Research on Intrusion Detection Algorithm of JRNB Network Based on Feature Weighting

Kunfu Wang, Shijun Xu and Pengyi Zhang
System Engineering Research Institute of China State Shipbuilding Corporation, Beijing, China.
Email: wangkunfu1166@163.com

Abstract. With the development of the Internet Era, the network attacks are on the rise. To some extent, the conditional independence assumption of Naive Bayes (NB) algorithm sacrifices the accuracy of classification, especially in dealing with complex network intrusion data. Aiming to solve this problem, this paper proposes a feature weighted JRNB intrusion detection algorithm. First, in order to remed for the deficiency of the equal analysis of all feature terms in Naive Bayesian algorithm, JS divergence method is introduced to measure the weight of each feature term to highlight the difference between different feature terms; Then, take into consideration of the impact of class frequency on sample classification, the reverse class frequency (RCF) is proposed to improve the calculation of feature weight and further reducing the impact of conditional independence. Compared with the traditional Naive Bayes algorithm and other popular classification algorithms, this algorithm in this paper has some improvement in detection performance.

1. Introduction
At present, the application of data mining algorithm of intrusion detection in big data environment has become a research hotspot\(^1\), the common algorithm is decision tree algorithm C4.5\(^2\), Extreme Learning Machine (ELM)\(^3\), artificial neural network (ANN)\(^4\), Naive Bayesian (NB)\(^5\) etc. The naive Bayes classification model is based on Bayes theorem and assumes that the feature conditions are independent, that is each feature used for classification is conditionally independent when the category is determined. This hypothesis makes the naive Bayesian classification model easy to implement, short training and learning time, and stable prediction efficiency, but at the same time, it also brings limitations\(^6\), which makes the dependency between attributes unable to be reflected, which is not consistent with the complex relationship of data in the Internet, so to some extent, it sacrifices the accuracy of classification, showing the network line For the disadvantage of poor prediction ability. So many scholars improve the algorithm in various ways to improve the classification effect.

Chen ZG \(^7\) proposed an intrusion detection algorithm using decision tree to expand naive Bayes, and combined with feature simplification technology to obtain a maximum posterior probability, improving the accuracy of classification. However, the algorithm will produce a large number of pseudo data in the process of data verification, which has the disadvantage of interference to classification. Koc I et al. \(^8\) used hidden naive Bayesian (HNB) algorithm in intrusion detection to add a hidden parent node to each attribute node on the basis of naive Bayesian, so as to reflect the correlation between attributes, reduce the impact of independence assumption between attributes, and improve the classification accuracy. However, when the network has multiple attributes, the efficiency of this algorithm is low. Dong Liyan et al \(^9\) proposed a semi supervised naive Bayesian (SNB) algorithm, which uses confidence to select a subset of unlabeled training set, and then combines the labeled samples,
iterating until the training is completed. The experimental results show that compared with the traditional NB algorithm, the accuracy of prediction is significantly improved, but the algorithm still needs to get the confidence list when the amount of data is small, which increases the time-consuming of the algorithm. Wang Shuangcheng et al.[10] proposed a dynamic and completely naive Bayesian classification algorithm, which uses multivariate Gaussian kernel function to calculate the conditional joint density of attributes, and uses smooth parameter optimization method to effectively combine the dependency between attributes, but in the big data environment, the efficiency of the algorithm is low.

Based on the previous research, this paper proposes an intrusion detection method using JS divergence and anti class frequency to improve naive Bayes algorithm. In the traditional NB algorithm, the weight factor is introduced to calculate the weight of each feature. The weight factor is obtained by JS divergence and anti category frequency calculation, which improves the detection rate and accuracy, and reduces the false detection rate.

### 2. JRNB Classification Model and Related Concepts

#### 2.1 Naive Bayes Classifier

Given a training sample set \( U = \{X_1, X_2, \ldots, X_n\} \), where elements \( X_i = \{a_1, a_2, \ldots, a_m | a_i \in A_i\} \) represent each data record, \( a_i \) represents the ith feature of this data record, \( A_i \) represents the ith attribute variable of the sample set, which can be a discrete value or a continuous value. There is a class set \( C = \{c_1, c_2, \ldots, c_k | k \leq n\} \), and the mapping function \( f : X_i \rightarrow c_i \) indicates that any data record is classified as a class label \( c_i \) in the set \( C \). Now consider a test sample set \( T = \{Y_1, Y_2, \ldots, Y_r\} \) to be classified, calculate the probability that test case \( Y_i \) belongs to class \( c_j (\forall j = 1, 2, \ldots, k) \), get the result set \( P = \{p_1, p_2, \ldots, p_k\} \), further get the largest element \( p_t \) in set \( P \), and finally classify the test case as \( c_t \).

The calculation steps are as follows:

1. Calculate the probability of category \( c \) in sample set \( U \):

\[
P(c) = \frac{\sum_{i=1}^{n} g(c, c_i)}{n}
\]

As such, \( g(c, c_i) = \begin{cases} 1 & c = c_i \\ 0 & c \neq c_i \end{cases} \), \( c_i \) is the category of sample \( X_i \), \( n \) is the number of elements in \( U \).

2. Calculate the conditional probability of the occurrence of characteristic \( a_j \) when the sample category in set \( U \) is \( c \), if attribute \( A_j \) is discrete value:

\[
P(a_j | c) = \frac{\sum_{i=1}^{n} g(c, c_i) g(a_i, a_j)}{\sum_{i=1}^{n} g(c, c_i)}
\]

As such, \( g(a_i, a_j) = \begin{cases} 1 & a_i = a_j \\ 0 & a_i \neq a_j \end{cases} \), \( a_j \) is the jth characteristic of training sample instance \( X_i \). If attribute \( A_j \) is a continuous value:

\[
P(a_j | c) = \frac{1}{\sqrt{2\pi \sigma_{c,j}}} \exp\left( -\frac{(a_j - u_{c,j})^2}{2\sigma_{c,j}^2} \right)
\]

3. Calculate the probability of feature \( a_j \) in sample set \( U \):
4. Calculate the probability that $y'$ belongs to category $c_i$:

$$P(c_i | y') = \frac{P(c_i | y') P(y')}{P(y')} = \frac{P(c_i) P(a_1 | c_i) P(a_2 | c_i) \cdots P(a_n | c_i)}{P(y')}$$

(5)

Where $y'$ is the sample instance to be tested and $m$ is the number of sample attributes.

5. From formula (5), we can calculate the probability that the sample $y'$ to be tested belongs to $\forall c_i (1 \leq i \leq k)$, and obtain $p = \{p_1, p_2, \cdots, p_k\}$. By normalizing $k$ probability values, we can get the similarity of samples $y'$ belonging to each category, and get the maximum posterior probability $MP$:

$$MP = \arg \max_{c_i} P(c_i) \prod_{j=1}^{m} P(a_j | c_i)$$

(6)

6. From the above results, the definition of naive Bayesian classifier can be given:

$$Classifier(T = \{Y_1, Y_2, \cdots, Y_k\}) = \arg \max_{c_i} P(c_i) \prod_{j=1}^{m} P(a_j | c_i)$$

(7)

2.2 Feature Weighting

The traditional NB algorithm assumes that the attributes are independent of each other and all features are analyzed equally, but in reality, each feature plays a different role in the process of sample classification. This will inevitably reduce the accuracy of classification [11], so this paper uses a feature weighting algorithm to give the corresponding weights of different feature items according to the situation, so as to improve the accuracy of classification.

**Definition 1** 

$weight(i, j)$ is the weight of feature $a_j$ in category $c_i$, which is to measure the importance of feature $a_j$ to category $c_i$ in the process of classification.

In this way, the naive Bayes formula is improved:

$$MP = \arg \max_{c_i \in \{c_1, c_2, \cdots, c_k\}} P(c_i) \prod_{j=1}^{m} P(a_j | c_i) \times weight(i, j)$$

(8)

(1) KL Divergence

To further quantify the importance of feature $a_j$ to class $c_i$, we can consider the difference between the probability distribution of $c_i$ in the sample set and that in the sample set with feature $a_j$. The greater the difference is, the more important feature $a_j$ is to category $c_i$; otherwise, the less representative it is to category $c_i$. KL divergence can be used to describe the difference between the two probability distributions, and then the calculated difference can be used to represent the importance of features in the classification process. The calculation formula is as follows:

$$KL(P(c_i | a_j) \| P(c_i)) = P(c_i | a_j) \log \frac{P(c_i | a_j)}{P(c_i)}$$

(9)

(2) JS Divergence

From KL divergence formula (9), we can see its limitations. It is used to express the distance difference between two probability distributions, but it does not have symmetry, so it cannot be regarded as a real measure. Secondly, there is no limit to the result range. Therefore, this paper introduces JS divergence method to improve it to make up for the above shortcomings. JS divergence has symmetry and is a real distance measure. In addition, its value range is from 0 to 1, so it is more ac-
curate and convenient to judge the similarity. Therefore, JS divergence can be introduced into naive Bayes classification algorithm, and it can be used to compare the distance difference between two probability cases to give the corresponding weight of feature terms. The calculation formula is as follows:

\[
JS[p(c_i | a_j), p(c_i)] = \frac{1}{2} KL[p(c_i | a_j), \frac{P(c_i | a_j) + P(c_i)}{2}] + \frac{1}{2} KL[p(c_i), \frac{P(c_i | a_j) + P(c_i)}{2}]
\]

(10)

**Definition 2** \(W_{JS}(i, j)\) is the JS weight factor of feature \(a_j\) in category \(c_i\).

The calculation method of \(W_{JS}(i, j)\) can be obtained by substituting formula (9) into formula (10). It can be seen from formula (11) that the more dispersed feature \(a_j\) is in the sample set, the smaller JS weight factor is for category \(c_i\). If feature distribution is more concentrated in category \(c_i\), the larger JS weight factor is.

\[
W_{JS}(i, j) = \frac{1}{2} P(c_i | a_j) \log \frac{2P(c_i | a_j)}{P(c_i) + P(c_i | a_j)} + \frac{1}{2} \frac{P(c_i) \log \frac{2P(c_i)}{P(c_i) + P(c_i | a_j)}}{P(c_i | a_j)}
\]

(11)

(3) RC Frequency

The feature weighting method with JS divergence focuses on the proportion of samples with feature items in different categories, which reflects the aggregation degree between categories, without considering the impact of class frequency on sample classification. Literature shows that the proportion between the number of categories with characteristic items and the total number of categories is also a basis for judging the importance of characteristic items. Because the characteristic items that are representative of a particular class appear in a few classes, the reverse class frequency (RCF) can be used to further improve it.

**Definition 3** \(W_{RCF}(j)\) is the RCF weight factor of the feature \(a_j\). The calculation formula is as follows:

\[
W_{RCF}(j) = \log \frac{2N_c}{\sum_{i=1}^{N_c} g(a_j, c_i)}
\]

(12)

Where \(N_c\) is the total number of categories, \(g(a_j, c_i) = \begin{cases} 1 & a_j \notin c_i \\ 2 & a_j \in c_i \end{cases} \).

3. JRNB Classification Model

3.1 Algorithm Description

The characteristic weight \(\text{weight}\) is calculated by combining the above weight factors \(W_{JS}\) and \(W_{RCF}\):

\[
\text{weight}(i, j) = W_{JS}(i, j) \times W_{RCF}(j)
\]

(13)

Based on the PCA construction of attribute set and the above feature weighting method, this paper proposes jmb classification algorithm, the specific implementation is as follows:

**Input:** Training sample set \(U\).

Sample instance to be classified \(Y = \{a_1, a_2, \ldots, a_m\}\).

Category label \(C = \{c_1, c_2, \ldots, c_k\}\).
Output: Category of sample $y$: $c_y$
1) $k$ = Category quantity
// Obtain prior probability $P(c_i)$
2) for each $i$ in $k$
3) $n = 0$
4) $s = 0$
5) for each $X$ in $U$
6) $n = n + 1$
7) if ($X \in c_i$)
8) $s = s + 1$
9) end for
10) $P(c_i) = s/n$
11) end for
// Record the probability that sample $y$ belongs to each category
12) $P = \{p_1, p_2, \ldots, p_k\}$
13) for each $i$ in $k$
14) $P(Y|c_i) = 1$
15) for each $j$ in $m$
16) weight(i,j) = $W_{JS}(i,j) \times W_{RCF}(j)$
17) $P(Y|c_i) = P(Y|c_i) \times P(a_j|c_i) \times weight(i,j)$
18) end for
19) $p_i = P(c_i) \times P(Y|c_i)$
20) end for
21) $c_y = (c_i | p_i = \max \{p_1, p_2, \ldots, p_k\})$
22) output($c_y$)

3.2 JRNB Intrusion Detection Model
The JRNB algorithm is used in the intrusion detection model. Through a series of process processing for the corresponding network traffic data, the classification results of network events are obtained. The flow of the intrusion detection model is shown in Figure 1.
The intrusion detection data set adopts NSL-KDD, which is an improvement of KDD99 and solves the inherent problems in KDD99. As a standard data set in the current intrusion detection field, it has been widely used. The data set has 41 features and the attack types are divided into four categories. In the experiment, the following indexes are usually used to evaluate the performance comparison:

\[
\text{Check Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Total number of test samples}}
\]

\[
\text{Detection Rate} = \frac{\text{Number of intrusion samples detected}}{\text{Total number of intrusion samples}}
\]

\[
\text{Noise Factor} = \frac{\text{Number of normal samples detected as intrusion by mistake}}{\text{Number of normal samples}}
\]

In intrusion detection, the harm caused by false positives is more serious than false positives, so people pay more attention to the requirement of detection rate.

4. Conclusion
In the face of a large number of complex network attacks in today's network environment, this paper uses naive Bayes related theoretical knowledge, and proposes an improved JRNB algorithm, which reduces the limitations of the conditional independence hypothesis of naive Bayes by introducing weight factors for each feature term through JS divergence and anti category frequency. Compared with the traditional naive Bayes algorithm and other popular classification algorithms, the network intrusion detection algorithm based on JRNB proposed in this paper has some improvement in detection performance. The next step is to study how to preprocess the data more effectively to improve the stability and adaptability of JRNB algorithm.

5. Reference
[1] Jian Liu, Purui Su, Min Yang, et al. Software and Cyber Security-A Survey[J], Journal of Software, 2018, 29(1):42-68.
[2] Zhang Yan, Cao Jian. Decision Tree Algorithms for Big Data Analysis [J]. Computer Science, 2016, 43(6A): 374.
[3] ZHOU Bang-bang, WEI Shu-ning, TANG Yong, et al. ELM network intrusion detection algorithm based on rough set attribute reduction [J]. Transducer and Microsystem Technologies 2019,38(01):128-131.
[4] Mohammadi S, Amiri F. An Efficient Hybrid Self-Learning Intrusion Detection System Based on Neural Net-works[J]. International Journal of Computational Intelligence and Applications, 2019, 18(01): 1950001.
[5] Gumus F, Sakar C O, Erdem Z, et al. Online Naive Bayes classification for network intrusion detection[C]//2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014). IEEE, 2014: 670-674.
[6] LIU Hong-li LI Lu HU Jiu-song WSNs intrusion detection algorithm based on CS-CPSO and SVM fusion [J]. Transducer and Microsystem Technologies, 2017,36(9):110-112.
[7] Chen Z G, Kim S R. Combining principal component analysis, decision tree and naïve Bayesian algorithm for adaptive intrusion detection[C]//Proceedings of the 2013 Research in Adaptive and Convergent Systems. ACM, 2013: 312-316.
[8] Koc L, Carswell A D. Network intrusion detection using a hnb binary classifier[C]//2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim). IEEE, 2015: 81-85.
[9] DONG Li-yan1, SUI Peng, SUN Peng, LI Yong-li, et al. Novel naïve Bayes classification algorithm based on semi-supervised learning [J].Journal of Jilin University Engineering and Technology Edition, 2016, 46(3): 884-889
[10] WANG Shuang-cheng, DU Rui-jie, LIU Ying et al. The Learning and Optimization of Full Bayes Classifiers with Continuous Attributes [J]. Chinese Journal of Computers, 2012, 35(10):2129-2138.

[11] Sarker I H, Kabir M A, Colman A, et al. An Improved Naive Bayes Classifier-Based Noise Detection Technique for Classifying User Phone Call Behavior[J]. 2017.