Hindawi
Security and Communication Networks
Volume 2024, Article ID 7065482, 13 pages
https://doi.org/10.1155/2024/7065482

Research Article

Feature-Weighted Naive Bayesian Classifier for Wireless Network Intrusion Detection

Hongjiao Wu

Henan Vocational College of Tuina, Luoyang 471000, China

Correspondence should be addressed to Hongjiao Wu; whjrpf@163.com

Received 23 December 2022; Revised 26 September 2023; Accepted 4 November 2023; Published 3 January 2024

Academic Editor: Helena Rifà-Pous

Objective. Wireless sensor networks, crucial for various applications, face growing security challenges due to the escalating complexity and diversity of attack behaviours. This paper presents an advanced intrusion detection algorithm, leveraging feature-weighted Naive Bayes (NB), to enhance network attack detection accuracy. *Methodology*. Initially, a feature weighting algorithm is introduced to assign context-based weights to different feature terms. Subsequently, the NB algorithm is enhanced by incorporating Jensen–Shannon (JS) divergence, feature weighting, and inverse category frequency (ICF). Eventually, the improved NB algorithm is integrated into the intrusion detection model, and network event classification results are derived through a series of data processing steps applied to corresponding network traffic data. *Results*. The effectiveness of the proposed intrusion detection algorithm is evaluated through a comprehensive comparative analysis using the NSL-KDD dataset. Results demonstrate a significant enhancement in the detection accuracy of various attack types, including normal, denial of service (DoS), probe, remote-to-local (R2L), and user-to-root (U2R). Moreover, the proposed algorithm exhibits a lower false alarm rate compared to other algorithms. *Conclusion*. This paper introduces a wireless network intrusion algorithm that not only ensures improved detection accuracy and rate but also reduces the incidence of false detections. Addressing the evolving threat landscape faced by wireless sensor networks, this contribution represents a valuable advancement in intrusion detection technology.

1. Introduction

In recent years, there has been a rapid development in computer communication and networking technologies, particularly with the emergence of the Internet of Things (IoT). These advancements have introduced new and effective ways to facilitate interaction between human society and the physical world. This has led to the integration of human society, the physical world, and the computing world [1, 2]. Wireless sensor networks (WSNs), as a technology derived from microelectronics, play an irreplaceable role in important fields such as healthcare, traffic control, and natural disasters [3–5]. Furthermore, they have profound impacts on daily life, such as smart homes and modern agriculture [6, 7].

WSN is a distributed wireless network composed of a large number of low-power sensor nodes deployed in the sensing area, communicating through wireless links [8]. These sensor nodes are miniaturized computing units with limited storage capacity, computational capabilities, and battery power [9, 10]. However, due to the openness of wireless networks and the inherent limitations of sensor nodes, WSNs face various security threats and network attacks.

To counter network attacks, existing network intrusion solutions have introduced key management and authentication mechanisms as the first line of defence to effectively withstand attacks from outside the WSN. However, by capturing nodes, attackers can gain access to confidential information inside the nodes and launch internal attacks, rendering the first line of defence ineffective against internal attacks in WSNs. Therefore, intrusion detection technology serves as a crucial second line of defence, capable of fundamentally detecting security threats and minimizing the losses caused by attacks [11]. However, due to the limitations of WSNs, traditional intrusion detection techniques cannot

be directly applied to WSN environments. As a result, the research on novel intrusion detection techniques suitable for WSNs has attracted widespread attention from experts and scholars both domestically and internationally.

In order to improve the classification accuracy and algorithmic efficiency of network intrusion detection algorithms, this paper proposes a new intrusion detection method for wireless networks, that is, an improved NB algorithm utilizing JS dispersion and inverse category frequency (ICF), i.e., JINB intrusion detection algorithm. The algorithm reduces the limitations of the conditional independence assumption of NB by introducing a weighting factor for each feature term through JS scattering and ICF, resulting in the improved detection rate and detection accuracy. The main innovations of this paper are as follows:

(1) The use of JS divergence to measure the weights of each feature term, highlighting the differences between different feature terms. By utilizing JS divergence, we are able to better assess the contributions of feature terms to intrusion detection, thereby improving classification accuracy.

(2) The introduction of ICF to enhance the calculation of feature weights, further reducing the influence of conditional independence. Traditional intrusion detection algorithms are typically based on the assumption of Naive Bayes, where features are considered mutually independent. However, in practical scenarios, there may be certain correlations among features that can impact the accuracy of classification. Incorporating ICF allows for more accurate calculation of feature weights, reducing the impact of conditional independence on classification results and enhancing detection accuracy.

This paper is divided into five main sections as follows: the introduction, the literature review, the methodology, the analysis and discussion of results, and the conclusion.

The motivation behind this study is to enhance the security of wireless sensor networks by developing an intrusion detection algorithm that improves detection accuracy and reduces false alarms. This research aims to contribute to the evolving field of intrusion detection in WSNs.

In addition to introducing innovative technologies such as JS divergence and ICF, this study also incorporates a feature weighting algorithm. This contribution enhances the algorithm's sensitivity to different features, aiding in better discrimination and utilization of various features, thereby improving the accuracy of intrusion detection. Furthermore, this study combines the improved Naive Bayes (NB) algorithm with an intrusion detection model and applies it to the corresponding network traffic data processing. This leads to an overall performance enhancement, resulting in higher detection accuracy when facing various attack types.

2. Literature Review

In the existing literature, various intrusion detection models and algorithms have been proposed for different network environments and challenges. Zhao et al. proposed an intrusion detection model based on a deep artificial neural network with backpropagation (DAN-BP) [12]. It is tailored for handling massive, complex, and multidimensional network data. The primary aim is to address the need for effective intrusion detection in such environments. Similarly, Maheswari and Arunesh present a new hybrid multilevel intrusion detection model that focuses on improving the detection rate of specific attack behaviours, including probe, U2R, and R2L [13]. The model combines the K-nearest neighbor (KNN) outlier detection algorithm with network traffic similarity to achieve accurate detection without interference from anomalous behaviours.

In the context of resource-constrained intrusion detection systems (IDS) in wireless sensor networks (WSNs), Huang and Zhu proposed a dynamic multistage intrusion detection model with a game-theoretic approach [14]. This model predicts the most vulnerable nodes to intrusion and incorporates Bayesian rule analysis to identify malicious nodes. Another improvement in intrusion detection is presented by Wang et al. [15], where an algorithm based on integration learning is introduced. This algorithm addresses the limitations of integrated learning intrusion detection methods, such as loss of edge information and time-consuming model fusion. By transforming the original problem into multiple binary classification problems and incorporating probabilistic prediction results, the algorithm achieves better performance.

While deep neural networks and integrated learning methods [16, 17] may not be suitable for certain network structures and computational limitations [18], alternative approaches have been explored. For example, Jaber and Rehman proposed a cloud computing-based intrusion feature extraction method for ship communication networks [19]. This method utilizes signal processing techniques and a feature detection framework to extract relevant features. A network intrusion detection method based on an improved random forest classifier is introduced by Zhang et al. [20]. The method utilizes Gaussian mixture models and random forest classifiers to extract intrusion features.

Ling et al. used a rough set theory to enhance intrusion detection models based on artificial immunity [21]. This method combines anomaly detection and misuse detection, achieving vaccine injection without terminating the intrusion detection behavior. In addition, Ling et al. introduced a *K*-means algorithm to address the clustering issue and local optima problems [22]. Although these methods demonstrate improvement in detection performance, challenges related to outliers and unbalanced clustering remain.

To tackle the issue of low efficiency caused by a large amount of data, Wang et al. combined adaptive affinity propagation (AP) clustering with intrusion detection [23]. This adaptive AP clustering algorithm reduces the number of samples to be clustered, resulting in decreased clustering time and continuous model adjustment. Duan and Xiao proposed an improved fuzzy *C*-means clustering algorithm, which uses the Mercer kernel and Lagrange multiplier method for enhanced optimization and convergence speed [24]. However, the impact of unbalanced clustering and noise points on clustering results is not addressed.

Moreover, recent research has explored innovative approaches for intrusion detection. For instance, a vehicular-edge computing fogging scheme is proposed by Mourad et al. [25], which offloads intrusion detection tasks to federated vehicles, considering their high mobility and resource availability. Rahman proposes a privacy-preserving joint learning scheme for IoT intrusion detection [26], where devices train their own models to maintain privacy and security-aware data. Deep learning models with recurrent neural network (RNN) with long short-term memory (LSTM) and gated recurrent unit (GRU) frameworks were developed by Gautam et al. [27]. This model can overcome the problem of longer dependencies in RNN models. However, the effectiveness of these models in classifying certain types of attacks remains a challenge.

The Naive Bayesian classifier (NBC) [28] has gained popularity in intrusion detection due to its simplicity and effectiveness. Various enhancements for the NB algorithm have been proposed. For instance, Alsaadi et al. effectively applied the NB model in IDS and established a framework for the primary intrusion detection process [29]. Alsharif adopted a PCA-based NB algorithm that reduces data redundancy and improves detection efficiency [30]. Zhang et al. constructed a network intrusion detection model based on the NB algorithm and the quantum particle swarm optimization (QPSO) algorithm [31], which performs feature selection and parallelized NB classification. However, the detection rate of low-frequency and small-sample type data still needs improvement in this method. Panigrahi et al. proposed an extension of the NB algorithm that combines feature simplification and decision tree techniques to improve classification efficiency and accuracy [32]. Nevertheless, this approach introduces interference in the classification of anomalous events. A semisupervised NB algorithm is introduced by Hara and Shiomoto [33], which leverages parallel computing to handle large amounts of network data. However, this algorithm exhibits shortcomings in detecting anomalous data in small-sized and medium-sized local area networks. Addressing the need for efficient intrusion detection, Li et al. presented the locally weighted Naive Bayesian (LWNB) algorithm [34]. This algorithm gradually reduces the feature space and divides it into subspaces for classification using a NB classifier. While this design improves classification speed and correctness, it requires complex preprocessing and increases time and space complexity. Jiang et al. proposed a novel weight calculation method based on the original NB approach [35]. This method estimates feature values derived from network data and significantly eliminates interference data, leading to improved classifier accuracy and recall rate. However, the algorithm still faces challenges related to feature attribute value extraction and the stability of classifier performance.

In summary, the literature review highlights various approaches to intrusion detection, including DAN-BP-based models, hybrid multilevel models, dynamic multistage models, integration learning algorithms, cloud computing-based methods, improved random forest classifiers, rough set-based models, and extensions of the NB algorithm. While these methods demonstrate advancements in intrusion detection, challenges such as outliers, unbalanced clustering, computational limitations, and classification of specific attack types still persist. Future research should focus on addressing these challenges to further enhance intrusion detection techniques.

3. Methodology

A feature weighting algorithm is employed to enhance the classification accuracy by assigning weights to different feature terms based on their relevance to the situation. This approach calculates each feature's weight by deriving a weighting factor through JS divergence and ICF.

The JINB algorithm is used to obtain the classification results of network events by performing a series of processes on the corresponding network traffic data (Figure 1).

3.1. NB Classifier. Given a training sample set $P = \{I_1, I_2, \cdots, I_t\}$, where element $I_x = \{g_1, g_2, \cdots, g_w \mid g_x \in G_x\}$ represents each data record and g_x denotes the x-th feature. G_x denotes the x-th attribute variable of the sample set. Consider a test sample set $C = \{c_1, c_2 \cdots, c_z \mid z \leqslant t\}$, and the mapping function f: $I_x \longrightarrow c_x$ indicates that any data record is classified as a category label c_x in C. Suppose $N = \{J_1, J_2, \cdots, J_r\}$ is a test sample set, then we calculate the probability that instance J_x belongs to category $c_y\,(\forall y = 1, 2, \cdots, z)$ and obtain the calculation result set $U = \{u_1, u_2, \cdots, u_z\}$. Then, the maximum element u_n in set U is further obtained, and the test instance is finally classified as c_n.

(1) We calculate the occurrence of category c in the sample set U as follows:

$$U(c) = \frac{1}{t}\sum_{x=1}^{t} a(c, c_x),$$
$$a(c, c_x) = \begin{cases} 1, c = c_x, \\ 0, c \neq c_x, \end{cases} \tag{1}$$

where c_x is the category to which the sample I_x belongs to and $a(c, c_x)$ is the symbolic function for determining c and c_x.

(2) We compute feature g_y for sample category c in the set P. If attribute G_y is discrete, then we have

$$U(g_y \mid c) = \frac{\sum_{x=1}^{t} a(c, c_x) a(g_{xy}, g_y)}{\sum_{x=1}^{t} a(c, c_x)},$$
$$a(g_{xy}, g_y) = \begin{cases} 1, g_{xy} = g_y, \\ 0, g_{xy} \neq g_y, \end{cases} \tag{2}$$

where g_{xy} is the y-th feature of the training sample instance I_x. If the attribute g_y is a continuous value, then we have

$$U(g_y \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,y}} \exp\left(-\frac{(g_y - p_{c,y})^2}{2\sigma_{c,y}^2}\right). \tag{3}$$

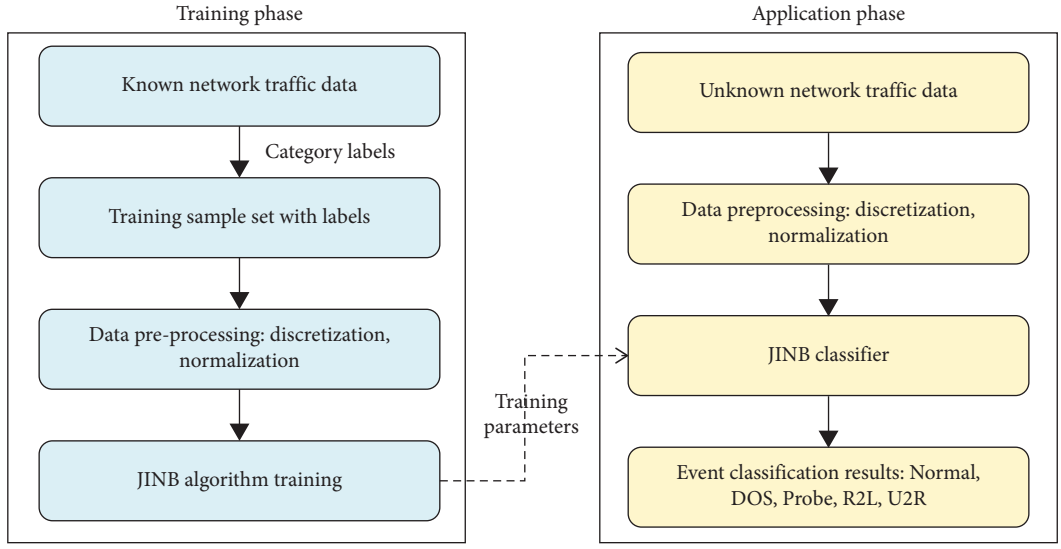

Figure 1: Intrusion detection model based on the JINB algorithm.

(3) Then, we count the occurrence of feature g_y in U as follows:

$$U(g_y) = \sum_{x=1}^{t} U(g_y \mid c_x) U(c_x). \tag{4}$$

(4) We calculate the probability that J' belongs to category c_x as

$$U(c_x \mid J') = \frac{U(c_x) U(J' \mid c_x)}{U(J')} = \frac{U(c_x)}{U(J')} U(g_1 \mid c_x) U(g_2 \mid c_x) \cdots U(g_w \mid c_x). \tag{5}$$

Among them, J' is the sample to be measured and w is the sample size.

(5) From equation (7), it is possible to calculate for sample J' within the range of $\forall c_x (1 \leqslant x \leqslant z)$, which results in a derived value. $U = \{u_1, u_2, \cdots, u_z\}$. The z probability values are normalized and sorted to obtain the similarity of the sample J'. The maximum posterior probability WU is thus obtained as follows:

$$\mathrm{WU} = \underset{1 \leqslant x \leqslant z}{\operatorname{argmax}} U(c_x) \prod_{y=1}^{w} U(g_y \mid c_x). \tag{6}$$

(6) The definition of the NB classifier can be calculated from the abovementioned results as

$$\mathrm{Classifier}(N = \{J_1, J_2, \cdots, J_r\}) = \underset{\forall \in J, 1 \leqslant x \leqslant z}{\operatorname{argmax}} U(c_x) \prod_{y=1}^{w} U(g_y \mid c_x). \tag{7}$$

3.2. JS Divergence. Weight (x, y) is the weight of g_y in c_x; that is, it measures the significance of the feature g_y to category c_x in classification. The NB equation is thus improved as shown in the following equation:

$$\mathrm{WU} = \underset{1 \leqslant x \leqslant z}{\operatorname{argmax}} U(c_x) \prod_{y=1}^{w} U(g_y \mid c_x) \times \mathrm{weight}(x, y). \tag{8}$$

The difference between the probability distribution of c_x and the probability distribution in the sample set with feature g_y can be considered. According to literature [36], the KL (Kullback–Leibler) divergence is used to indicate the importance of the features.

$$\mathrm{KL}\left[U(c_x \mid g_y) \| U(c_x)\right] = U(c_x \mid g_y) \log \frac{U(c_x \mid g_y)}{U(c_x)}. \tag{9}$$

The limitations are evident from the KL divergence calculation in equation (9), which cannot be regarded as a metric in the true sense. Second, the range of its results is not bounded. In this paper, JS divergence [37] is introduced to make up for deficiencies. Since JS divergence possesses symmetry, it serves as a true distance metric. Moreover, its values range from 0 to 1, making it more precise and convenient for similarity assessment. Therefore, employing JS divergence to compare the difference in distance between two probability scenarios assigns appropriate weights to

feature items, enabling a better evaluation of the contribution of feature items to intrusion detection.

$$\mathrm{JS}\left[U\left(c_x \mid g_y\right) \| U\left(c_x\right)\right] = \frac{1}{2}\mathrm{KL}\left[U\left(c_x \mid g_y\right) \middle\| \frac{U\left(c_x \mid g_y\right) + U\left(c_x\right)}{2}\right] + \frac{1}{2}\mathrm{KL}\left[U\left(c_x\right) \middle\| \frac{U\left(c_x \mid g_y\right) + U\left(c_x\right)}{2}\right]. \tag{10}$$

$W_{\mathrm{JS}}(x, y)$ is the JS weighting factor of g_y in category c_x.

The calculation of $W_{\mathrm{JS}}(x, y)$ can be derived by subsuming equation (9) into equation (10). From equation (11), it can be observed that if the distribution of feature g_y is more dispersed, the JS weighting factor for c_x becomes smaller.

$$\begin{cases} W_{\mathrm{JS}}(x, y) = \frac{1}{2}U\left(c_x \mid g_y\right)\log \frac{2U\left(c_x \mid g_y\right)}{U\left(c_x\right) + U\left(c_x \mid g_y\right)} +, \\ \frac{1}{2}U\left(c_x\right)\log \frac{2U\left(c_x\right)}{U\left(c_x\right) + U\left(c_x \mid g_y\right)}, \\ U\left(c_x \mid g_y\right) = \frac{U\left(c_x\right)U\left(g_y \mid c_x\right)}{U\left(g_y\right)}. \end{cases} \tag{11}$$

3.3. Anticategory Frequency. Since feature terms representative of a particular category occur in a small number of classes, it can be further improved by using ICF [38].

The inverse category frequency and feature entropy are introduced into the calculation of feature weights in sample classification. The category frequency (CF) is the number of categories in which feature g_y occurs.

The calculation equation of anticategory frequency ICF is similar to IDF, which can be represented as follows:

$$W_{\mathrm{ICF}}\left(g_y\right) = lh\left(\frac{|C|}{cf\left(g_y\right)}\right). \tag{12}$$

The introduction of ICF in feature weighting is based on the assumption that the fewer the number of categories in which a feature g_y appears, the greater the amount of category information it carries. This assumption is called the ICF assumption, which focuses on low- and medium-frequency features at the category level while suppressing high-frequency features. However, ICF only considers the distribution of features between categories and does not consider the distribution of features within each category. If a feature term exhibits a more balanced distribution within a class, it signifies greater representativeness of that class. This indicates a higher capability for class differentiation, warranting a larger weight assignment across all samples belonging to that class. Conversely, if a feature term is concentrated in only a few samples within a class, it does not effectively capture the characteristics of the class. Feature terms with low category differentiation ability should be assigned lower weights. The analysis shows that the size of the entropy value of the distribution of feature terms within a class is consistent with the amount of categorical information that the feature term can provide.

The term entropy of the feature g_y in the class c_x is defined as follows:

$$\begin{aligned} \mathrm{TE}\left(g_y, c_x\right) &= -\sum_{y=1}^{|c_x|} \frac{\mathrm{tf}\left(g_y, d_{\mathrm{xy}}\right)}{\mathrm{tf}\left(g_y, c_x\right)} \mathrm{lb} \frac{\mathrm{tf}\left(g_y, d_{\mathrm{xy}}\right)}{tf\left(g_y, c_x\right)}, \\ \mathrm{TF}\left(g_y, c_x\right) &= \sum_{y=1}^{|c_x|} \mathrm{tf}\left(g_y, d_{\mathrm{xy}}\right), \end{aligned} \tag{13}$$

where $\mathrm{TF}(g_y, c_x)$ represents the total number of frequencies of feature g_y occurring in the samples of class c_x.

$\mathrm{TE}\ (g_y, c_x)$ well reflects the distribution of feature terms within the class, and its value is proportional to the category differentiation ability of the feature. Based on the above-mentioned analysis, this section introduces the XCF and te factors into the feature weight calculation and proposes two new feature weight calculation schemes, namely, TF.XCF.TE and TF.RF.XCF.TE.

(a) The equation for the TF.XCF.TE program is as follows:

$$\mathrm{TF.XCF.TE}\left(g_y, d_y, c_x\right) = \mathrm{tf}\left(g_y, d_y\right) \times \mathrm{lb}\left(1 + \frac{|C|}{\mathrm{cf}\left(g_y\right)}\right) \times \mathrm{te}\left(g_y, c_x\right). \tag{14}$$

In contrast to TF.XDF, TF.XCF.TE is a hybrid feature weighting model. XCF factors are calculated at the category level, and TE factors measure the distribution of features within classes.

(b) The TF.RF.XCF.TE program is calculated as follows:

$$\mathrm{TF.RF.}XCF\mathrm{.TE}\left(g_y, d_y, c_x\right) = \mathrm{tf}\left(g_y, d_y\right) \times \mathrm{lb}\left(2 + \frac{g}{\max(l, c)}\right) \times \mathrm{lb}\left(1 + \frac{|C|}{\mathrm{cf}\left(g_y\right)}\right) \times \mathrm{te}\left(g_y, c_x\right), \tag{15}$$

where g is the number of samples in which feature g_y appears in the positive class. c is the number of samples in which feature g_y appears in the negative class. It can be seen that the TF.RF.XCF.TE scheme contains four factors. The TF is the original feature frequency. The RF factor measures the distribution of feature g_y between positive and negative correlation categories. The XCF factor measures the distribution of feature g_y between categories. The TE factor is a measure of the distribution of features within classes.

3.4. Algorithm Description. By combining the aforementioned weighting factors, W_{JS} and W_{ICF}, the feature weights, denoted as weight, can be calculated.

$$\mathrm{Weight}(x, y) = W_{\mathrm{JS}}(x, y) \times W_{\mathrm{ICF}}\left(g_y\right). \tag{16}$$

The steps of the JINB algorithm are as follows.

Referring to the process outlined in Algorithm 1, we can analyze the time and spatial complexity of the JINB algorithm proposed in this paper. The time complexity of the JINB algorithm can be approximated as $O(z+z*z+z*w+z*w)$, which can be simplified to $O(z*z+z*w)$. Regarding the space complexity, it refers to the additional storage space required during the execution of the algorithm, primarily for storing the training sample set U and variables used for calculations. Assuming that the sizes of other variables are negligible compared to z, the space complexity of the JINB algorithm can be approximated as $O(z)$.

4. Result Analysis and Discussion

4.1. Experimental Dataset. The NSL-KDD dataset is used for the experiments in this paper. The NSL-KDD dataset is a modified version of the KDD Cup 1999 dataset, which was created for the purpose of evaluating intrusion detection systems. It was developed to overcome some of the limitations and issues found in the original KDD Cup 1999 dataset, which had problems related to redundancy and repetitive data records. It provides a diverse set of network traffic data with various types of attacks, making it a valuable resource for researchers and practitioners in the field of cybersecurity. The distribution of NSL-KDD across different attack categories is illustrated in Table 1.

The experimental setup involved a Windows 10 PC equipped with an Intel Core i7-9750H CPU running at 2.60 GHz and 8 GB of RAM. The algorithms proposed in this paper were implemented using Python 3.7.3. The simulation experiments were conducted using the publicly available NSL-KDD dataset, which served as the dataset for this study. The experimental parameters of the algorithm in this paper are set as follows. The number z of categories C is set to 5, due to the 4 attack types (DoS, Probe, R2L, and U2R) and normal state. The number w of sample instances is shown in Table 1, and the number of sample J is 132427.

The steps for data preprocessing are as follows:

(1) Data collection: first, the NSL-KDD dataset was obtained for the experiments

(2) Data cleaning: the data were cleaned to check for missing values, duplicate records, or inconsistencies

(3) Data exploration: exploratory data analysis (EDA) techniques were employed to gain in-depth insights into the dataset's features and the distribution of attacks across different categories.

(4) Data partitioning: the dataset was divided into a training set and a testing set, with the partitioning carried out using the KDD Train +_20Percent.TXT and KDD Test +.TXT files, respectively.

4.2. Experimental Evaluation Methods. The intrusion detection system evaluation index is calculated by the confusion matrix, and its main evaluation index is divided into accuracy rate (Acc), detection rate (DR), false alarm rate (FAR), and missing alarm rate (MAR).

Acc is a measure of overall performance that takes into account all samples correctly classified, but it does not provide detailed information about the accuracy of intrusion detection. DR focuses on intrusion detection accuracy, which measures the extent to which the classifier correctly detects actual intrusions. FAR and MAR are associated with false positives and false negatives and typically exhibit a negative correlation. Reducing FAR may lead to an increase in MAR, and vice versa, representing a trade-off. When evaluating intrusion detection systems, it is common to strike a balance between FAR and MAR to reduce false positives while minimizing false negatives, thereby enhancing DR and accuracy.

Their calculation equations are as follows:

$$\begin{aligned} \mathrm{Acc} &= \frac{\mathrm{NU} + \mathrm{NT}}{NU + FT + FU + NT}, \\ \mathrm{DR} &= \frac{\mathrm{NU}}{\mathrm{NU} + \mathrm{FT}}, \\ \mathrm{FAR} &= \frac{\mathrm{FU}}{\mathrm{NU} + \mathrm{FU}}, \\ \mathrm{MAR} &= \frac{\mathrm{FT}}{\mathrm{NU} + \mathrm{FT}}, \end{aligned} \tag{17}$$

where NU is the number of abnormal traffic data samples classified as abnormal. NT is the number of normal traffic data samples classified as normal. FU is the number of normal traffic data samples classified as abnormal. FT is the number of abnormal traffic data samples classified as normal. The confusion matrix is shown in Table 2.

The main types of attacks in WSN networks are black hole attacks, gray hole attacks, flooding attacks, replay routing attacks, and wormhole attacks [39]. The following four types of attacks are included in the NSL-KDD dataset: denial of service attacks (DoS) [40], sniffing attacks (probe) [41], unauthorized access from a remote machine to a local machine (R2L) [42], and unauthorized access to local superuser (root) privileges (U2R) [43].

Input: training sample set U, sample instances to be classified $J = \{g_1, g_2, \cdots, g_w\}$, and category label $C = \{c_1, c_2, \cdots, c_z\}$
Output: sample J belongs to category c_j
z = number of categories
Obtain the prior probability $U(c_x)$
for each x in z
  $t = 0$
  $s = 0$
  for each x in U
    $t = t + 1$
    if $(I \in c_x)$ $s = s + 1$
  end for
  $U(c_x) = s/t$
  $U = \{u_1, u_2, \cdots, u_z\}$
  for each x in z
    $U(J \mid c_x) = 1$
    for each y in w
      weight $(x, y) = W_{\mathrm{JS}}(x, y) * W_{\mathrm{ICF}}(g_y)$
      $U(J \mid c_x) = U(J \mid c_x) * U(g_y \mid c_x) *$ weight (x, y)
    end for
    $u_x = U(c_x) * U(J \mid c_x)$
  end for
  $c_J = (c_x \mid u_x = \max\{u_1, u_2, \cdots, u_z\})$
end for
output (c_J)

Algorithm 1: The JINB algorithm.

Table 1: Distribution of NSL-KDD training data.

Types of attacks	Data distribution	
	Number of samples	Percentage (%)
Normal	69232	52.28
DoS	40816	30.82
Probe	12768	9.64
R2L	9545	7.21
U2R	66	0.05
Total	132,427	100

Table 2: Confusion matrix.

Predict	Actual	
	0	1
0	NT	FT
1	FU	NU

4.3. Experimental Results and Analysis

4.3.1. Algorithm Comparison Experiment. The experiments in this section are conducted using the LIB MATLAB simulation platform. To validate the performance of the JINB intrusion detection algorithm, a series of simulation experiments are conducted. These experiments involve a comparative analysis between the algorithm proposed in this paper and other existing algorithms. Furthermore, the proposed misuse detection module and anomaly detection module are individually simulated to evaluate their effectiveness in detecting specific types of attacks. The experiments in this subsection test the NSL-KDD dataset using OAA (one against all) [44], SVM (support vector machine), IBT (improved binary tree), HNB (hidden Naive Bayesian) [45], XLSTM [46], and the proposed algorithm. Each group of experiments uses the ten-fold crossover method to find the Acc, DR, FAR, and MAR of different attack types and finally takes the corresponding average value as the experimental result. The final experimental results are shown in Table 3.

The JINB algorithm is compared with other mainstream algorithms (OAA, SVM, IBT, HNB, and XLSTM) in terms of intrusion detection, and the results are measured using accuracy, DR, FAR, and MAR (see Figure 2).

As seen in Figure 2(a), the detection accuracy of the JINB algorithm has improved more significantly. From Figure 2(b), we can see that in terms of the detection rate, the detection rate of various intrusion types has been improved except for DoS, which is slightly lower than that of the HNB method. Figure 2(c) illustrates that the JINB algorithm has significantly reduced the false alarm rate compared to each of the other intrusion types, except for DoS and R2L. Figure 2(d) illustrates that the JINB algorithm has a significantly lower false alarm rate for each intrusion type than the other compared algorithms.

To assess the growth ratio, which reflects the relationship of detection rates between the JINB algorithm and various algorithms for different types of network attacks, Table 4 demonstrates the proportional comparison of the JINB algorithm with the OAA, SVM, IBT, HNB, and LSTM algorithms. Here, the growth ratio is calculated by comparing the detection rate of the JINB algorithm to that of various reference algorithms in the context of different network

Table 3: Comparison of detection performance (%) before and after algorithm improvement.

Algorithm	Normal	DoS	Probe	R2L	U2R
JINB Acc	94.69	98.96	99.18	95.8	99.95
NB Acc	91.48	87.57	97.19	95.91	98.04
JINB DR	94.38	98.46	82.29	8.32	13.62
NB DR	87.38	84.99	65.73	1.83	1.17
JINB FAR	2.86	0.72	0.58	1.09	0.09
NB FAR	5.14	3.58	2.25	4.71	1.11
JINB MAR	1.37	0.86	0.95	5.17	11.63
NB MAR	4.06	1.35	2.51	8.23	14.39

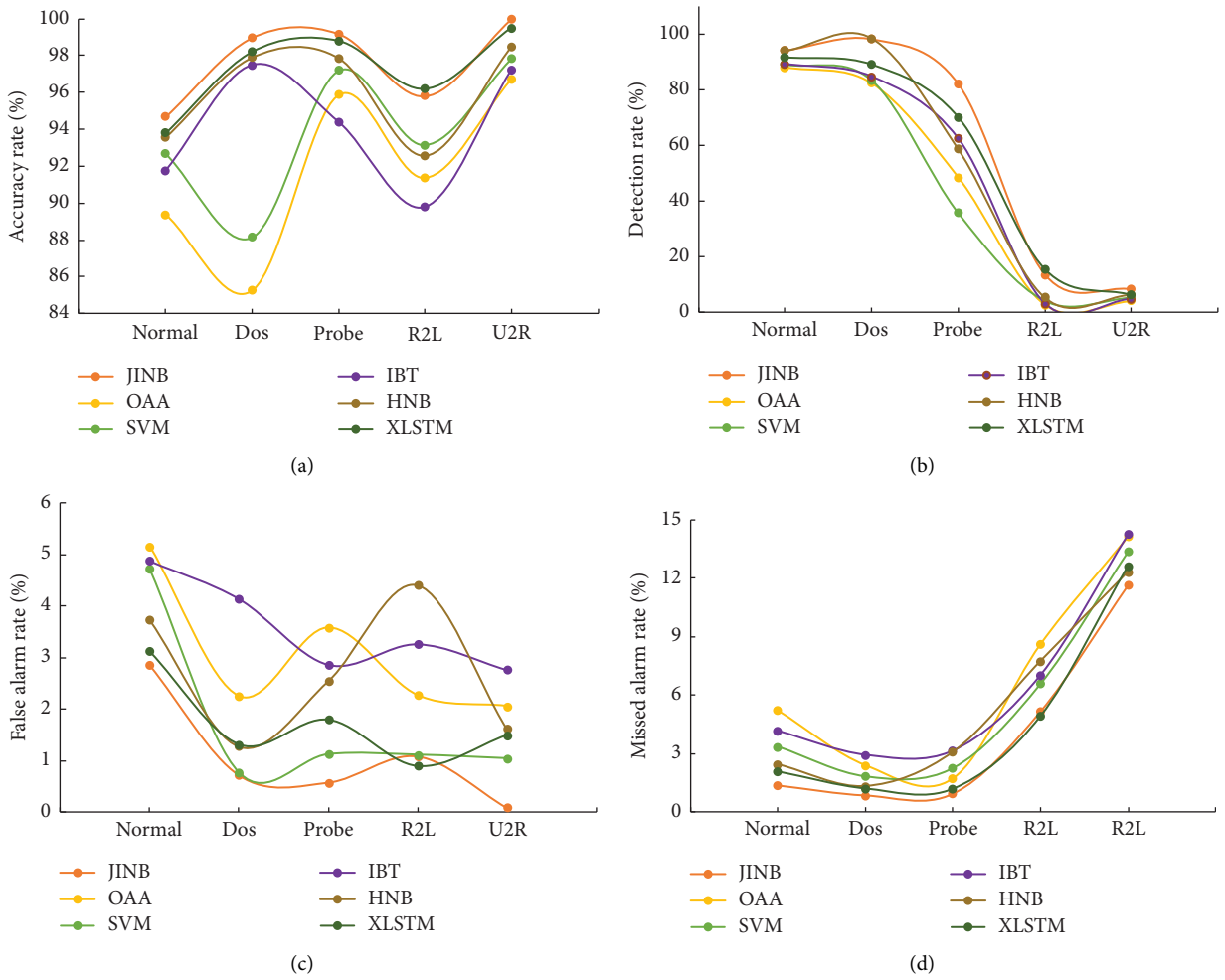

Figure 2: Comparison of the performance of each classification algorithm. (a) Comparison of accuracy. (b) Comparison of DR. (c) Comparison of FAR. (d) Comparison of MAR.

Table 4: Detection rate growth ratio of this algorithm to other algorithms.

Comparison	Growth ratio				
	Normal	DoS	Probe	R2L	U2R
JINB vs. OAA	6.2	16.1	33.7	4.2	10.9
JINB vs. SVM	5.1	14.9	46.5	3.0	9.5
JINB vs. IBT	5.1	13.7	9.8	3.5	10.5
JINB vs. HNB	0.3	−0.1	3.6	1.9	8.2
JINB vs. LSTM	1.2	2.5	5.7	−0.2	4.3

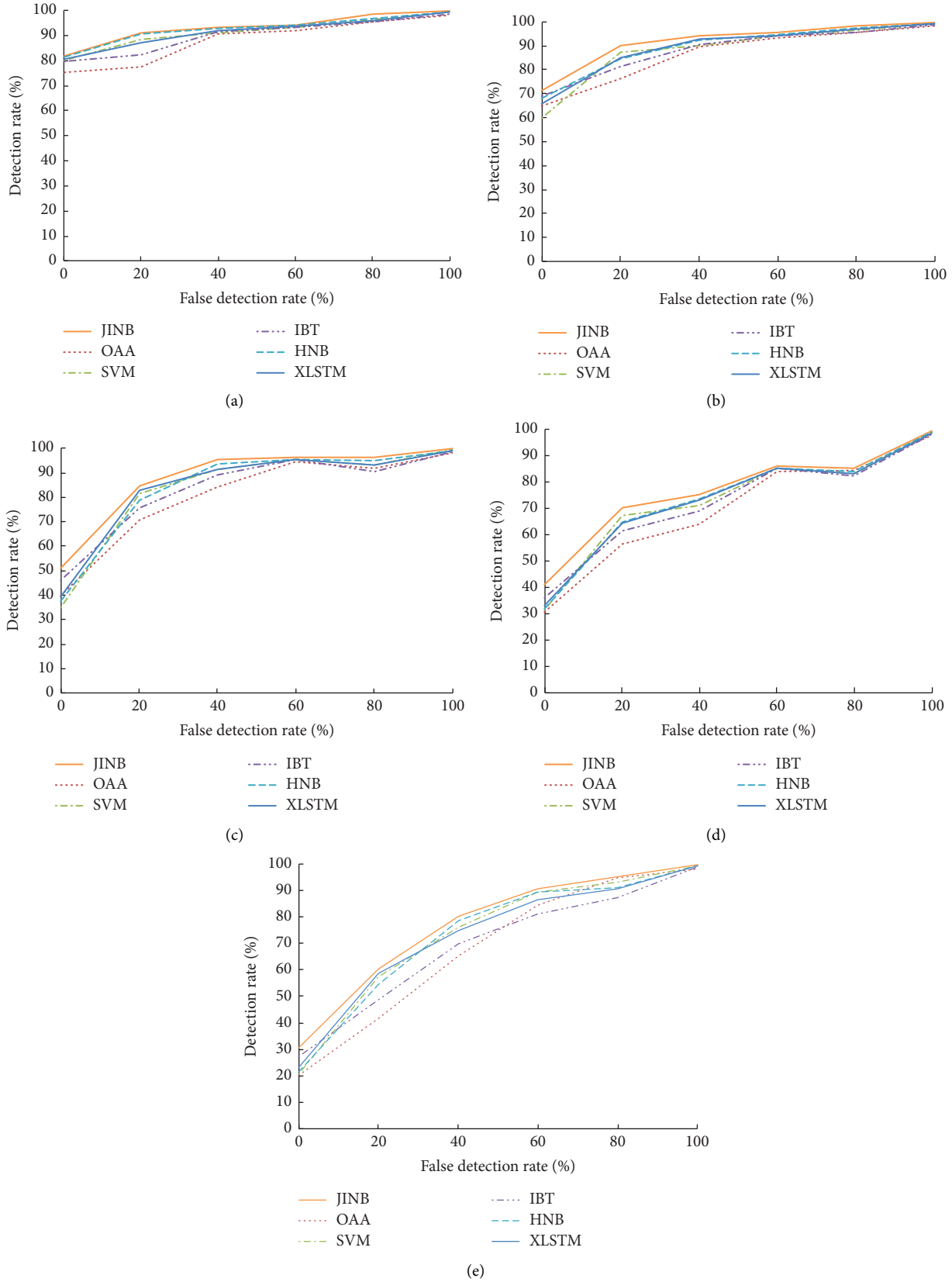

Figure 3: ROC diagram using five methods under different types of attacks. (a) Normal detection. (b) DoS attack. (c) Probe attack. (d) R2L attack. (e) U2R attack.

attack types. The calculation involves determining the difference between the detection rate of the JINB algorithm and the detection rate of each reference algorithm. This difference is then divided by the detection rate of the reference algorithm, and the result is multiplied by 100 to obtain a percentage representation.

By looking at Table 4, it is clear that

(1) In the detection of normal and R2L attacks, the growth ratio is not much different.

(2) The growth ratio reaches the maximum when detecting probe attacks, reaching 46.5%. This shows a very large growth ratio compared to the detection of normal, DoS, R2L, and U2R, and the growth ratio increases as the number of samples is smaller than the sample set. However, it decreases in U2R, which may be caused by the small percentage and insufficient data samples.

Normal detection and DoS, probe, R2L, and U2R attack types are compared under these five algorithms, and the ROC curve of experimental results is obtained, as shown in Figures 3(a)–3(e).

The variation of DR and FAR of the five algorithms for different kinds of attacks can be more intuitively seen by observing Figure 3. The detection effect of the proposed algorithm is significantly higher than the other four compared algorithms as shown by the ROC graph. The detection effect of the algorithm in this paper is more intuitively shown as the data imbalance rate increases.

Finally, an observation of the detection times recorded for the five algorithms (as shown in Table 5) reveals that the proposed algorithm has significantly reduced the time complexity. In particular, the reduction in time is larger when testing, and the exact calculation yields a 52% reduction in testing time relative to OAA. Compared to XLSTM, the training time and detection time of the algorithm in this paper are reduced by 48.9% and 55.5%, respectively.

The time complexity includes the training time for the classifier corresponding to the algorithm and the detection time for the algorithm to perform the attack detection, and the time complexity is expressed as shown in the following equation:

$$\text{Time complexity} = \text{training time} + \text{testing time}. \quad (18)$$

Based on the abovementioned simulation environment, intrusion detection is simulated using NS2. NS2 records the actions of each packet at every link and node during the simulation using a specific format trace file. It is instrumental in simulating probabilistic broadcast schemes for conventional ad hoc networks, intelligent routing protocols for wireless sensor networks, and routing protocols for flying ad hoc networks (FANETs).

As the quantity of malicious nodes in the network grows, the network topology becomes more intricate and the network is subjected to more DoS attacks, causing the detection rates of various algorithms to decline. The proposed algorithm resolves diversity and outlier sensitivity issues. This enhances its generalization capacity, making it outperform other comparative algorithms in this paper. Even when there are more malicious nodes, the algorithm in this paper has a good performance (as shown in Figure 4).

Table 5: Time complexity of the five algorithms.

Algorithms	Training time (s)	Testing time (s)
OAA	14939	2161
SVM	11217	1686
IBT	10588	1395
HNB	10376	1274
XLSTM	18762	2327
JINB	9586	1038

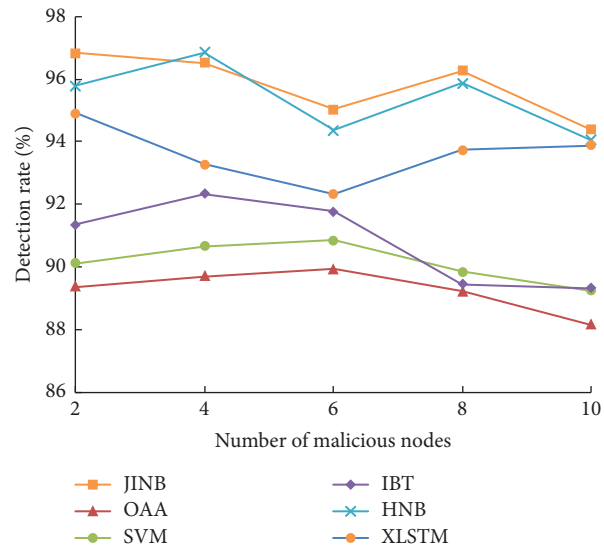

Figure 4: Relationship between number of malicious nodes and detection rate.

4.3.2. Network Energy Efficiency Analysis. As shown in Figures 5 and 6, the average remaining energy of the network nodes and the number of surviving nodes change over time for OAA, SVM, IBT, HNB, XLSTM, and the proposed algorithm. The network survival time is the longest for OAA defence because there is no additional energy consumption. HNB defence requires two modules to be turned on at the same time, which leads to a sharp increase in the energy consumption of the cluster head nodes and reduces the survival time of the network. In contrast, JINB defence only turns on one detection module at a time. SVM defence, IBT defence, and XLSTM defence activate a similar number of intrusion and misuse detection modules during network attacks and defence. This leads to comparable energy consumption and similar average energy and survival number curves of network nodes.

Experiments have proven that the JINB defence strategy effectively extends the survival time of the network and balances the accuracy and energy efficiency of the intrusion detection system.

The proposed intrusion detection algorithm based on feature-weighted NB improves the detection accuracy of

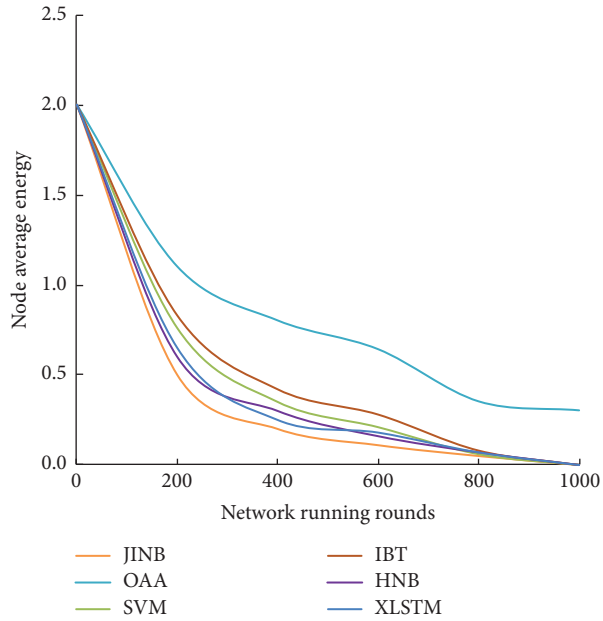

Figure 5: Average energy of nodes.

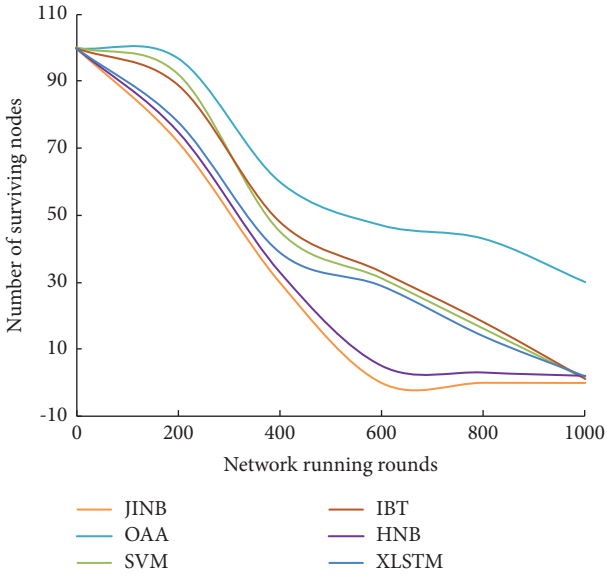

Figure 6: Number of surviving nodes.

network attacks in wireless sensor networks. By integrating a feature weighting algorithm, Jensen–Shannon divergence, and inverse category frequency, the algorithm enhances the performance of the Naive Bayes classifier. This allows for real-time processing of network events using the corresponding network traffic data, resulting in a significant improvement in the detection accuracy for various attack types. Therefore, the proposed model provides practicality for real-time network intrusion detection. Furthermore, the proposed intrusion detection algorithm based on feature-weighted NB offers cost-effectiveness in network security. By improving the detection accuracy and reducing false detections, the algorithm enhances the effectiveness of network attack detection. This leads to cost savings by minimizing expenses related to false alarms and mitigating potential losses from undetected attacks. In addition, the algorithm utilizes existing network traffic data and incorporates weighting factors without imposing significant resource requirements. This makes it cost-effective to implement and integrate into existing systems. The algorithm's real-time event processing capability further contributes to its cost-effectiveness by enabling prompt responses to detected attacks. Overall, the proposed model provides a cost-effective solution for enhancing the security of wireless sensor networks.

5. Conclusion

The security of wireless sensor networks is facing various challenges. To improve the detection accuracy of network attacks, an improved intrusion detection algorithm based on feature-weighted NB is proposed in this paper. A feature weighting algorithm is proposed by assigning corresponding weights to different feature terms according to the situation. The Jensen–Shannon (JS) divergence is combined with feature weighting and inverse category frequency (ICF) to improve the Naive Bayes algorithm. In the experimental session, the algorithm of this paper is compared and analyzed with other algorithms on the NSL-KDD dataset. The results show that the wireless network intrusion algorithm proposed in this paper can ensure improved detection accuracy and detection rate, while reducing the false detection rate.

The limitations of this study are related to the algorithm's performance in the data preprocessing stage. Specifically, (1) the discretization and normalization abilities of the algorithm require improvement. To enhance the algorithm's stability and adaptability, future work should focus on analyzing and implementing more effective data preprocessing methods. (2) The study acknowledges the need to develop a threat model for evaluating the algorithms' performance in reducing system threat metrics. This development will comprehensively assess the algorithms' effectiveness in real-world threat scenarios. In the future, we will comprehensively discuss real-time computing efficiency checks and possibilities, including an exploration of the applicability in broader network environments and large-scale deployments.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This work was supported by the Henan Vocational College of Tuina.

References

[1] A. E. Omolara, A. Alabdulatif, O. I. Abiodun et al., "The internet of things security: a survey encompassing unexplored areas and new insights," *Computers and Security*, vol. 112, Article ID 102494, 2022.

[2] J. H. Anajemba, C. Iwendi, I. Razzak, J. A. Ansere, and I. M. Okpalaoguchi, "A counter-eavesdropping technique for optimized privacy of wireless industrial iot communications," *Institute of Electrical and Electronics Engineers Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6445–6454, 2022.

[3] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.

[4] A. Hilmani, A. Maizate, and L. Hassouni, "Automated real-time intelligent traffic control system for smart cities using wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–28, 2020.

[5] A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba, and S. A. Bahaj, "Deep learning for intrusion detection and security of internet of things (iot): current analysis, challenges, and possible solutions," *Security and Communication Networks*, vol. 2022, Article ID 4016073, 13 pages, 2022.

[6] M. Rokonuzzaman, M. K. Mishu, N. Amin et al., "Self-Sustained autonomous wireless sensor network with integrated solar photovoltaic system for internet of smart home-building (IoSHB) applications," *Micromachines*, vol. 12, no. 6, p. 653, 2021.

[7] D. D. K. Rathinam, D. Surendran, A. Shilpa, A. S. Grace, and J. Sherin, "Modern agriculture using wireless sensor network (WSN)," in *Proceedings of the 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 515–519, IEEE, Coimbatore, India, March 2019.

[8] S. R. Jondhale, R. Maheswar, and J. Lloret, *Received Signal Strength Based Target Localization and Tracking Using Wireless Sensor Networks*, Springer, Cham, Berlin, Germany, 2022.

[9] V. S. Patil, Y. B. Mane, and S. Deshpande, "Fpga based power saving technique for sensor node in wireless sensor network (wsn)," *Computational Intelligence in Sensor Networks*, Springer, Berlin, Germany,pp. 385–404, 2019.

[10] B. Bhushan and G. Sahoo, "Requirements, protocols, and security challenges in wireless sensor networks: an industrial perspective," *Handbook of computer networks and cyber security: principles and paradigms*, Springer, Cham, Berlin, Germany,pp. 683–713, 2020.

[11] T. Moulahi, S. Zidi, A. Alabdulatif, and M. Atiquzzaman, "Comparative performance evaluation of intrusion detection based on machine learning in in-vehicle controller area network bus," *Institute of Electrical and Electronics Engineers Access*, vol. 9, pp. 99595–99605, 2021.

[12] R. Zhao, J. Yin, Z. Xue et al., "An efficient intrusion detection method based on dynamic autoencoder," *Institute of Electrical and Electronics Engineers Wireless Communications Letters*, vol. 10, no. 8, pp. 1707–1711, 2021.

[13] S. Maheswari and K. Arunesh, "Unsupervised Binary BAT algorithm based Network Intrusion Detection System using enhanced multiple classifiers," in *Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 885–889, IEEE, Trichy, Tamil Nadu, India, September 2020.

[14] L. Huang and Q. Zhu, "Dynamic bayesian games for adversarial and defensive cyber deception," *Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings*, Springer, Cham, Berlin, Germany,pp. 75–97, 2019.

[15] Z. Wang, Y. Zeng, Y. Liu, and D. Li, "Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection," *IEEE Access*, vol. 9, pp. 16062–16091, 2021.

[16] D. Ngabo, W. Dong, E. Ibeke, C. Iwendi, and E. Masabo, "Tackling pandemics in smart cities using machine learning architecture," *Mathematical Biosciences and Engineering*, vol. 18, no. 6, pp. 8444–8461, 2021.

[17] A. Alsharef, S. Sonia, K. Kumar, and C. Iwendi, "Time series data modeling using advanced machine learning and AutoML," *Sustainability*, vol. 14, no. 22, Article ID 15292, 2022.

[18] P. Wu and H. Guo, "LuNET: a deep neural network for network intrusion detection," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 617–624, IEEE, Xiamen, China, December 2019.

[19] A. N. Jaber and S. U. Rehman, "FCM–SVM based intrusion detection system for cloud computing environment," *Cluster Computing*, vol. 23, no. 4, pp. 3221–3231, 2020.

[20] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Computer Networks*, vol. 177, Article ID 107315, 2020.

[21] Z. Ling, Z. J. Wei, F. N. Mei, and Z. H. Hao, "Intrusion detection model based on rough set and random forest," *International Journal of Grid and High Performance Computing*, vol. 14, no. 1, pp. 1–13, 2022.

[22] T. Ling, L. Chong, X. Jingming, and C. Jun, "Application of self-organizing feature map neural network based on K-means clustering in network intrusion detection," *Computers, Materials and Continua*, vol. 61, no. 1, pp. 275–288, 2019.

[23] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S. J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2579, 2019.

[24] L. Duan and Y. Xiao, "An intrusion detection model based on fuzzy C-means algorithm," in *Proceedings of the 2018 8th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 120–123, IEEE, Beijing, China, June 2018.

[25] A. Mourad, H. Tout, O. A. Wahab, H. Otrok, and T. Dbouk, "Ad hoc vehicular fog enabling cooperative low-latency intrusion detection," *Institute of Electrical and Electronics Engineers Internet of Things Journal*, vol. 8, no. 2, pp. 829–843, 2021.

[26] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: centralized, on-device, or federated learning?" *Institute of Electrical and Electronics Engineers Network*, vol. 34, no. 6, pp. 310–317, 2020.

[27] S. Gautam, A. Henry, M. Zuhair, M. Rashid, A. R. Javed, and P. K. R. Maddikunta, "A composite approach of intrusion detection systems: hybrid RNN and correlation-based feature optimization," *Electronics*, vol. 11, no. 21, p. 3529, 2022.

[28] C. Dhasarathan and H. Shrestha, "An NLP based sentimental analysis and prediction: a dynamic approach," in *Proceedings of the Communication, Networks and Computing: Second International Conference, CNC 2020*, pp. 343–353, Springer Singapore, Gwalior, India, December 2021.

[29] H. I. H. Alsaadi, R. M. Almuttari, O. N. Ucan, and O. Bayat, "An adapting soft computing model for intrusion detection system," *Computational Intelligence*, vol. 38, no. 3, pp. 855–875, 2022.

[30] N. Alsharif, "Ensembling PCA-based feature selection with random tree classifier for intrusion detection on IoT network," in *Proceedings of the 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 317–321, IEEE, Jakarta, Indonesia, October 2021.

[31] L. Zhang, H. Yan, and Q. Zhu, "An improved LSTM network intrusion detection method," in *Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pp. 1765–1769, IEEE, Chengdu, China, December 2020.

[32] R. Panigrahi, S. Borah, A. K. Bhoi et al., "A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets," *Mathematics*, vol. 9, no. 7, p. 751, 2021.

[33] K. Hara and K. Shiomoto, "Intrusion detection system using semi-supervised learning with adversarial auto-encoder," in *Proceedings of the NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–8, IEEE, Budapest, Hungary, April 2020.

[34] W. Li, L. Ke, W. Meng, and J. Han, "An empirical study of supervised email classification in Internet of Things: practical performance and key influencing factors," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 287–304, 2022.

[35] L. Jiang, L. Zhang, L. Yu, and D. Wang, "Class-specific attribute weighted naive Bayes," *Pattern Recognition*, vol. 88, pp. 321–330, 2019.

[36] B. Bouyeddou, B. Kadri, F. Harrou, and Y. Sun, "Non-parametric Kullback-Leibler distance-based method for networks intrusion detection," in *Proceedings of the 2020 International Conference on Data Analytics for Business and Industry: Way towards a Sustainable Economy (ICDABI)*, pp. 1–5, IEEE, Sakheer, Bahrain, October 2020.

[37] A. J. C. Sunder and A. Shanmugam, "Jensen–Shannon divergence based independent component analysis to detect and prevent black hole attacks in healthcare WSN," *Wireless Personal Communications*, vol. 107, no. 4, pp. 1607–1623, 2019.

[38] T. L. Huoh, Y. Luo, and T. Zhang, "Encrypted network traffic classification using a geometric learning model," in *Proceedings of the 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 376–383, IEEE, Bordeaux, France, May 2021.

[39] Z. Alansari, N. B. Anuar, A. Kamsin, and M. R. Belgaum, "A systematic review of routing attacks detection in wireless sensor networks," *PeerJ Computer Science*, vol. 8, Article ID e1135, 2022.

[40] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, no. 6, p. 916, 2020.

[41] K. O'Neal and S. Yilek, "Interactive history sniffing with dynamically-generated QR codes and CSS difference blending," in *Proceedings of the 2022 IEEE Security and Privacy Workshops (SPW)*, pp. 335–341, IEEE, San Francisco, CA, USA, May 2022.

[42] Y. Su, W. Zhang, W. Tao, and Z. Qiao, "A network illegal access detection method based on PSO-SVM algorithm in power monitoring system," in *Proceedings of the Cloud Computing and Security: 4th International Conference, ICCCS 2018*, pp. 450–459, Springer International Publishing, Haikou, China, June 2018.

[43] R. Yadav, I. Sreedevi, and D. Gupta, "Augmentation in performance and security of WSNs for IoT applications using feature selection and classification techniques," *Alexandria Engineering Journal*, vol. 65, pp. 461–473, 2023.

[44] J. Wu, P. Guo, Y. Cheng, H. Zhu, X. B. Wang, and X. Shao, "Ensemble generalized multiclass support-vector-machine-based health evaluation of complex degradation systems," *Institute of Electrical and Electronics Engineers*, vol. 25, no. 5, pp. 2230–2240, 2020.

[45] H. A. Mahmood and S. H. Hashem, "Network intrusion detection system (NIDS) in cloud environment based on hidden Naïve Bayes multiclass classifier," *Al-Mustansiriyah Journal of Science*, vol. 28, no. 2, pp. 134–142, 2018.

[46] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Computer Communications*, vol. 199, pp. 113–125, 2023.