Naïve Bayes Classifier and Particle Swarm Optimization Feature Selection Method for Classifying Intrusion Detection System Dataset

A S Talita\textsuperscript{1*}, O S Nataza\textsuperscript{2} and Z Rustam\textsuperscript{3}

\textsuperscript{1}Faculty of Computer Science and Information Technology, Gunadarma University
\textsuperscript{2,3}Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Indonesia

*e-mail: ainikrw@gmail.com

Abstract. The security of a network might be threatened by an intrusion aim to steal classified data or to find weaknesses on the network. In general, network main security systems use a firewall to control and monitor both incoming and outgoing network traffic. Intrusion Detection System can be used to strengthen network security. Several data mining methods have been used to solve Intrusion Detection System (IDS) problem on a network. On this paper we will use Naïve Bayes Classifier along with Particle Swarm Optimization (PSO) as the feature selection method specifically on one of the benchmark dataset on IDS problem, KDD CUP'99. The dataset consists of more than 40 features with more than 400 thousands records. To solve IDS problem on the dataset, it needs a quite expensive cost either on time computation or memory usage hence the use of PSO as the feature selection method. The best classification result was reached when we use 38 features where the accuracy is 99.12%. Particle Swarm Optimization method has several parameters that may affect the classification performance. For future improvement, it is possible to use a parameter optimization method to ensure the best classifier performance.

Keywords: Classification, Intrusion Detection, Naïve Bayes, Particle Swarm Optimization

1. Introduction

Nowadays computer networks held an important role since it enable people to gain information, stored it, and manipulated it online. With the increasing on the global internet user, it comes the increasing of network system. It attracts some unwanted actions such as network intrusion both from inside the system or by outsider. The network security can be compromised by the intrusion, that is any action that endangered integrity, confidentiality, or data availability \cite{1}. Network manager use firewall to strengthen the network security, but there are some applications that could not get its full protection features. On this kind of event, another additional tools was needed to strengthen the network security against any intrusions. One of the possibility is to use Intrusion Detection System (IDS). It is a system that can detect any dangerous intrusion towards the network \cite{2}. IDS can be categorized into two main categories, Misuse Detection and Anomaly based detection system. Misuse detection based IDSs are able to detect known attacks accurately, but it is failed to detect unknown attacks. Anomaly detection
based IDS are able to detect unknown attacks since it compares a new behavior with normal behavior in order to find anomalies.

There are several approaches on solving IDS related problem. One of the example is data mining methods. It aims to extract information automatically from a large database [3]. Specifically, classification technique was commonly used to solve IDS related problems to differ between attacks and normal behaviors.

On the previous researches, there are several different classification techniques that were used on IDS problem. Kabir, Onik, and Samad use Bayesian network method on [4], while Neural Network was used on [5]. Support Vector Machine (SVM) one of the most powerful classification technique was used on [6]. Varuna and Ramsya on [7] use Naïve Bayes method along with Binary Bat algorithm with accuracy 92.24%. While Shrivastava and Richariya on [8] use Naïve Bayes classifier and Ant Colony Optimization with accuracy reaches 97%.

For problems with a highly dimensional dataset, it is quite common to implement a feature selection or feature extraction method in order to reduce the computational cost. Computational cost here might be related with time computation or memory usage. Some feature selection methods used on previous research are Ant Colony Optimization on [9] and Genetic Algorithm [10].

There are several techniques on feature selection methods. On this paper we use Particle Swarm Optimization that choose feature values by training model on subsets of the features. This type of feature selection performances depends on classifier that being used. Another example on this type of feature selection methods are Genetics Algorithm and Greedy Search.

Particle Swarm Organization (PSO) is one of the metaheuristics method to solve optimization problems inspired by social behavior and movement dynamics of animals swarm as in fish or birds. On PSO, the swarms was treated as set of particles that have positions and speed, and each of its individual (particle) was treated as a point. Each particle moves on a certain space and “remember” its best position according to its position towards source of foods. When a particle find the best path to food sources, it will pass the information to other particles that will adjust their position and speed based on the information. On PSO, each particle has a fitness value that need to be evaluated at each generation based on global best (gbest) and personal best (pbest) value. These values are obtained based on the particles’ experiences to attain the best solution. Parameter pbest is the best position of a particle based on fitness values on the previous position and current position, while gbest is the best position of a particle relative to fitness values of the whole particles at the swarm. The particles’ experiences can be used as inertia weight (\(\omega\)), a parameter that reflect the change of particles’ velocity. Velocity of a particle (\(v\)) is a vector that determine the particle position’s change of direction.

Cognitive and social parameter (\(c_1\) and \(c_2\)) represents the abilities of particles and swarm as a population. According to Chen and Shih on [11], the position of each particle may considered as a solution candidate. We call it fitness value. It was computed by objective function of the optimization problem. When a particle moves to a certain new position, it will be a new pbest. And then particles will communicate to each other and remembering the gbest value. Next each particle, by considering pbest and gbest values, will determine its new position and velocity in order to reach optimal solution.

On this paper, we will use Naïve Bayes (NB) method as the classifier. We implement NB method along with a feature selection method since the dataset that we use have more than 40 features. The feature selection method that we choose is Particle Swarm Optimization (PSO) that is one of the heuristics optimization technique as in Genetic Algorithm (GA) but on PSO there is no mutation computation, so it is technically simpler than GA. PSO has another advantage that is have a fast and stable optimization convergence. The data set that we use on this paper is KDD Cup ’99 data set that consist of more than 40 features and more than 400 thousands records [12]. From the raw KDD Cup ’99 data set, we first implement a feature selection method and the classify the data as several type of attacks or normal behavior.
2. Methods

As we explained before one of the approach to solve IDS problem is by Data Mining technique, especially classification method. It aims to classify data into several classes. There are several classification method, Support Vector Machine (SVM), Neural Network (NN), Fuzzy based-classification method as in Fuzzy C-Means method, Naive Bayes classifier, etc. Some research involving classification or decision making that based on fuzzy method as in [13] where Fuzzy Kernel K-Medoids was used to classify IDS problem data, Fuzzy Kernel C-Means was used for the same problem [14], and on [15] Fuzzy Logic method was used to make a decision related with stock exchange problem. On this research we use Naive Bayes Classifier that has a good accuracy and fast to reach convergence state when applied to large database [16].

Naive Bayes Classifier (NBC) is a classification method that based on Bayes Theorem, with the details of the method was given below. Flowchart of NBC can be seen on Figure 1.

1. If $D$ is training data consist of n-tuple data and connected with class label. Each tuple was represented by an n-dimensional vector, $X = (x_1, x_2, ..., x_n)$. It means that n different measurement was held on the tuple for n features, $A_1, A_2, ..., A_n$, respectively.
2. Set m classes, $C_1, C_2, ..., C_m$. Given a tuple $X$, classifier is going to predict that $X$ belongs to a class with highest posterior probability, conditioned on $X$. It means that NBC will predict that a tuple $X$ belongs to $C_i$ if and only if:

$$P(C_i \mid X) > P(C_j \mid X)$$

(1)

For $1 \leq i, j \leq m : j \neq i$.

So we are going to maximized $P(C_i \mid X)$. The class $C_i$ where $P(C_i \mid X)$ was maximized is called maximum posteriori hypothesis. By Bayes Theorem:

$$P(C_i \mid X) = \frac{p(X,C_i)p(C_i)}{p(X)}$$

(2)

Where $P(C_i \mid X)$ is a probability of $C_i$ given $X$, $P(X \mid C_i)$ is a probability of $X$ given $C_i$, $P(C_i)$ is a prior probability of $C_i$ and $P(X)$ represents probability of $X$.

3. When $P(X)$ is constant for each classes, we only have to maximize $P(X \mid C_i)P(C_i)$. If a prior probability of a class is unknown, generally we assume that $P(C_1) = P(C_2) = ... = P(C_m)$, so we only maximize $P(X \mid C_i)$. A prior probability of a class can be estimated by $P(C_i) = \frac{|C_{i,D}|}{|D|}$, where $|C_{i,D}|$ is the number of training data form class $C_i$ on $D$, and $|D|$ is the number of training data.

4. When we have a high dimensional dataset, it is quite expensive to compute $P(X \mid C_i)$. In order to reduce computational cost, we build a naive assumption of class conditional independence to have:

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i)P(x_2 \mid C_i) ... P(x_n \mid C_i)$$

(3)

where $x_k$ represents the value of attribute of $A_k$ for a tuple $X$. For each attribute, we have to determine whether it is on continuous form or categorical form. If $A_k$ is categorical form, then $P(x_k \mid C_i)$ is the number of class $C_i$ tuple on $D$ with $x_k$ as the value on attribute $A_k$, divided by $|C_{i,D}|$. If $A_k$ is on continuous form, we assume the distribution is Gaussian with mean $\mu$ and standard deviation $\sigma$, defined by:

$$P(x_k \mid C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_k-\mu)^2}{2\sigma^2}}$$

(4)

\[\text{Flowchart of NBC can be seen on Figure 1.}\]
Figure 1. Naïve Bayes Classifier Flow Chart
5. To predict the class label of $X$, $P(X \setminus C_i)P(C_i)$ was evaluated for every class $C_i$. Classifier predict that the label of $X$ is class $C_i$ if and only if

$$P(X \setminus C_i)P(C_i) > P(X \setminus C_j)P(C_j)$$  \hspace{1cm} (5)

For $1 \leq i, j \leq m : j \neq i$.

If the estimated probability value is zero then it will dominate classifier when it classify a new tuple. To prevent it, we may apply Laplacian correction.

Based on Chen and Shih on [11], to start Particle Swarm Optimization (PSO), initial velocity and position were determined randomly. The next steps are given below:

1. Population initialization (swarm)
   Population on PSO method is a set of particle with size N. Each particle has two different criteria that can be measured, position and velocity. Initial velocity and position were determined randomly and then set iteration equal to 1.

2. Fitness value evaluation
   Fitness value was used to measure particle distance in order to find best solution. Fitness value of each particle computed by the given objective function. If the fitness value for each particle on current position is better than pbest, then pbest was updated. And then we compare the particle fitness value with gbest. If gbest has a better value then we update it.

3. Updating Velocity and Position of ith- particle
   Equation (6) and (7) were used to update velocity and position of each particle.

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id} - x_{id}^t) + c_2 r_2 (p_{gd} - x_{id}^t)$$  \hspace{1cm} (6)

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$$  \hspace{1cm} (7)

where $\omega$ represents inertia weight, $v$ represents particle velocity, $x$ represents particle position, $c_1, c_2$ represents cognitive and social parameters of swarm, $r_1, r_2$ represents distributed random parameters $\in [0,1]$, $i$ as the particle itself, iteration $t$, and $p$ represents pbest, while $p_q$ represents gbest.

4. Termination
   If the algorithm has reached the stopping criteria then it stops and the optimum solution was reached or it continue to the next iteration and start from step 2. Based on [17], there are several conditions that can be used as a stopping criteria. We can use a maximum number of iteration, stop after the solution was found, or stop after there is no significant difference after several iterations.

On Figure 2 we can see flowchart of this research where Particle Swarm Optimization use as the Selection Feature Method and Naïve Bayes Classification as the classifier. For the experiment data, we use KDD Cup '99 dataset [12] that is a set of data used on The Third International Knowledge Discovery and Data Mining Tools Competition that held together with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. The competition ask the contestant to build a detector on network intrusion, a predictive model that was able to differ bad connection called attack or intrusion and normal behavior. There are four classes of attacks on these dataset:

1. Denial-of-Service (DoS): Intruder tries to prevent valid user to use the service. Some examples of this attack is pod, land, smurf, and neptune.
Figure 2. Research Flow Chart

2. Probe: Intruder tries to have information about the network, such as its weakness, confidential data or other classified information. Port scanning is one of the common examples of this type of attack. Other examples are saint, nmap, and ipsweep.
3. User-to-Root (U2R): The intruder had local access to the network and try to get privilege as a super user. Some attack type on this category such as buffer_overflow, rootkit, pearl and landmodule.

4. Remote-to-Local (R2L): The intruders do not have any access to the network but try to have access on it, as in multihop, ftp_write, phf, spy, and imap.

On Table 1, we can see 15 random examples of KDD Cup '99 data set with its 5 features and its label.

| Data | Features | Class |
|------|----------|-------|
| 1    | tcp      | http  | SF    | 202   | 6895  | normal |
| 2    | tcp      | http  | SF    | 361   | 7396  | normal |
| 3    | tcp      | http  | SF    | 291   | 236   | normal |
| 4    | tcp      | telnet| SF    | 260   | 2635  | attack |
| 5    | tcp      | telnet| SF    | 86    | 183   | attack |
| 6    | tcp      | ftp_data| SF  | 0     | 5921  | attack |
| 7    | icmp     | ecr_i | SF    | 1032  | 0     | attack |
| 8    | tcp      | private | S0  | 0     | 0     | attack |
| 9    | tcp      | http  | SF    | 54540 | 8314  | attack |
| 10   | tcp      | ftp_data| SF  | 0     | 467968| attack |
| 11   | tcp      | ftp   | SF    | 950   | 2551  | attack |
| 12   | tcp      | telnet| SF    | 112   | 847   | attack |
| 13   | tcp      | telnet| SH    | 0     | 0     | attack |
| 14   | tcp      | Other | REJ   | 0     | 0     | attack |
| 15   | tcp      | private | REJ  | 0     | 0     | attack |

3. Results and Discussion
Initially we use classification process by NBC without any feature selection method to get its accuracy. The value later will be used as fitness value on feature selection process by PSO. After we have the optimum feature set, classification stage on testing data was performed to measure the classifier performance. On this research, the classifier performance was measured by its accuracy. We use 90% data as the training data and 10% as the testing data.

To determine the PSO parameters value, $c_1$, $c_2$ and $\omega$, we performed some preliminary experiment. First, we choose $\omega = 0.1$, $N = 5$, and we try several possibilities of $c_1$, $c_2$. We only use 5 features and iteration number 50. For simplicity, we choose $c_1 = c_2$. The results were given on Table 2.

| $c_1$ | $c_2$ | Accuracy (%) |
|-------|-------|--------------|
| 0.1   | 0.1   | 79.52        |
| 0.2   | 0.2   | 79.77        |
| 0.3   | 0.3   | 19.68        |
| 0.4   | 0.4   | 79.47        |
| 0.5   | 0.5   | 79.04        |
| 0.6   | 0.6   | 83.94        |
| 0.7   | 0.7   | 80.36        |
| 0.8   | 0.8   | 19.68        |
| 0.9   | 0.9   | 19.67        |
| 1.0   | 1.0   | 19.68        |
Based on the preliminary results, we choose \( c_1 = c_2 = 0.6 \). With the same reason, we perform another preliminary experiment, where we use \( c_1 = c_2 = 0.6 \), and try to find the best value for \( \omega \). The best result was achieved when \( \omega = 0.1 \), with accuracy 83.94\%. So the parameters that we use on this research are \( c_1 = 0.6, \ c_2 = 0.6, \omega = 0.1, N = 5 \). Next we perform PSO and NBC to classify KDD '99 data set with the parameters obtained by previous preliminary experiments. The results were given on Table 3.

**Table 3. Classifier Performance by Number of Features Used**

| Number of Features | Accuracy (%) | Running Time (minutes) |
|--------------------|--------------|------------------------|
| 6                  | 77.00        | 61.84                  |
| 7                  | 76.58        | 76.00                  |
| 8                  | 94.09        | 83.03                  |
| 9                  | 94.02        | 94.64                  |
| 10                 | 93.97        | 107.11                 |
| 15                 | 94.35        | 166.10                 |
| 20                 | 88.01        | 222.37                 |
| 25                 | 98.20        | 236.83                 |
| 30                 | 98.58        | 322.67                 |
| 31                 | 98.58        | 325.30                 |
| 32                 | 98.58        | 353.21                 |
| 33                 | 98.65        | 320.94                 |
| 34                 | 98.59        | 338.21                 |
| 35                 | 98.63        | 351.61                 |
| 36                 | 98.92        | 354.59                 |
| 37                 | 98.95        | 357.39                 |
| 38                 | 99.12        | 360.27                 |
| 39                 | 98.90        | 368.45                 |
| 40                 | 98.85        | 368.96                 |
| 41                 | 98.93        | 383.09                 |

We can see on Table 3, there are some improvement while using only 38 features, the running time was shorter and the accuracy reach the highest point 99.12\%. Even for 15 selected features, the accuracy has already reached 94.35\% with much shorter running time, almost a third of running time by using the whole 41 features.

### 4. Conclusions

On this paper, we implemented Particle Swarm Optimization (PSO) as feature selection method and Naïve Bayes Method as classifier to classify one of the benchmark data set on IDS problem, KDD Cup’99 data set. The best performance was reached when we use 38 features with accuracy reaches 99.12\%.

To determine several PSO parameters value, we perform some simple preliminary experiment. For future works, it is possible to use parameter optimization method to find the best parameter value for PSO.
Reference

[1] Chaudari R R and Patil S P 2017 Int. Res. J. of Eng. and Tech. 4 1860-66
[2] Hemant P, Sarkhedi B and Vaghamshi H 2013 Int. J. of Adv. Res. in Elect. Elect. and Inst. Eng. 2 3063-70
[3] Kumar K V and Veerendranath B 2014 Int. J. of Adv. Res. in Comp. Sci. & Tech. 2 66-9
[4] Kabir R, Onik A R and Samad T 2017 Int. J. of Comp. App. 166 13-7
[5] Kumar S and Yadav A 2014 Proc. 2014 IEEE Int. Conf. on Adv. Comm. Cont. and Comp. Tech. 1935-39
[6] Yazhini S P and Devipriya R 2016 Int. J. of Sci. & Eng. Res. 7 37-42
[7] Varuna S and Ramya R 2017 Int. J. of Adv. Res. in Comp. and Comm. Eng 6 164-8
[8] Shrivastava N and Richariya V 2012 Int. J. of Comp. Eng. and Man. 15 (1) 54-63
[9] Aghdam M H and Kabiri P 2016 Int. J. of Net. Sec. 18 420-31
[10] Jahromy B M, Honarvar A R, Saif M and Jahromy M A M 2016 J. of Eng. and App. Sci. 4 810-5
[11] Chen R M and Shih H F 2013 Algo. 6 (2) 227-44
[12] Hettich S and Bay S D 1999 The UCI KDD Archive http://kdd.ics.uci.edu. Irvine, CA : University of California, Department of Information and Computer Science
[13] Rustam Z and Talita A S 2017 AIP Conf. Proc. 1862 (1) 030154
[14] Rustam Z and Talita A S 2015 J. of Theo. and App. Inf. Tech. 81 (1) 161-5
[15] Rustam Z, Talita A S and Pertiwi L A M 2019 J. of Theo. and App. Inf. Tech Vol. 97 (14) 3937-46
[16] Han J, Kamber M and Pei J 2012 Data Mining Concepts and Techniques 3rd ed. (USA: Elsevier Inc)
[17] Engelbrecht A P 2007 Computational Intelligence 2nd ed. (England: John & Sons Ltd)