2020

Proportional Voting based Semi-Unsupervised Machine Learning Intrusion Detection System

Yang G. Kim
CUNY Queensborough Community College

Ohbong Kwon
CUNY New York City College of Technology

John Yoon
Mercy College - Main Campus

Journal of Computer Science and Information Technology
December 2020, Vol. 8, No. 2, pp. 1-9
ISSN: 2334-2366 (Print), 2334-2374 (Online)

Published by American Research Institute for Policy Development
DOI: 10.15640/jcsit.v8n2a1
URL: https://doi.org/10.15640/jcsit.v8n2a1

Proportional Voting based Semi-Unsupervised Machine Learning Intrusion Detection System

Yang G. Kim[1], Ohbong Kwon[2] & John Yoon[3]

Abstract

Feature selection of NSL-KDD data set is usually done by finding co-relationships among features, irrespective of target prediction. We aim to determine the relationship between features and target goals to facilitate different target detection goals regardless of the correlated feature selection. The unbalanced data structure in NSL-KDD data can be relaxed by Proportional Representation (PR). However, adopting PR would deny the notion of winner-take-all by attracting a majority of the vote and also provide a fairly proportional share for any grouping of like-minded data. Furthermore, minorities and majorities would get a fair share of power and representation in data structure distribution. Particle Swarm Optimization (PSO) utilizes attack data for minority while majority employs non-attack data along with targeted classes to increase detection rate and reduce false alarms, especially for R2L and U2R attacks, as the output target goal influences feature selections and corresponding detection rate and false alarm rate. Our simulation study confirms the feasibility of the Voting Representation for minority protection and increased detection rate while reducing false alarms, which is favorable to minority over the majority.

Keywords: Intrusion Detection System, Particle Swarm Optimization, Machine Learning, Supervised and Unsupervised Learning, Anomalous Detection Algorithm, Clustering

1. Introduction

Artificial Intelligence (AI) could be categorized as either inductive or deductive. For inductive learning, there is a reason for the selection, while deductive learning does not have a specific reason for the selection. Machine Learning (ML) means the machine learns from the data, and ML adapts to different situations through trial and error while recognizing the pattern. ML provides an answer without explicitly explaining the reason due to deductive learning similar to how we could follow our instincts without a reasonable explanation. In the future, the ultimate goal of inductive and deductive learning would be to unite them, mimicking how the human brain functions. Machine Learning algorithm employed in Intrusion Detection System (IDS) categorizes into supervised and unsupervised learning, and the difference is analogous to having a class with or without a teacher. Based on our previous work [1], the IDS can be utilized by PSO and K-means for global optimal solution and local optimal solution, respectively. [2] and [3] are hybrids of K-means and PSO that reinforce K-means' weakness. K-means easily falls into local minima due to initial random value and the number of clusters. However, PSO is efficient at finding the global minimum with reasonable complexity by performing both exploitation and exploration in a search space. [4] combines K-means, Fuzzy K-means, and PSO. Although it resolves the local convergence problem in Fuzzy K-means by PSO and the sharp boundary problem in PSO by K-means, false alarm rate remains relatively high. [5] uses the Learning Process for its own predictions to teach itself through self-training in which it is first trained with labeled data.

[1] Department of Engineering Technology, Queensborough Community College, 222-05 56th Ave, Bayside, NY 1136, Email: yakim@qcc.cuny.edu, Phone: 718-631-6207
[2] Department of Computer Engineering Technology, New York City College of Technology, 300 Jay St., Brooklyn, NY 11201, Email: okwon@citytech.cuny.edu, Phone: 718-260-5439, Fax: 718-260-5425
[3] Department of Math/Computer Science/Cybersecurity, Mercy College, 555 Broadway Dobbs Ferry NY 10522, Email: jyoon@mercy.edu, Phone: 914-674-7461

The unlabeled data with their predicted labels is then utilized to predict other unlabeled data, from which similarity function is maximized based on the knowledge that higher similarity means the same class, i.e., minimum distance indicates the same class. [6] utilizes two different data types: labeled and unlabeled with the USPS (US postal service) handwritten dataset being applied while formulating two different objective functions for each with a weight factor, β. If β = 0, it is unlabeled data, which is unsupervised Machine Learning. On the other hand, if β = 1, it is labeled data, which is supervised Machine Learning. We utilize unsupervised Machine Learning in which the minority refers to attack data samples while the majority refers to non-attack data samples for intrusion detection system.

Non-attack data is more common than attack data, so most researchers have utilized only non-attack data as training model to predict whether new data is normal or an attack. For DoS and Probe data, the prediction rate is sufficient even though only non-attack data is trained due to sheer amount of data. However, the prediction rates for R2L and U2R attacks are insufficient because the amount of data for R2L and U2R attacks is not enough to discern compared to that of DoS and Probe. One of the most important deficiencies in the NSL-KDD data set is the huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records, and this bias prevents the learning algorithms from learning infrequent records which are usually harmful to networks, especially an intrusion into high-classified network through U2R. In addition, the existence of these repeated records in the test set will cause the evaluation results to be biased for the methods that have better detection rates on the frequent records while suppressing infrequent data. Many of the previous works disregard the suppression of minority detection rate. However, the proposal utilizes both non-attack and attack data to reveal the suppressed data that would be an indicator of directionality, toward which particles in PSO will search.

2. Feature Selction

The following is our algorithm procedure: feature selections by target-based correlation feature selection scheme. In previous works, the features are selected based on favorability to normal data (**favorable to majority**) because the amount of normal data is dominant over attack data samples. In addition, feature selection has been done without any consistency between feature selection and detection algorithm, i.e., feature selection is based on supervised learning while detection algorithm is based on unsupervised learning (e.g., classification), regardless of target goals. In IDS, feature selection would be done with information gain based on prior knowledge [7]. The information gain is the difference between the prior entropy (e.g., knowledge) and the selected feature entropy where the highest information gain calculated based on labeled data is selected. Information gain algorithm is not feasible without the labeled data due to its characteristic of supervised learning, so we apply information gain as verification for our target-based correlation feature selection scheme.

Unlike feature selection by information gain, correlation feature selection could be performed on either supervised or unsupervised learning. In previous works, correlation feature selection is done in an unsupervised way it performs features with one another, ignorant of the target goal. Without considering the target goal, i.e., feature selection among features, that is not consistent with the intrusion detection system that contains different target goals while there are different number of data sets. We select features based on correlation with feature and a specific target goal instead of simply selecting correlation of the features with one another. From our observation, we are able to determine the consistency between feature selection and detection method while applying target goal into feature selection and detection method. The reduction of the number of correlations among normal data causes a decrease in the normal detection rate while increasing the detection rate for both R2L and U2R. Our goal is to select the right features based on attack type to increase the detection rate while reducing false alarms, especially in U2R and R2L (**favorable to minority**). After the feature selection, the number of clustering can be determined by silhouette clustering for each target goal along with five attacks and Normal.

3. Selection Of The Number Of Clusters By Silhouettes

Silhouette [8] refers to a method of interpretation and validation of consistency within clusters of data. The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). Let $a(i)$ be the average of distance between *i* and all other data written the same cluster and $b(i)$ be the smallest average distance of *i* to all points in any cluster, of which *i* is not a member. The number $s(i)$ is obtained by combining $a(i)$ and $b(i)$ as follows:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & if\ a(i) < b(i)) \\ 0, & if\ a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1\,, & if a(i) > b(i) \end{cases}$$

which can be written as single formula:

$$s(i) = \frac{b(i) - a(i)}{max\,\{a(i), b(i)\}}$$

So, it is clear that

$$-1 \leq s(i) \ll 1$$

The silhouette ranges from -1 to $+1$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

4. Particle Swarm Optimization

The notation of real-valued PSO is as follows. N_a denotes the total number of particles. *Let* X_a^i = (x_{a1}^i,x_{a2}^i, ……,x_{aD}^i), where$x_{ad}^i \in \Re^2$, be the particle *a* in *D* (two dimensions) at iteration *i*. Each particle is represented by a position in the search space as X_a^i, which is a potential solution. Denote the velocity as V_a^i = (v_{a1}^i,v_{a2}^i, ……,v_{aD}^i), where$v_{ad}^i \in \Re^2$. Let P_a^i = (p_{a1}^i,p_{a2}^i, ……,p_{aD}^i) be the personal best that particle *a* has obtained until iteration *i*, and P_g^i = (p_{g1}^i,p_{g2}^i, ……,p_{gD}^i) be the global best obtained fromp_a^i at iteration *i*. The movements of the particles in the real-valued PSO are governed by the following [1].

$$v_{ad}^i = w^i * v_{ad}^{i-1} + c_1^i * \mathrm{r}_1 * \left(p_{ad}^{i-1} - x_{ad}^{i-1}\right) + c_2^i * \mathrm{r}_2 * \left(p_{gd}^{i-1} - x_{ad}^{i-1}\right) \quad (1)$$

$$x_{ad}^i = x_{ad}^{i-1} + v_{ad}^i \quad (2)$$

There are three important parameters in Eq. 1 directly affecting the particle behaviors: w^i,c_1^i and c_2^i. The w^i represents inertia weight, which provides the global search ability (exploration) at the beginning and then the local search ability (exploitation) at the end of the process. Thus, w^i varies as *i* progresses because the particle initially moves fast, and then slows down as it approaches the target to avoid overflying [9]. The "cognitive" coefficient c_1^i and the "social" coefficient c_2^i define how fast each particle moves towards p_{ad}^{i+1} and p_{gd}^{i+1} positions. Therefore, as with w^i, varying c_1^i and c_2^i not only promotes exploration of a remote target, but also encourages exploitation at a nearby target. If p_{gd}^i for a particle is selected more often than others, the likelihood of the particle being on the right track towards the global best solution increases, so c_1^i increases, and the other particles would be more likely to follow that direction. In contrast, if p_{gd}^i for a particle is selected less often, c_2^i increases because the solution quality of that particle is poor compared to those of other particles, and that particle follows another direction. Consequently, each particle updates its velocity and position depending on the frequency of p_{gd}^i . r_1 and r_2 are random numbers uniformly distributed within [0,1). A maximum velocity $\pm V_{max}$ (½ of *D*) is necessary, not only to prevent a particle from escaping the search space, but also to provide the particle a high rate of self-mutation.

5. Semi-Unsupervised IDS Algorithm Based On Particle Swarm Optimization

Semi-Unsupervised PSO-based IDS on the dataset *X* with target class *C* and the number of features *D* can be seen as searching for the optimal positions for the centroids of data clusters in a *D*-dimensional space [6]. The position of each particle contains *K* centroids along with dimensional variable *D* in which each particle has its own position and velocity with fitness function. Each particle maintains a matrix $M_i = (C_1, C_2, \ldots, C_i, \ldots, C_k)$ where C_i is the *i*th cluster centroids and *K* is the number of clusters. The dimension of each particle equals the product of the number of features *D* and the number of targets class *C*.

Fitness function plays an important role in PSO because an efficient fitness function can quickly find the optimization positions of the particles. The fitness function is computed as the sum of the Euclidean distances between all the training samples and the centroids being encoded in the particle they belong to.

The minority data samples, e.g., the target data samples, are too few to represent the real distribution of dataset while non-attack data samples are abundant, so the combination may be helpful to capture the minority data pattern in order to avoid the shadowiness over dominant data. Therefore, we modify the fitness function by introducing the structure information of non-attack type dataset samples to that of attack type dataset samples. With the assumption that the neighborhood dataset should have the same targets type with a proportional ratio, we propose to use a new fitness function in our proposed unsupervised PSO-based IDS, and the fitness of the *i*th particle is defined as:

$$\varphi(p_i) = \alpha \frac{1}{N}\frac{1}{U_K}\sum_{i=1}^{N}\sum_{j=1}^{U_K} w_{i,j}\ ||X_j - C_{Target\ (j),i}|| + \beta \frac{1}{N}\frac{1}{U_N}\sum_{i=1}^{N}\sum_{k=1}^{U_N} w_{i,j}\ \min\{d\left(X_k, C_{1,i}\right), d\left(X_k, C_{2,i}\right), \ldots, d\left(X_k, C_{Target\ ,i}\right)\} \tag{3}$$

$$where, w_{i,j} = \frac{1}{\sum_{k=1}^{N_c}\left\{\frac{||X_i - C_j||}{||X_i - C_k||}\right\}^2}$$

$\varphi(p_i)$ is the fitness value of the *i*th particle. Target *(j)* denotes the target class and X_j is the number of the training sample for the target data, $C_{Target(j),i}$ denotes the centroid vector of the target class*(j)* encoded in the *i*th particle, and $||X_j - C_{Target\ (j),i}||$ is d $(x_j, C_{Target(j),i})$ that is the Euclidean distance between the training sample X_j and the target centroid $C_{Target(j),i}$. α and β are a weight factor in the range between [0,1], which controls the ratio of the information obtained from the attack and non-attack dataset samples. α is ranged from 1% to 42% while β is from 78 to 99%. U_K is the number of attack data and U_N is the number of non-attack data set samples. The number of target data set samples are different in terms of specific target goal, which is dominant in sequence, DoS, Probe, R2L, and U2R. The number of non-attack data samples is maintained under different attack type goal. $w_{i,j}$ is selected proportionally depending on the most important feature at the start values (e.g., 25%, 20%, 15%, 10%, 10%, 5%, 5%, and 5% for eight feature selections) and eventually will become binary distribution while updating data samples with the centroids (p_{gd}^i), in which in-cluster data become closer and closer while out-cluster data become farther and farther. Eventually, it becomes noticeable which data belongs to which cluster to differentiate between attack and non-attack data while making a noticeable difference between target data's centroids and non-attack data's centroids. The first term on the right side of the fitness function is favorable to minority dataset while the second term is favorable to majority dataset.

Proportional Voting based Semi-Unsupervised PSO-based IDS algorithm

Input. The non-attack dataset is $X_U = \{x_1, x_2, \ldots, x_u\}$. The target dataset is $X_K = \{x_1, x_2, \ldots, x_k\}$ and the corresponding target is $Y_L = \{y_1, y_2, \ldots, y_l\}$.

*1.*Load training dataset with target and non-attack dataset samples while normalizing [0,1].

2. Initialize $w^i, c_1^i, c_2^i, \pm V_{max}^i$, and direction (X_K).

*3.*Selecting the number of *N* particles and generating both the position and velocity vectors for each particle.

3-1. Calculate the fitness value $\varphi(p_{ad}^i)$ for each particle in each iteration with [3].

3-2. Update the best fitness value $\varphi(p_{gd}^i)$ and the best particle of ith particle p_{gd}^i; that is, if $\varphi(p_{ad}^i) < \varphi(p_{gd}^i)$, then $\varphi(p_{gd}^i) = \varphi(p_i^t)$, and $p_{gd}^i = p_{ad}^i$.

*3-3.*If necessary, update the global best particle p_{gd}^i ; that is, $b^t = \arg min_{p^t}\{\varphi(p_{a1}^i),\ \varphi(p_{a2}^i), \ldots, \varphi(p_{1D}^i)\}$, if $\varphi(b^t) < \varphi(p_{gd}^i)$, then $\varphi(p_{gd}^i) = \varphi(b^t)$ and $p_{gd}^i = b^t$.

3-4. End, update p_{ad}^i with p_{gd}^i.

3-5. Update particles' velocity with (1).

3-6. Update particles' position with (2).

4. Iterate until the maximum number of iterations is reached.

Output. The structure of the clustering.

The cluster shape of the output with optimum centroids represented by *gbest* could be changed, so we utilize "proportional minority vote" to determine whether data is normal or an attack.

In the clusters, a dataset for each cluster "votes" individually and majority voter wins over the minority unless the minority is occupied by 25%, so its cluster declares whether data is normal or an attack.

6. Proportional Voting Based *k*NN Neighborhood Detection Algorithm

After modeling the training data by completely Semi-Unsupervised PSO-based IDS, a detection algorithm is utilized for kNN algorithm in which new data is normalized with [3], distance from the closest centroids is calculated, and the closest centroid is then selected. The new data can be determined to be either normal or an attack based on the data's distance to the closest centroid that has already been declared either normal or an attack. In addition, the kNN collects *k*-neighbors within the distance and those neighbors having already named attack or normal are divided into majority and minority. The algorithm then applies a proportional representation. For example, if there is more than 25% minority, the data will be turned into an attack. Otherwise, the data is normal. The final decision is followed by AND logic operation, i.e., 1 AND 1 = 1, otherwise 0. kNN algorithm is more suitable for applying proportional minority voting due to flexibility of a decision process and our data set attribute is convenient to kNN while verifying the selection over other ML algorithms, Linear Regression, Decision Tree, and Random Forest during the simulation. Therefore, our prediction algorithm is based on kNN along with proportional minority voting.

Proportional Voting based *k*NN Neighborhood Detection Algorithm
Input. Training data: $T = \{ (x_i, y_i) \}$, $X_i \in \Re^2$
Each cluster declares whether data is an attack or not with **proportional minority vote** (25% minority protection)
Prediction.
1. A positive integer k (*min to max spillover)* is estimated by Hierarchical clustering.
2. A new data x_0 predicts y_0
    2.1 The new data applies **proportional minority vote** for neighbors as well as the inverse of the distance between the centers.
    2.2 Determined by AND logic, 1 AND 1 = 1, otherwise 0.

Output. Classification $y_i \in [\{1,\ldots, C\}]$.

7. The NSL-KDD dataset and data preprocessing

The NSL-KDD data set [10], which is a refined version of its predecessor KDD'99 data set [7], has been tasked with the WEKA tool to compare three classification algorithms, J48, SVM and Naïve Bayes, while specifying in detail about NSL-KDD data set. There are 41 features for NSL-KDD data structure, such as Basic Features 1-9, Content related Features 10-22, Time related Features 23-31, and Host related Features 32-41. The original data includes discrete attribute features and continuous attribute features. This paper does not process discrete attribute features due to the possibility of misleading while interpolating discrete values [2] and [3], so each data sample contains only 38 attributes. For continuous attribute features, different attributes have different measure standards due to the different magnitudes that would cause large numbers to cover up small numbers, and some attribute features data will be concealed without any contribution. In order to solve this problem, attribute feature value of data must be standardized.

First, we calculate overall dataset samples mean,

$m = \frac{1}{n}\sum_{i=1}^{n} X_i$ for all the training dataset samples.

For each dataset, we can calculate the absolute deviation value from the mean,

$$S = \frac{1}{n}\sum_{i=1}^{n}(X_i - m).$$

Finally, we are able to calculate the standardized data,

$$Y_i = \frac{X_i - m}{S}.$$

This is equivalent to attribute feature of original instance being mapped to standard attribute space by statistical characteristics.

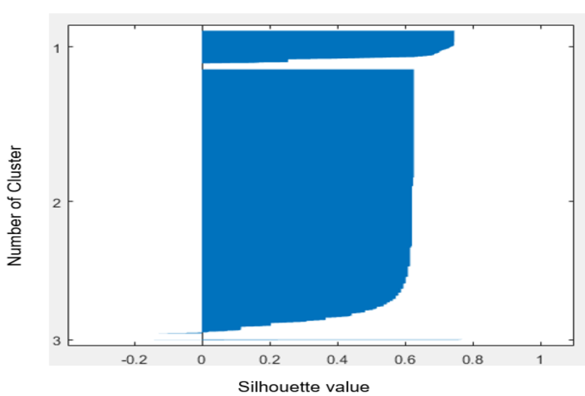
8. Performance Evaluation

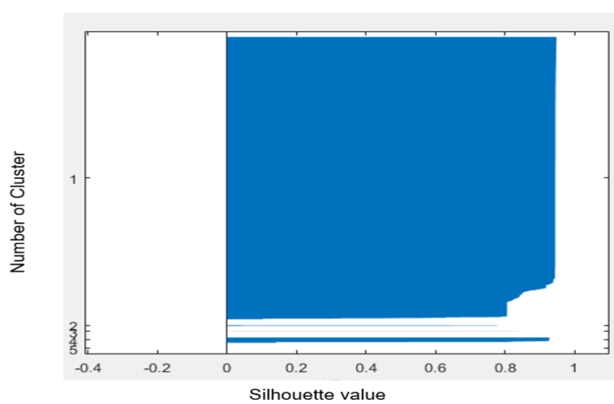
There are five attack target goals, R2L, U2R, DoS, Probe, and Normal. During the simulation, all the five attack types are used at the same time and they extract an individual attack value, e.g., DoS is 99.6%, U2R is 34.32%, R2L is 95.39%, Probe is 97.19%, and Normal is 95.68%. The simulation would be treated differently because it seems unfair for U2R and R2L since the amount of data is way too small compared to the other three. The larger amount of data would be more favorable to be detected than the less amount of data because the subservient data could be overshadowed. Thus, our proposal is to determine the relationship between features and each target goal to increase detection rate for the overshadowed attack data over the dominant data set, regardless of the correlated feature selection. The simulation parameters for our algorithm are the same as in our previous work [1]. First, we select feature selection for each target goal by determining the highest co-relationship between features and each target goal as shown in Figure 1 without projecting any previous entropy values, such as information gain.

Target Goal	Feature Selection
R2L vs Normal	1, 11, 22, 10, 24, 23, 33, 6
U2R vs Normal	17, 16, 13, 14, 10, 1, 24, 38
DoS vs Normal	6, 5, 29, 41, 30, 28, 27, 34
Probe vs Normal	27, 28, 41, 30, 29, 6, 34, 33
R2L, U2R, DoS, Probe vs Normal	5, 6, 27, 28, 29, 30, 33, 35

Table 1. Feature selection with different target goals

R2L vs Normal shows that the first feature is the closest correlation feature. This is reasonable considering how the characteristic of the first feature (length of time duration of the connection) relates to R2L attack. In U2R vs Normal, the closest correlation feature is 17, which is Num_file_creation that also relates to U2R. DoS vs Normal shows that the closest feature is 6, which is the number of data bytes transferred from destination to source in single connection. Probe vs Normal shows that the closest feature is 27, which is the percentage of connections that have activated the flag, R_{error_rate}. On the other hand, R2L, U2R, DoS, Probe, and Normal show that the closest correlation is 5, which is the number of data bytes transferred from source to destination in single connection because the number of data samples is dominant in DoS that matches with the outcome of DoS vs Normal. Therefore, as shown in the figure, feature selection relies solely on the target goal. We determine the number of clusters by Silhouette clustering for R2L vs Normal, U2L vs Normal, DoS vs Normal, and Probe vs Normal, to be 3, 5, 7, and 7, respectively, as shown in Figure 2. R2L vs Normal and U2L vs Normal show clear separation among the number of clusters while DoS vs Normal and Probe vs Normal show some spill over the negative value, in which some data is isolated and clustered poorly. In order to avoid the spill data, we increase the number of clusters, but doing so still does not optimize the number of clusters.





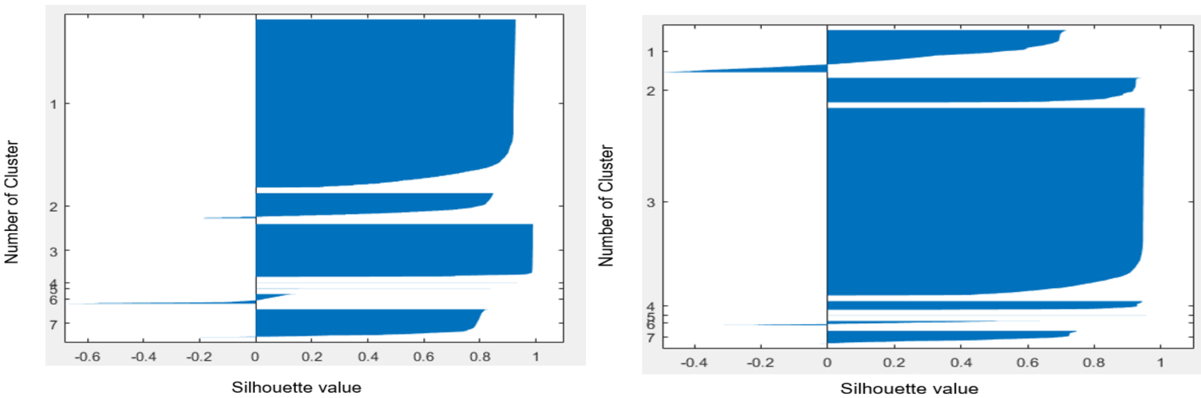


Figure 1. The number of clusters for U2R vs Normal, R2L vs Normal, DoS vs Normal, and Probe vs Normal

There are two simulation methods; the first is for five attack types and the other is for individual attack type. The significantly different results prove that individual attack type simulation is superior to the five attack type simulation. To effectively analyze the data patterns, one should consider each attack type along with normal data and appropriate countermeasures.

Confusion Matrix

a	b
10003	0
67	0

a: Normal
b: U2R
Detection rate: 99.3 %

a	b
9987	16
9	58

a: Normal
b: U2R
Detection rate: 99.8 %

As we can see from the confusion matrix, comparison of favorable minority and non-favorable minority shows significant improvement on U2R detection with a bit increase overall. In the non-favorable minority, the detection rate of U2R is zero, but the overall detection rate is still comparable to that of favorable minority, which most researchers simply ignore. Favorable minority is increased substantially from 0% to 87%, and this increase is significant for a company that is usually attacked by U2L that eventually produces false belief. From here, we can see a theory of "no free lunch for optimization" in the favorable minority. In order to increase detection rate of U2L, the detection rate of Normal must be decreased as seen above.

Attack Type	**Detection Rate**	**False Alarm**
U2R vs. Normal	99.82%	0.002
R2L vs. Normal	99.24%	0.005
DoS vs. Normal	99.30%	0.007
Probe vs. Normal	99.19%	0.006

U2R, R2L, DoS and Probe vs. Normal	**Detection Rate**	**False Alarm**
Favorable to Minority	98.70%	0.0072
Favorable to Majority (NB)	82.16%	0.081
Favorable to Majority (J48)	97.68%	0.0082
Favorable to Majority (SVM)	89.28%	0.063

Table 2. Characteristics of Favorable to Minority and comparison of Favorable to Majority in Five Target Goals

The simulation results demonstrate that our proposal achieves significant improvement in favorable minority detection. All four target goals achieved over 99% detection rate while noticeably reducing false alarm rate to less than 1%.

The comparison of U2R, R2L, DoS, and Probe vs. Normal shows our Favorable to Minority achieved highest detection rate and the lowest false alarm rate.

Favorable to Minority	Detection Rate	False Alarm
Normal	95.18%	0.018
DoS	99.65%	0.006
U2R	86.62%	0.002
R2L	95.39%	0.007
Probe	97.19%	0.003

Favorable to Majority (NB)	Detection Rate	False Alarm
Normal	91.86%	0.04
DoS	89.35%	0.012
U2R	10%	0.001
R2L	72.50%	0.026
Probe	88.48%	0.012

Favorable to Majority (J48)	Detection Rate	False Alarm
Normal	98.30%	0.016
DoS	99.40%	0.003
U2R	58.20%	0.001
R2L	92.62%	0.017
Probe	97.13%	0.004

Favorable to Majority (SVM)	Detection Rate	False Alarm
Normal	93.28%	0.126
DoS	89.91%	0.009
U2R	40.32%	0.001
R2L	69.35%	0.025
Probe	96.72%	0.01

Table 3. Comparison of Favorable to Minority and Favorable to Majority in Five Target Goals

Individual attack type simulation results show that Favorable to Majority NB detection rate for U2R is only 10%. As we can see, the detection rates of U2R and R2L for Favorable to Minority are substantially higher than their counterparts in Favorable to Majority types.

9. Conclusion

Our proposal aim**s** to determine the relationship between features and target goals to facilitate different target detection goals regardless of the correlated feature selection. As unbalanced data would introduce misleading bias, we can mitigate the bias via proportional minority vote without adding more data. Proportional minority vote would provide a fairly proportional share for any groupings of like-minded data. Minorities and majorities get a fair share of power and representation in data structure distribution. Particle Swarm Optimization (PSO) utilizes attack data for minority while majority employs non-attack data along with targeted classes to increase detection rate and reduce false alarms, especially for R2L (Remote to Local) and U2R (User to Root). As the output target goal influences feature selection and corresponding detection rate and false alarm rate, our feature selection utilizes purely unsupervised learning, rather than supervised learning, e.g., information gain. We can increase detection rate and reduce false alarm rate, especially in U2R and R2L, by proportional minority vote that is favorable to minority over majority.

References

Y.G. Kim and M.J. Lee, “Scheduling Multi-channel and Multi-timeslot in Time Constrained Wireless Sensor Networks via Simulated Annealing and Particle Swarm Optimization,” IEEE Communication Magazine, (2014), vol. 52, no. 1, pp. 122-129.

Zhengjie Li, Yongzhong Li and Lei Xu, "Anomaly Intrusion Detection Method Based on K-means Clustering Algorithm with Particle Swarm Optimization" – 2011 International Conference of Information Technology, Computer Engineering and Management Sciences.

Lizhong Xiao,Zhiqing Shao and Gang Liu, "K-means Algorithm Based on Particle Swarm Optimization Algorithm for Anomaly Intrusion Detection", Proceedings of the 6th World Congress on Intelligent Control and Automation.

R Ensafi, and S Dehghanzadeh, "Optimizing Fuzzy K-means for network anomaly detection using PSO", 2008 IEEE/ACS International Conference on Computer Systems and Applications.

Shi Cheng, Yuhui Shi and Quande Qin, "Particle Swarm Optimization based Semi-Supervised Learning on Chinese Text Categorization", WCCI 2012 IEEE World Congress on Computational Intelligence,June, 1-15, 2012-Bribance, Australia.

Xiangrong Zhang,Licheng Jiao,Anand Paul,Yongfu Yuan,Zhengli Weiand Qiang Song, "Semisupervised Particle Swarm Optimization for Classification", Hindawi Publishing Corporation, Mathematical Problems in Engineering Volume 2014.

H. GunesKayacik, A. Nur Zincir-Heywood, and Malcolm I. Heywood, "Selecting Features for Intrusion Detection: A feature Relevance Analysis on KDD 99 Intrusion Detection Datasets" third Annual Conference on Privacy, Security and Trust, October 12-14, 2005, the Fairmont Algonquin, St. Andrews, New Brunswick, Canada.

Rousseeuw and P.J.: Silhouettes, "A graphical aid to the interpretation and validationof cluster analysis". J. Comput. Appl. Math. 20, 53–65 (1987).

T. Wang, Z. Wu, and J. Mao, "PSO-based Hybrid Algorithm for Multi-objective TDMA Scheduling in Wireless Sensor Networks," in Second international Conference onCommunication and Networking, 2007, pp. 850 – 854.

L. Dhanabal and S.P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification", International Journal of Advanced Research in Computer and Communication Engineering Vol.4, Issue 6, June 2015.