LUPINE PUBLISHERS

Advances in Robotics & Mechanical Engineering

DOI: 10.32474/ARME.2023.04.000183

ISSN: 2643-6736

Research Article

Research on Intrusion Detection Based on Enhanced Random Forest Algorithm

Lu Caiwu[1], Cao Yunxiang[1] and Wang Zebin[2]*

[1]*School of Resources Environment and Materials, Xi'an University of Architecture and Technology; Xi'an, China*

[2]*School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an, China*

***Corresponding author:** Wang Zebin, School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China

Received: September 08, 2023 **Published:** September 14, 2023

Abstract

To address the challenges posed by high data dimensionality and class imbalance in intrusion detection, which result in increased computational complexity, resource consumption, and reduced classification accuracy, this paper presents an intrusion detection algorithm based on an improved Random Forest approach. The algorithm employs the Bald Eagle Search (BES) optimization technique to fine-tune the Kernel Principal Component Analysis (KPCA) algorithm, enabling optimized dimensionality reduction. The processed data is then fed into a cost-sensitive Random Forest classifier for training, with subsequent model validation conducted on the reduced-dimension data. Experimental results demonstrate that compared to traditional Random Forest algorithms, the proposed method reduces training time by 11.32 seconds and achieves a 5.59% increase in classification accuracy, an 11.7% improvement in specificity, and a 0.0558 increase in G-mean value. These findings underscore the promising application potential and performance of this approach in the field of network intrusion detection.

Keywords: Machine learning; Data dimensionality reduction; Cost sensitive; Random Forest; Intrusion detection

Introduction

With the increasing risks in network security, the implementation of effective intrusion detection mechanisms has become a crucial strategy for safeguarding computer systems and network security [1-4]. Traditional intrusion detection methods heavily rely on known attack patterns and behaviors, acquired through expert knowledge or historical data. Consequently, their effectiveness in detecting new and unknown attack methods is limited [5,6]. In addressing this issue, decision tree algorithms [7] in the form of attribute splitting have improved the efficiency of intrusion detection by classifying network behaviors and determining their involvement in the intrusion process. However, these methods often neglect the aspect of detection accuracy. Support Vector Machines, on the other hand, excel in intrusion detection with superior accuracy and classification efficiency when dealing with limited data experience. However, they face significant challenges in encoding and normalizing data, particularly with the emergence of novel unknown attack methods, making them less suitable for the evolving landscape of information technology [8]. K-means clustering, known for its speed and interpretability, has also found its place in intrusion detection. It excels in grouping similar instances into subsets for intrusion determination. However, it exhibits sensitivity to outliers and the difficulty of determining the optimal K value [9]. In contrast, the Random Forest algorithm stands out as a favorable approach in the field of intrusion detection and has seen extensive exploration and application [10,11]. As research has progressed, literature [12-15] has compared various datasets commonly used in intrusion detection, revealing their complex high-dimensional nature and the imbalance between positive and negative classes.

These characteristics can lead to lower detection rates for certain classes. Nevertheless, the Random Forest algorithm, when dealing with large datasets, demands the construction

of multiple decision trees and ensemble integration. When computational resources are limited, this can result in excessive resource consumption and reduced computational efficiency. Additionally, when high-dimensional sparse data serves as input to the classification algorithm, the similarity between data samples becomes pronounced, making it challenging to identify effective split points. Furthermore, Random Forest employs Gini impurity [16] in node splitting, which tends to favor majority classes in imbalanced datasets, resulting in poor classification performance for minority classes.

To address the challenges posed by high-dimensional data and class imbalance in the Random Forest algorithm, researchers have primarily focused on two approaches: data dimensionality reduction and feature selection. To improve classification accuracy when facing imbalanced datasets, efforts have been made at the data, algorithm, and decision levels. When dealing with high-dimensional datasets, literature [17-20] has employed feature selection methods to reduce dataset dimensionality by selecting the most relevant features. However, this approach may inadvertently remove some useful features, potentially leading to the loss of inter-feature relationships. Literature [21] explored the combination of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), mapping the original feature set into a lower-dimensional space and subsequently applying the reduced data to classifier training, offering a direction for data dimensionality reduction in intrusion detection scenarios. Moreover, to address the issue of classification results leaning toward majority classes due to class imbalance, literature [22] combined adaptive oversampling, under sampling clustering algorithms, and Gaussian mixture models at the data level and applied them to preprocess the original dataset for classification decision-making. While sampling methods hold advantages in handling imbalanced datasets, strategies that modify data distributions may affect subsequent model construction and hinder the effective extraction of underlying relationships between data. Literature [23] proposed improvements at the algorithm level by embedding cost information into the model, obtaining cost-sensitive information through expected loss minimization, and applying it to both binary and multi-class classification scenarios, demonstrating superior performance in classification problems. Literature [24,25] addressed the issue at the decision level by adjusting classifier decision thresholds, shifting them towards the majority class to reduce the probability of misclassifying minority classes, effectively mitigating class imbalance. Traditional Random Forest algorithms applied in intrusion detection suffer from excessive computational costs, low classification accuracy, and a tendency to favor minority class results. Current research indicates that to address these issues, optimizing the Random Forest algorithm should consider three key aspects: data dimensionality reduction, algorithm enhancement, and decision evaluation. Therefore, this paper proposes a Random Forest classification model designed for handling high-dimensional and class-imbalanced data. The aim is to address the limitations in network intrusion detection research, ensuring the security and integrity of computer networks.

Research method

Introduction to Intrusion Detection

Intrusion detection involves the identification of unauthorized activities [26]. It entails the collection and analysis of network behavior, security logs, audits, data, information available on other networks, and critical information from various points within a computer system. The goal is to inspect whether there are indications of security policy violations or signs of attacks within a network or system. Intrusion detection, as an actively proactive security technique, offers real-time protection against internal and external attacks as well as accidental mishaps. It intercepts and responds to intrusions before they can harm a network, making it considered the second line of defense after firewalls. It accomplishes this without significantly affecting network performance, making it suitable for continuous network monitoring [27].

The Random Forest Algorithm

The Random Forest algorithm is an ensemble learning method that builds a large-scale ensemble model by combining multiple independent decision trees. Each decision tree is trained on a randomly sampled subset of the training data. Ultimately, classification decisions are made through a voting mechanism among the decision trees.

The specific construction process is as follows:

1. Sample selection: Select randomly from the original data set using Bootstrap sampling to form independent data subsets $\{D_N \; n = 1, 2, \ldots, N\}$

2. Feature selection: features are randomly selected from features.

3. Decision tree construction: Build a decision tree according to the samples of the data set. At each node, the best partition feature is selected according to Gini impurity.

$$\mathrm{Gini}(t) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

Integrated prediction: For a new sample, a majority vote is conducted through the prediction results of each decision tree to finally determine the classification result of the sample. In the context of intrusion detection, applying the Random Forest method can be challenging due to high-dimensional data in the dataset and class imbalance, which leads to suboptimal performance in metrics such as classification accuracy. Furthermore, the algorithm's classification performance is heavily dependent on two factors: the accuracy of individual decision trees and the bias in the voting results among multiple decision trees. Therefore, to enhance the overall performance of Random Forest in intrusion detection, we are considering the integration of data dimensionality reduction algorithms and cost-sensitive learning methods. The critical steps

and improvement strategies are depicted in Figure 1. This paper addresses the enhancement of the Random Forest algorithm from three key perspectives: Data Dimensionality Reduction for Classifier Input: In the first aspect, we focus on reducing the dimensionality of the data fed into the classifier. Construction of Cost-Sensitive Base Classifiers: The second aspect involves the creation of cost-sensitive base classifiers. Weighted Majority Voting at Decision Stage: In the third aspect, we introduce weighted majority voting techniques at both the leaf nodes of decision trees and the ensemble decision stage. These improvements collectively aim to elevate the performance and effectiveness of the Random Forest algorithm in the context of intrusion detection.

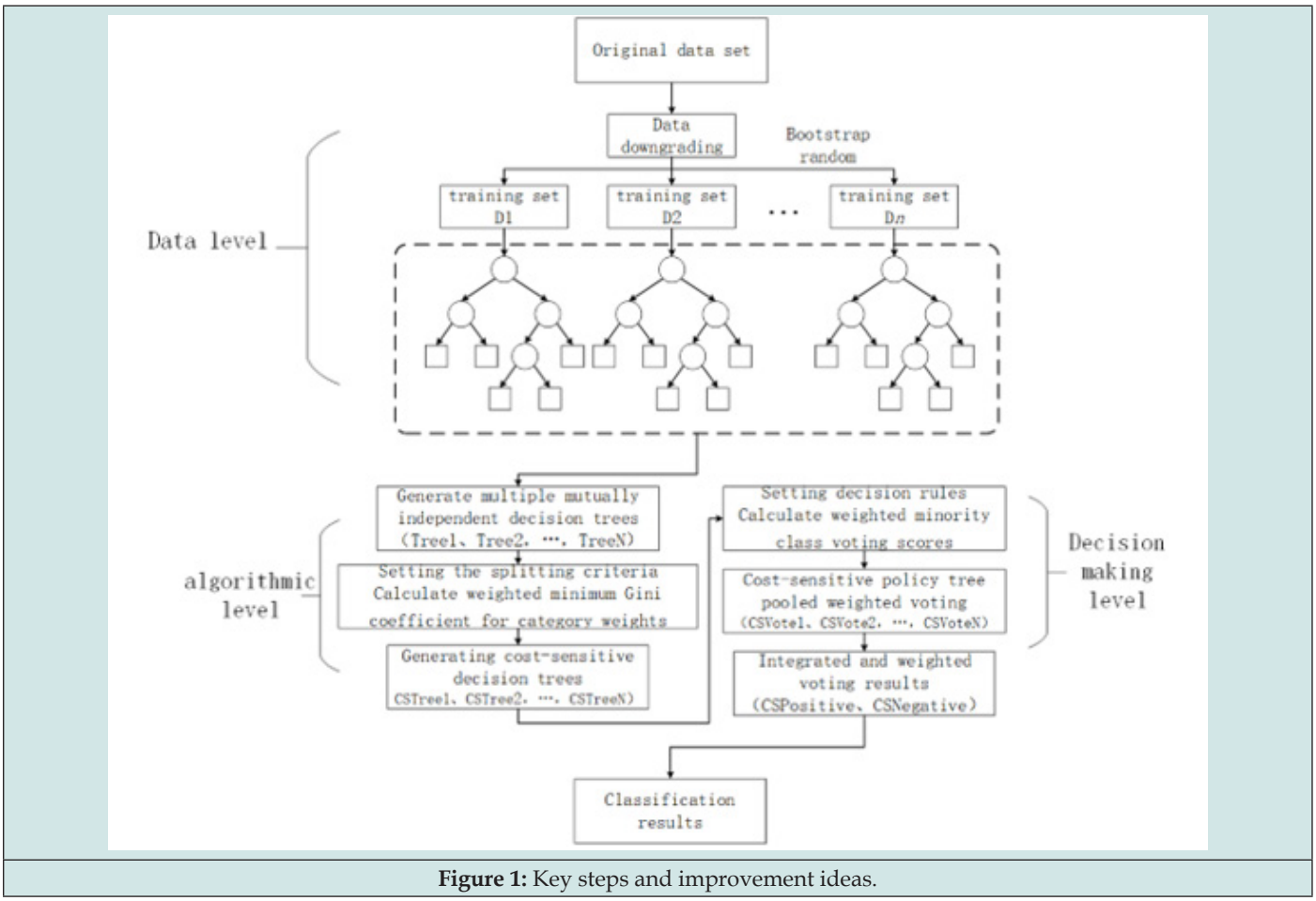

Figure 1: Key steps and improvement ideas.

Cost-Sensitive Random Forest Algorithm

Data-Level Improvement

The redundancy and high dimensionality of intrusion detection datasets can significantly impact classification algorithms. On one hand, high-dimensional datasets increase the computational complexity of classifiers and consume more storage resources. On the other hand, redundant features reduce model usability and may introduce noise and unnecessary information, affecting classification performance. To mitigate the adverse effects of high-dimensional data, we introduce the Kernel Principal Component Analysis (KPCA) algorithm [28]. This approach maps the original dataset's samples to a higher-dimensional space and then projects and reduces the high-dimensional sample data while maximizing the variance of the projected data. This results in a reduction of data dimensions and the removal of redundant information. The working principle is as follows:

Let each column in the sample be x_i , The sample set is $X = [x_1, x_2, ..., x_N]$ 。Now use a nonlinear map ϕ of vector x_i in X to a higher dimensional space τ （Record as dimension） , Obtain $D \times N$ new matrix for A $\phi(X) = [\varphi(x_1), \varphi(x_2), ..., \varphi(x_n)]$ deduces that the covariance matrix in τ against $\phi(X)$ is:

$$C_\tau = \frac{1}{N}\phi(X)[\phi(X)]^T = \frac{1}{N}\sum_{i=1}^{N}\varphi(x_i)[\varphi(x_i)]^T$$

The eigenvalues for solving the covariance matrix are:

$$C_\tau p = \lambda p$$

Where the D-dimensional column vector P is the weight vector of the feature space, which can be expressed as a linear combination of $\varphi(x_i)$, namely:

$$p = \sum_{i=1}^{N} a_i \varphi(x_i) = \phi(X)a$$

By defining the matrix $K=[\phi(X)]^T \phi(X)$, you can solve for non-zero eigenvalues:

$$Ka = \lambda a$$

While the KPCA algorithm demonstrates advantages in handling non-linear relationships and complex data structures, it also has certain limitations. The algorithm's kernel function parameters need to be adjusted according to the data's characteristics, and different parameter choices can result in varying dimensionality reduction outcomes. To address this, we integrate the Bald Eagle Search (BES) optimization algorithm [29], which effectively explores the solution space, reduces reliance on initial conditions, and completes parameter optimization for KPCA. The utilization of the BES algorithm for optimizing KPCA involves the following six steps:

1. Initialize the algorithm parameters and initialize the number of condor population $npop$, the number of algorithm iterations $Maxlt$, and the fitness function fobj.

$$BestSol.\cos t = \inf$$

2. Objective function evaluation to calculate the fitness of each individual's current position.

$$pop.\cos t(i) = fobj(pop.pos(i,:))$$

3. Select the search space, randomly select the search area, and determine the optimal search position as .

$$P_{i,new} = P_{best} \cdot \alpha \cdot rand \cdot (P_{mean} - P_i)$$

4. Search for space prey, find the best dive position, update the condor position.

$$P_{i,new} = P_i + x(i) \cdot (P_i - P_{mean}) + y(i) \cdot (P_i - P_{i+1})$$

5. Diving to capture prey, a rapid dive from the best position in the search space to the target prey, the rest of the population also move to the best position and attack the prey. rand is a random number between 0 and 1.

$$P_{i,new} = rand \cdot P_{best} + P_i + x(i) \cdot (P_i - c_1 \cdot P_{mean}) + y(i) \cdot (P_i - c_2 \cdot P_{best})$$

6. Determine whether the end condition is reached. If so, output the optimal result; otherwise, repeat step step2-step6.

When using BES optimization algorithm to improve KPCA algorithm, fitness function setting is the key point. According to the application scenario analysis of intrusion detection, we want the projected samples to be clustered as much as possible in the low-dimensional space, and the samples of different categories are far away from each other. The fitness function is constructed by calculating the inter-class distance and intra-class distance. The calculation process is as follows:

By taking samples of two different categories in the original data set $\phi(x_1)$ and $\phi(x_2)$, we can see that the mean vectors of the samples are $\phi(x_1)=\frac{1}{N}\sum_{i=1}^{N}\phi(x_i)$ and $\phi(x_2)=\frac{1}{M}\sum_{i=1}^{M}\phi(x_j)$ respectively, and the overall mean value of the samples can be calculated:

$$\phi = \frac{1}{N+M}\sum_{i=1}^{N+M}\varphi(x_i)$$

The intra-class divergence matrix S_x and inter-class divergence matrix S_y can be obtained:

$$\begin{cases} S_x = \sum_{i=1}^{2} n_i(\phi_i - \phi)(\phi_i - \phi)^T \\ S_y = \sum_{i=1}^{2}\sum_{x\in classi} n_i(\phi_i - \varphi(x_k))(\phi_i - \varphi(x_k))^T \end{cases}$$

Therefore, when the distance between samples of different categories is larger and the distance between samples of the same category is smaller, the separability of data samples is better. Therefore, the fitness function of BES is set as follows:

$$fobj = \frac{S_x}{S_y}$$

When fitness takes the minimum value, the kernel parameter of KPCA gets the optimal value, and the samples have the best separation in the feature space.

Algorithmic Enhancement

In the face of class imbalance within intrusion detection datasets, traditional Random Forest algorithms employ decision trees as base classifiers, with node splits based on randomly selected attributes. Typically, the chosen splitting criterion involves calculating the minimal impurity of child nodes after the split. For imbalanced datasets, the class distribution tends to concentrate within the category with lower impurity, leading to misclassification of the minority class. In the realm of intrusion detection, malicious attacks represent the minority class, while normal behavior constitutes the majority class. In the context of network traffic detection, conventional Random Forest models tend to detect normal behavior while overlooking malicious attacks. Inability to effectively detect malicious behavior in the minority class could pose significant security threats to computer systems. Therefore, improving the generation process of each base classifier is pivotal in classification work. By considering the cost associated with different classes and embedding cost-sensitive thinking into the training process of each base classifier, we utilize class weights to calculate the weighted minimum Gini coefficient for selecting the optimal split point. Such a cost-sensitive approach better accounts for cost disparities between different classes, resulting in

more accurate classification outcomes. This transformation of the expression ensures a focus on the integration of cost sensitivity into the training process of each base classifier, ultimately leading to improved classification accuracy. The expression of $Gini(t)$ is transformed into:

$$Gini_{\cos t}(t) = 1 - \sum_{i=1}^{c} [p(i \mid t) \cot s(i, j)]^2$$

Where, and represent categories; represents the number of categories.

Decision-Level Enhancement

To address the issue of imbalanced classification, it's not only essential to set splitting criteria within each base classifier at the algorithmic level, but also to incorporate cost-sensitive information into the voting process at the decision-tree leaf nodes at the decision level. In intrusion detection data, the minority class (N) represents malicious access behavior, while the majority class comprises regular access behavior (P). In the model's decision voting phase, traditional methods fail to adequately consider sample weights. They treat each sample equally through majority voting, without accounting for the differing costs associated with different types of errors. The consequences of misclassifying malicious behavior as normal behavior involve various losses that differ from the cost of misclassifying normal access behavior. The introduction of cost-sensitive methods in the Random Forest algorithm primarily revolves around defining an appropriate cost matrix. This cost matrix allocates different error classification costs to different classes, considering the costs associated with false positives and false negatives, with the aim of minimizing the total cost. Currently, cost matrices are typically obtained in two ways: through domain experts providing their expertise or by validating different cost matrices during the classifier training phase. However, in practical imbalanced classification scenarios, it may not be feasible to rely on expert knowledge alone to obtain a reliable cost matrix, especially when expert experience is limited. Therefore, employing different

methods to validate the cost matrix is more practical. As the intersection point of sensitivity and specificity curves represents high sensitivity and specificity simultaneously, this paper determines the classification threshold by selecting the intersection point of the sensitivity and specificity curves. This threshold can be derived by using the sensitivity and specificity curve intersection method on the validation set, allowing us to obtain the corresponding cost matrix:

$$\begin{pmatrix} cost(P,P) = 0 & cost(P,N) = threshold_{curve-cross} \\ cost(N,P) = threshold_{curve-cross} & cost(N,N) = 0 \end{pmatrix}$$

The cost matrix is incorporated into the classification decision-making process of random forest algorithm, and the weighted voting of minority class samples is used to improve the minority class's voice in the final decision and reduce the excessive bias to the majority class. Therefore, in the decision-making process, the probability of leaf node voting for a minority class is improved to:

$$p(N \mid t) cost(N,P) > p(P \mid t) cost(P,N)$$

$$p(N \mid t) cost(N,P) > (1 - p(N \mid t)) cost(P,N)$$

$$p(N \mid t) > \frac{cost(P,N)}{cost(N,P) + cost(P,N)}$$

The final category prediction result of random forest is obtained by weighted majority voting of all trees, and the decision tree with higher weight is more sensitive to the unbalanced classification problem, and its majority voting decision stage has greater decision weight.

Enhanced Intrusion Detection Algorithm Workflow

The algorithm workflow in this paper consists of four phases, as depicted in Figure 2:

1. Data Preprocessing: This phase involves preprocessing the original dataset. Categorical features are one-hot encoded, transforming unmanageable categorical features into numerical ones. Additionally, to mitigate significant differences in feature values, the dataset is normalized.
2. Data Dimensionality Reduction: In this phase, the algorithm calculates distances between categories and within categories to construct a fitness function. The Bald Eagle Search (BES) algorithm is employed for optimizing KPCA parameters. The optimal parameters are then used in the B-KPCA algorithm to reduce dimensionality in the intrusion detection dataset, creating a new feature subset.
3. Construction of Cost-Sensitive Random Forest Model: Cost matrices are introduced and applied to both the Gini function of base classifiers and the prediction in the decision tree classification voting. The model is trained with these considerations.
4. Model Validation: The final phase involves testing the trained model using a testing dataset. Multiple metrics are employed to evaluate and validate the classification performance of the model.

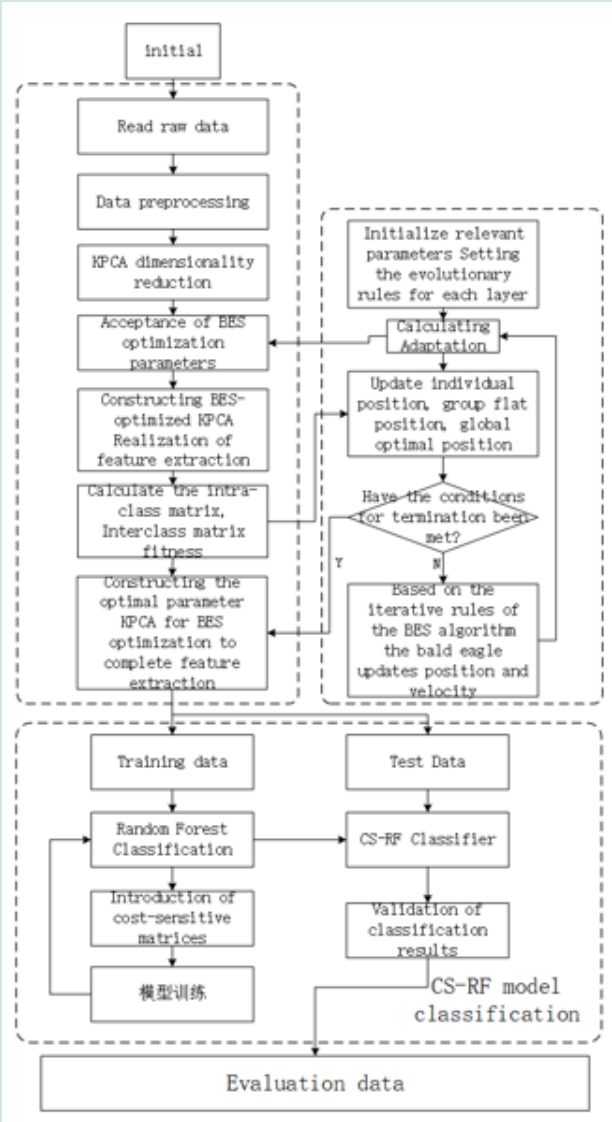

Figure 2: Intrusion detection algorithm process.

Experimental verification and result analysis

Experimental setup

Experimental environment setting

The experimental hardware environment is Windows 10, Intel Core i7-7700HQ@2.80GHz CPU, GeForce GTX1050Ti GPU, and 16GB RAM. The software environment uses VScode1.80 and Python 3.7.9.

Data set selection

Classic intrusion detection datasets such as KDD Cup 1999, NSL-KDD, and DARPA [30] are well-known, but they often suffer from high redundancy, duplicate samples, and may not fully represent modern network environments with emerging attack behaviors. The UNSW-NB15 dataset [31], developed by the University of New South Wales, stands out as a network intrusion detection dataset with diversity and authenticity. It provides real and complex intrusion detection scenarios, offering insights into network attacks and threats in real-world environments. Hence, utilizing the UNSW-NB15 dataset enables the evaluation of classifiers in realistic and intricate network environments. Detailed distribution information for this dataset is presented in Table 1. To address the challenges of high dimensionality and class imbalance in the dataset, this paper takes a two-pronged approach. Firstly, it introduces the Bald Eagle Search (BES) algorithm to optimize multiple parameters of the Kernel Principal Component Analysis (KPCA) algorithm. This optimization process results in an enhanced version of KPCA (B-KPCA), which effectively eliminates redundant data, reduces dimensionality, and improves dataset usability. Secondly, the paper integrates the concept of cost sensitivity with the Random Forest algorithm, enhancing sensitivity towards minority classes. This combination not only reduces the training time of the detection model but also enhances the accuracy of detecting intrusion

behavior within the minority class. By adopting these two strategies, the paper aims to create a more efficient and effective intrusion detection system, capable of handling the challenges posed by high-dimensional and imbalanced datasets.

Table 1: Information in UNSW-NB15 training and testing.

UNSW-NB15		
	Training set	**Testing set**
Normal	56000	37000
Anallysis	2000	677
Backdoor	1746	583
Dos	12264	4089
Exploits	33393	11132
Fuzzers	18184	6062
Generic	40000	18871
Reconnaissance	10491	3496
Shellcode	1133	378
Worms	130	44
Total	175341	82332

Selection of Evaluation Metrics

In the context of classification problems, classification outcomes can be categorized into two main results: correct or incorrect. These outcomes can be further classified into four distinct scenarios, as outlined in Table 2: TP (True Positives): The model correctly detects attack traffic.

Table 2: Confusion Matrix.

Status	Judged As an Attack	Judged As a Norm
Attack traffic	TP	FP
Normal flow	FN	TN

FN (False Negatives): The model fails to detect attack traffic, misclassifying it as normal traffic.

TN (True Negatives): The model correctly identifies normal traffic.

FP (False Positives): The model incorrectly identifies normal traffic as an attack.

FP and FN are typically referred to as "false alarms."

Based on these four parameters, one can derive four key metrics to assess the practical performance of a model.

samples correctly predicted by the model to the sum of all samples.

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Sensitivity: refers to the model's ability to correctly identify positive example samples.

$$sensitivity = \frac{TP}{(TP + FN)}$$

Specificity: refers to the ability of the model to correctly identify negative example samples.

$$specificity = \frac{TN}{(FP + TN)}$$

G-mean: an assessment metric that combines sensitivity and specificity, taking into account the classification accuracy of both positive and negative examples, which is more sensitive to the classification effect of a few classes, and it is the geometric mean of sensitivity and specificity.

$$G - mean = \sqrt{sensiticity \times specificity}$$

Data set preprocessing

Unique Thermal Encoding of Character-based Features:

Used to convert non-numeric character-based features into a numeric form that can be processed by a computer. The discrete feature values are extended into Euclidean space for data feature encoding such that each possible feature value corresponds to

a new binary variable. The UNSW_NB15 dataset is uniquely hot coded, where the features in columns 3,4,5 are character-based (corresponding to "proto" "service" "state"), "proto" "service" "state" and "state". The features in columns 3,4,5 are character-based (corresponding to "proto", "service", "state"), and "proto" is mapped to 131-dimensional features, "service" is mapped to 12-dimensional features, "state" is mapped to 7-dimensional features, and the feature data are mapped to 7-dimensional features. is mapped to 7 dimensions, "scrip" and "dstip" columns in the feature dataset represent the IP address, since determining whether the data is abnormal has nothing to do with the IP address, the first column ID is only an identifier, to simplify the removal of these three columns, the dimension is increased from 43 dimensions to 185 dimensions after the preprocessing. The dimensions are increased from 43 to 187 dimensions.

Feature data normalization:

A technique that converts the entire range of values of a set of features into a predetermined range. Usually, there is a huge difference in the range of data values between different features, which can cause the training process of machine learning algorithms to be affected, and features with a larger range of values will be given more weight in the training of the algorithm. Therefore, min-max normalization is introduced to map all the data to the interval [0, 1], thus speeding up the convergence of the model and improving the accuracy of the classification results, which is given by the formula:

$$X' = \frac{X - Min}{Max - Min}$$

Where is the minimum value of the feature, is the maximum value of the feature and is the feature value after normalization.

Dimension reduction of feature data

In this section, KPCA optimized based on Condor search algorithm is used to reduce the dimensionality of the preprocessed data set. Firstly, the size of condor population $npop$ is set to 50, and the number of iterations $Maxlt$ is set to 200, and the kernel parameters in KPCA algorithm are optimized. According to fitness function , the type of kernel function is Gaussian kernel function, and the optimal value of parameter γ is 0.007. Set the kernel parameter to the optimal value. The cumulative contribution rate of feature for data dimensionality reduction is set at 95%. Figure 3 shows the cumulative contribution rate of the first 15 feature vectors after data dimensionality reduction. According to the data in the figure, after the algorithm completed the dimensionality reduction of the experimental data set, the cumulative contribution rate of the first 12 features reached 96.26%, meeting the threshold of 95% contribution rate.

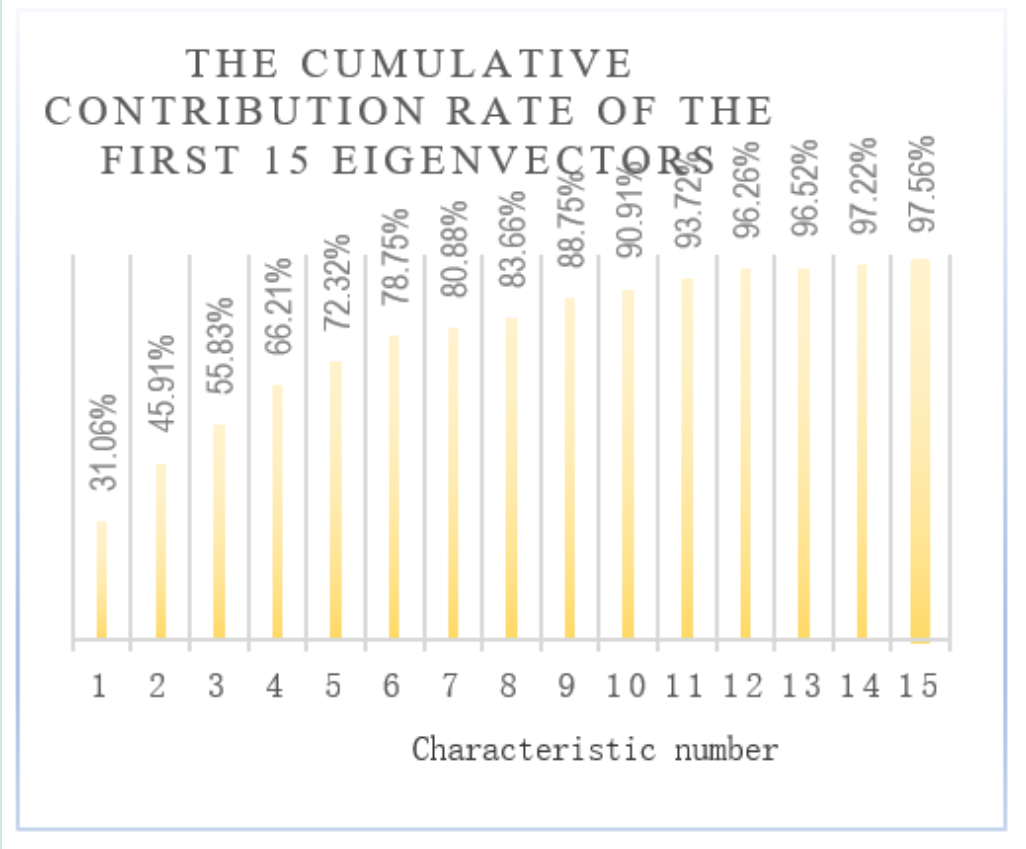

Figure 3: Feature cumulative contribution rate.

Experiment and result analysis

In accordance with the data dimensionality reduction algorithm and cost-sensitive random forest algorithm proposed in this paper, experimental validation is carried out, and in order to more scientifically and accurately assess the performance of the algorithms on high-dimensional and category-unbalanced data, validation is carried out based on the evaluation indexes of accuracy, sensitivity and specificity, and G-mean value. First, in order to evaluate the performance of different data dimensionality reduction algorithms in intrusion detection, three commonly used algorithms, namely, principal component analysis algorithm (PCA), isometric feature mapping algorithm (ISOMAP) [32], and local linear embedding algorithm (LLE) [33], are selected as the reference terms to be compared with the data dimensionality reduction algorithm improved in this paper (B-KPCA), and the four methods are used to evaluate the data dimensionality of the preprocessed UNSW- NB15 dataset's data dimensionality down to 15 dimensions, and use the random forest algorithm to classify and identify tasks on the downgraded data, and the results are shown in Table 3. From the data comparison results, it can be clearly observed that after the data dimensionality reduction using the B-KPCA algorithm, the accuracy of the classification test is significantly better than the other three algorithms. Analyzing the dimension reduction time, compared with PCA algorithm, B-KPCA algorithm only takes 3.69 seconds more, but compared with ISOMAP algorithm and LLE algorithm, it saves 20.68 seconds and 16.58 seconds of running time, respectively. Considering the accuracy and running time together, it can be concluded that the B-KPCA algorithm outperforms the other three compared algorithms in terms of performance.

Table 3: Dimensionality reduction algorithm comparison results.

	B-KPCA	PCA	ISOMAP	LLE
Accuracy after dimensionality reduction /%	97.25	90.21	87.42	93.27
Dimensionality reduction time /s	9.16	5.47	29.84	25.74

Secondly, in this paper, the original RF algorithm [34], the RF algorithm sensitive based on the unbalanced proportion method [35], the RF algorithm sensitive based on the Youden threshold method [36], and the RF algorithm sensitive based on the curve intersection [37] are selected for the comparative experiments to test the performance of the various cost-sensitive RF algorithms (Table 4 & Figure 4). According to the above comparison experiments, it can be seen that the RF of the unbalanced proportion method still does not take into account the cost weights of the categories in the majority voting stage, and does not differentiate between the different categories to deal with the Youden threshold method needs to set the appropriate threshold according to the specific problem, and its selection may be affected by subjective factors and lacks a certain degree of objectivity, and at the same time the results indicate that the algorithm proposed in this paper compared to the original model in terms of accuracy is improved by 5.59%, 11.7% in specificity, 0.0558 in G-mean value, and 0.91% in sensitivity because the original model is more inclined to the classification of the majority of the categories in the classification, and is ineffective in detecting the unbalanced minority of the categories, resulting in a large data difference in sensitivity and specificity, while the model in this paper can effectively and correctly identify the minority of the categories. Therefore, the improvement of data degradation and random forest can obtain better misclassification cost of the prediction target model and can have more efficient classification performance when facing high dimensionality and class imbalanced data.

Figure 4: Classification and comparison of multiple algorithms.

Table 4: Classification and comparison of multiple algorithms.

Algorithm	Acc/%	Sen/%	Spe/%	G-mean	Training Time	Test Time
Traditional RF model	93.11	99.3	87.51	0.9322	23.89s	0.87s
Unbalanced ratio method -RF	97.47	98.05	96.55	0.973	27.46s	1.24s
Youden Threshold method -RF	97.37	97.39	97.34	0.9736	25.92s	1.28s
Textual algorithm	98.7	98.39	99.21	0.988	12.57s	0.25s

Conclusion

For the intrusion detection in the high dimensionality of data and sample class imbalance caused by the traditional random forest computational complexity, high consumption of storage resources and low classification accuracy, this paper proposes an intrusion detection model based on the improvement of the random forest algorithm, the use of vulture search algorithm optimized KPCA to complete the data dimensionality reduction, the introduction of cost-sensitive learning methods to the random forest.

1. Through experiments, it is verified that the method proposed in this paper has better performance compared with the traditional method, with an improvement of 5.59% in accuracy, 11.7% in specificity, and 0.0558 in G-mean value, and it can complete efficient detection for a few categories of samples in a shorter period of time under the premise of higher accuracy.

2. However, some limitations of the model were realized during the experimental process. The evaluation of our study was based on publicly available datasets and was not tested in a real-world environment. Therefore, future work may focus on real-time environment testing, and on the basis of evaluating the model's performance in a laboratory environment, the model will be tested and validated in an actual real-time environment, providing more realistic contexts and data to further validate the model's reliability and practicality.

References

1. Chris Florackis, Christodoulos Louca, Roni Michaely, Michael Weber (2022) Cybersecurity Risk. The Review of Financial Studie 36: 351-407.
2. David Rios Insua, Aitor Couce Vieira, Jose A Rubio, Wolter Pieters, Katsiaryna Labunets, et al. (2019) An Adversarial Risk Analysis Framework for Cybersecurity. Risk Analysis 41: 6-36.
3. Ryan Mills, Angelos K Marnerides, Matthew Broadbent, Nicholas Race (2021) Practical Intrusion Detection of Emerging Threats. IEEE Transactions on Network and Service Management 19: 582-600.
4. Elijah M Maseno, Zenghui Wang, Hongyan Xing (2022) A Systematic Review on Hybrid Intrusion Detection System. Security and Communication Networks.
5. Hawkar Kh Shaikha, Wafaa M Abduallah (2017) A Review of Intrusion Detection Systems. Academic Journal of Nawroz University 6(3): 101-105.
6. Hari Om, Aritra Kundu (2012) A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. The Review of Financial Studies 131-136.
7. Liu Zhi, Ning Wei, Fu Xianya, Zhang Mengmeng, Wang Yuhao (2020) Fast Intra-Mode Decision Algorithm for Virtual Reality 360 Degree Video Based on Decision Tree and Texture Direction. Twelfth International Conference on Digital Image Processing (ICDIP 2020) pp. 11519.
8. Reising Donald, Cancelleri Joseph, Loveless T Daniel, Kandah Farah, Skjellum Anthony (2021) Radio Identity Verification-Based IoT Security Using RF-DNA Fingerprints and SVM. IEEE Internet Of Things Journal 8(10): 8356-8371.
9. Qingqing Han, Jingmei Liu, Zhiwei Shen, Jingwei Liu, Fengkui Gong (2020) Vector partitioning quantization utilizing K-means clustering for physical layer secret key generation. Information Sciences 512: 137-160.
10. Chih Yu Hsu, Shuai Wang, Yu Qiao (2021) Intrusion detection by machine learning for multimedia platform. Multimedia Tools and Applications 80(19): 29643-29656.
11. Chunying Zhang, Donghao Jia, Liya Wang, Wenjie Wang, Fengchun Liu, et al. (2022) Comparative research on network intrusion detection methods based on machine learning. Computers & Security pp. 121.
12. Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, Andreas Hotho (2019) A survey of network-based intrusion detection data sets. Journal of Big Data 86: 147-167.
13. Sikha Bagui, Kunqi Li (2021) Resampling imbalanced data for network intrusion detection datasets. The Review of Financial Studies 8: 351-407.
14. Zhen Yang, Xiaodong Liu, Tong Li, Di Wu, Jinjiang Wang, et al. (2022) A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Computers & Security 116: 102675.
15. Maryam Yousefnezhad, Javad Hamidzadeh, Mohammad Aliannejadi (2021) Ensemble classification for intrusion detection via feature extraction based on deep Learning. Soft Computing 25: 12667-12683.
16. Eduardo Laber, Lucas Murtinho (2019) Minimization of Gini Impurity: NP-completeness and Approximation Algorithm via Connections with the k-means Problem. Electronic Notes in Theoretical Computer Science 346: 567-576.
17. Hoang Uyen N, Mojdeh Mirmomen S, Meirelles Osorio, Yao Jianhua, Merino Maria, et al. (2018) Assessment of multiphasic contrast-enhanced MR textures in differentiating small renal mass subtypes. Abdominal radiology 43(12): 3400-3409.
18. Dibyajyoti Chutia, Dhruba Kumar Bhattacharyya, Jaganath Sarma, Penumetcha Narasa Lakshmi Raju (2017) An effective ensemble classification framework using random forests and a correlation based feature selection technique. Transactions in GIS 21(6): 1165-1178.
19. Amit Kumar Mishra, Shweta Paliwal (2022) Mitigating cyber threats through integration of feature selection and stacking ensemble learning: the LGBM and random forest intrusion detection perspective. Cluster Computing 26(4): 2339-2350.
20. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, et al. (2017) Feature Selection: A Data Perspective. ACM Computing Surveys 50(6): 1-45.
21. Wanfu Gao, Liang Hu, Ping Zhang, Jialong He (2018) Feature selection considering the composition of feature relevancy. Pattern Recognition Letters 112: 70-74.

22. Thippa Reddy G, Praveen Kumar Reddy M, Kuruva Lakshmanna, Rajesh Kaluri, Dharmendra Singh Rajput (2020) Analysis of Dimensionality Reduction Techniques on Big Data. Journals & Magazines 8: 54776 - 54788.
23. Hongpo Zhang, Lulu Huang, Chase Q Wu c, Zhanbo Li (2020) An Effective Convolutional Neural Network Based on SMOTE and Gaussian Mixture Model for Intrusion Detection in Imbalanced Dataset. Computer Networks 177: 107315.
24. Yanan Li, Tao Qin, Tao Qin, Yongzhong Huang, Jinghong Lan, ZanHao Liang (2022) HDFEF: A hierarchical and dynamic feature extraction framework for intrusion detection systems. Computers & Security 121: 102842.
25. Yun Chun Wang, Ching Hsue Cheng (2021) A multiple combined method for rebalancing medical data with class imbalances. Computers in Biology and Medicine 134: 105527.
26. Herrera Semenets Vitali, Bustio Martínez Lázaro, Hernández León Raudel, van den Berg Jan (2021) A multi-measure feature selection algorithm for efficacious intrusion detection. Knowledge Based Systems 227: 107264.
27. Guangjie Han, Xun Li, Jinfang Jiang, Lei Shu, Jaime Lloret (2015) Intrusion Detection Algorithm Based on Neighbor Information Against Sinkhole Attack in Wireless Sensor Networks. The Computer Journal 58(6): 1280-1292.
28. Li D, He X, Dai X (2016) Improved kernel principal component analysis algorithm for network intrusion detection. ICIC Express Letters 10(4): 971-975.
29. Alaa A Zaky, Rania M Ghoniem, Selim F (2023) Precise Modeling of Proton Exchange Membrane Fuel Cell Using the Modified Bald Eagle Optimization Algorithm. Sustainability 15(13).
30. Benedetto Marco Serinelli, Anastasija Collen, Niels Alexander Nijdam (2020) Training Guidance with KDD Cup 1999 and NSL-KDD Data Sets of ANIDINR: Anomaly-Based Network Intrusion Detection System. Procedia Computer Science 175: 560-565.
31. Sankalp Jain, Eleni Kotsampasakou, Gerhard F Ecker (2018) Comparing the performance of meta-classifiers-a case study on selected imbalanced data sets relevant for prediction of liver toxicity. Journal of Computer Aided Molecular Design 32: 583-590.
32. Zhenhua Huang, Xin Xu, Lei Zuo (2014) Reinforcement learning with automatic basis construction based on isometric feature mapping. Information Sciences 286: 209-227.
33. Li Mingai, Luo Xinyong, Yang Jinfu, Sun Yanjun (2016) Applying a Locally Linear Embedding Algorithm for Feature Extraction and Visualization of MI-EEG. Journal of Sensors 2016(2): 1-9.
34. Mogollón Gutiérrez Óscar, Núñez José Carlos Sancho, Vegas Mar Ávila, Lindo Andrés Caro (2023) A Novel Ensemble Learning System for Cyberattack Classification. Intelligent Automation & Soft Computing 37(2): 1691-1709.
35. Fang X, Zhang H, Gao S, Tan Y (2015) Imbalanced web spam classification based on nested rotation forest. ICIC Express Letters 9(3): 937-944
36. Tahani Coolen Maturi, Frank PA Coolen, Manal Alabdulhadi (2020) Nonparametric predictive inference for diagnostic test thresholds. Communications in Statistics Theory and Methods 49(3): 697-725.
37. Pradhan Biswajeet, Sameen Maher Ibrahim, AlNajjar Husam AH, Sheng Daichao, Alamri Abdullah M, et al. (2021) A Meta-Learning Approach of Optimisation for Spatial Prediction of Landslides. Remote Sensing 13(22): 4521-4521.

DOI: 10.32474/ARME.2023.04.000183