNNLS.2019.2935975

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers and security, 28(1-2), 18-28. doi.org/10.1016/j.cose.2008.08.003

Gharibian, F., & Ghorbani, A. A. (2007, May). Comparative study of supervised machine learning techniques for intrusion detection. In Fifth Annual Conference on Communication Networks and Services Research (CNSR'07) (pp. 350-358). IEEE. doi.org/10.1109/CNSR.2007.22

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 79(8), 2554-2558. https://www.pnas.org/content/79/8/2554.short

Hsu, C. W., Chang, C. C., & Lin, C. J. (2008). A practical guide to support vector classication. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University. http://www-personal.umich.edu/~yongjiaw/NLPproject/guide.pdf

Kadis, M. R., & Abdullah, A. (2017). Global and local clustering soft assignment for intrusion detection system: A comparative study. Asia-Pacific Journal of Information Technology and Multimedia, 6(1), 57–67. doi.org/10.17576/apjitm-2017-0601-05

Kang, S. H., & Kim, K. J. (2016). A feature selection approach to find optimal feature subsets for the network intrusion detection system. Cluster Computing, 19(1), 325-333. doi.org/10.1007/s10586-015-0527-8

Laskov, P., Düssel, P., Schäfer, C., & Rieck, K. (2005, September). Learning intrusion detection: Supervised or unsupervised?. In International Conference on Image Analysis and Processing (pp. 50-57). Springer, Berlin, Heidelberg. doi.org/10.1007/11553595_6

Latah, M., & Toker, L. (2018). Towards an efficient anomaly-based intrusion detection for software-defined networks. IET networks, 7(6), 453-459. doi.org/10.1049/iet-net.2018.5080

Lee, W., & Stolfo, S. J. (2000). A framework for constructing features and models for intrusion detection systems. ACM transactions on Information and system security (TiSSEC), 3(4), 227-261. doi.org/10.1145/382912.382914

Mugabo, E., & Zhang, Q. Y. (2020). Intrusion Detection Method Based on Support Vector Machine and Information Gain for Mobile Cloud Computing. Int. J. Netw. Secur., 22(2), 231-241.

Nashat, S., Abdullah, A., Aramvith, S. and Abdullah, M. Z. (2011). Support vector machine approach to realtime inspection of biscuits on moving conveyor belt. Computers and Electronics in Agriculture, 75(1):147 – 158. doi.org/10.1016/j.compag.2010.10.010

Nguyen, H. T., Franke, K., & Petrovic, S. (2012). Feature extraction methods for intrusion detection systems. Threats, Countermeasures and Advances in Applied Information Security, 3, 23–52. doi.org/10.4018/978-1-4666-0978-5.ch002

Nguyen, K. K., Hoang, D. T., Niyato, D., Wang, P., Nguyen, D., & Dutkiewicz, E. (2018, April). Cyberattack detection in mobile cloud computing: A deep learning approach. In 2018 IEEE wireless communications and networking conference (WCNC) (pp. 1-6). IEEE. doi.org/10.1109/WCNC.2018.8376973

Panetta, K. (2017). 5 trends in cybersecurity for 2017 and 2018. https://www.gartner.com/smarterwithgartner/5-trends-in-cybersecurity-for-2017-and-2018/ [Online; posted 14-June-2017].

Pervez, M. S., & Farid, D. M. (2014, December). Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014) (pp. 1-6). IEEE.doi.org/10.1109/SKIMA.2014.7083539

Raza, R.A., Wang, X., Huang, J., Abbas, H., & He, Y. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. Inf. Sci., 378, 484-497.

Staudemeyer, R. C., & Omlin, C. W. (2014). Extracting salient features for network intrusion detection using machine learning methods. South African computer journal, 52(1), 82-96. doi.org/10.18489/sacj.v52i0.200

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). IEEE. doi.org/10.1109/CISDA.2009.5356528

Tolles, J., & Meurer, W. J. (2016). Logistic regression: Relating patient characteristics to outcomes. Jama, 316(5), 533-534. doi.org/10.1001/jama.2016.7653

Wang, A., Liu, J., Wang, H., & Tao, R. (2007, September). A novel fault diagnosis of analog circuit algorithm based on incomplete wavelet packet transform and improved balanced binary-tree SVMs. In International Conference on Life System Modeling and Simulation (pp. 482-493). Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-540-74769-7_52

Zhang, Z. L., Luo, X. G., González, S., García, S., & Herrera, F. (2018). DRCW-ASEG: One-versus-one distance-based relative competence weighting with adaptive synthetic example generation for multi-class imbalanced datasets. Neurocomputing, 285, 176-187. doi.org/10.1016/j.neucom.2018.01.039

Zolfi, H., Ghorbani, H., & Ahmadzadegan, M. H. (2019, December). Investigation and classification of cyber-crimes through IDS and SVM algorithm. In 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 180-187). IEEE. doi.org/10.1109/I-SMAC47947.2019.9032536