Academic Journal of Computing & Information Science
ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

Feature Interaction Based Feature Selection Algorithm for In-trusion Detection

Yimeng Wang[1,a], Zongpu Jia[1,b], Xiaoyan Pang[1,c], Shan Zhao[1,d,*]

[1]*School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan, 454000, China*
[a]*212109020013@home.hpu.edu.cn*, [b]*jiazp@hpu.edu.cn*, [c]*pangxy@hpu.edu.cn*, [d]*zhaoshan@hpu.edu.cn*
[*]*Corresponding author*

Abstract: *Fog computing facilitates the placement of data at the network's edge for processing, which effectively reduces energy consumption and enhances efficiency. However, the limited resources inherent in fog computing render it vulnerable to extensive volumes of high-dimensional anomalous traffic. This study proposes a novel feature selection algorithm called filtered interaction maximum relevance minimum redundancy, which incorporates feature interaction to enable effective intrusion detection in fog computing. Through feature selection, the algorithm downscales the high-dimensional data captured in the fog nodes to reduce redundant features. The experimental results show that the parsimonious feature set obtained using the algorithm in this paper improves the classification accuracy while reducing the execution time compared to the original dataset.*

Keywords: *Feature Selection; Fog Computing; Intrusion Detection; Feature Interaction; Machine Learning*

1. Introduction

The emergence of the Internet of Things (IoT) paradigm has gained wide adoption, facilitating data sharing among interconnected computing devices and sensors over the Internet. This seamless connectivity aims to address diverse challenges and offers new services without requiring human intervention. However, the inherent vulnerabilities of IoT networks, coupled with limitations in hardware properties, expose them to security threats, which increases the risk of attacks [1]. One of the key issues is the heterogeneous and distributed nature of IoT networks, which makes it challenging to deploy previous security mechanisms in a distributed IoT environment. This includes resource scarcity, high latency, high bandwidth consumption, and degradation of quality of service. To address these challenges, distributed learning methods based on fog computing have proven more effective [2].

Fog computing, an extension of cloud computing, focuses on managing data from sensors and edge devices. It decentralizes data, data processing and applications in devices at the edge of the network rather than relying entirely on cloud data centers. It extends cloud services to the IoT edge to minimize data transfer overhead and save processing time and communication resources [3]. This concept was proposed to address the challenges in IoT applications that require low latency, geographic remoteness, and high mobility [4]. However, most end devices in fog nodes, such as smart appliances, smartphones, and VR devices, are resource-constrained. Networks with these characteristics are susceptible to threats such as denial of service, man-in-the-middle, malicious gateways, privacy leakage, and service manipulation. Integrating an intrusion detection system (IDS) into fog computing infrastructures can effectively mitigate these security threats [5].

The concept of IDS originated in April 1980 and evolved into intrusion detection ex-pert systems (IDES) in the mid-1980s. By 1990, IDS was further divided into network-based IDS and host-based IDS. The network intrusion detection system (NIDS) is a com-monly used tool for detecting network intrusions by collecting data on the current network operational status and analyzing network traffic using the system's pre-built algorithms and historical experience [6]. Intrusion detection systems can be categorized into misuse-based intrusion detection systems (MIDS) and anomaly-based network intrusion detection systems (AIDS) based on their engine detection mechanism. The MIDS is based on the de-tection of known signatures, and it is effective in identifying attacks in the signature base. However, the MIDS struggles with identifying unknown and variant attacks, such as zero-day attacks, resulting in a lower overall detection rate. The AIDS is used to classify traffic by learning the network traffic behavior and offers flexibility, robustness, and scalability in detecting unknown attacks, which makes it suitable for dynamic intrusion detection systems [7]. Also, the application of machine learning (ML) in intrusion detection systems further enhances optimal and accurate recognition results.

However, the limited resources in fog computing make it vulnerable to an extensive volume of high-dimensional anomalous traffic. Traditional IDS methods, when applied to process multi-featured data, not only consume time but also lack accuracy. This study proposes a maximum correlation minimum redundancy feature selection algorithm (FI-mRMR) that incorporates feature interaction for effective intrusion detection in fog compu-ting. Through feature selection, the high dimensional data captured in the fog node is downscaled to reduce the redundant features.

The main contributions of this study are as follows:

(1) Previous algorithms have primarily focused on correlation and redundancy. Howev-er, the proposed FI-mRMR feature selection method considers not only the maximum correlation and minimum redundancy but also the interaction between features, which enhances the traditional approach;

(2) Experiments conducted on the NSL-KDD and CICIDS-2017 dataset show that the FI-mRMR greatly outperforms existing algorithms such as mRMR, MRI, CCMI and GFS in terms of classification precision and accuracy;

(3) The performance of different classifiers was evaluated on the NSL-KDD dataset using the filter feature selection method.

2. Related Works

Feature selection can be defined as the process of removing irrelevant and redundant features to enhance the efficiency of models [8-10]. The filtering-based approach deter-mines the importance of the features by scoring and ranking them based on their score size. Classical feature selection algorithms, such as information gain (IG) and mutual in-formation maximization (MIM), remove irrelevant features using mutual information be-tween the features and class labels. While these methods are simple and fast, they often ignore redundancy between features.

Priscilla [11] et al. proposed a two-stage feature selection method using mutual in-formation in the first stage and recursive feature elimination (RFE) in the second stage to eliminate redundant features. Pashaei [12] et al. used minimum redundancy maximum relevance (mRMR) as a first-level filter and then introduced simulated annealing and crossover operators into a binary arithmetic optimization algorithm to select the mini-mum set of informative genes. To address the limitations in mutual information, Zhou [13] introduced the maximum mutual information coefficient to measure the correlation and redundancy between features and labels. Qing [14] et al. proposed a Correlation and Conditional Mutual Information (CCMI) algorithm that combines two components: the im-proved Pearson correlation coefficient and the improved conditional mutual information measure. Wang [15] et al. proposed a Max-Relevance and Max-Independence (MRI) algorithm. They assembled newly provided and retained information that is negatively correlated with redundant information. In the new terminology, redundant and new infor-mation are properly harmonized and treated equally.

Nguyen et al. also used an improved feature selection algorithm based on mRMR, Generic Feature-Selection (GFS), and they considered the combination of the feature correlation feature selection (CFS) metric with the mRMR algorithm. Whereas in this paper, conditional interaction of features is realized through conditional mutual information by considering feature relevance, i.e., changing one feature based on the value of another feature [16].

In the context of NIDS, Wang [17] et al. proposed an optimized neural network hyperparameter using an improved particle swarm optimization algorithm with a loss function as population localization. Once the optimal parameters were obtained, a scaled convolutional neural network was constructed, and the model was trained through back-propagation. Saksham Mittal [18] et al. applied supervised class machine learning algorithms to classify different types of attacks using four mathematical models on datasets CICIDS2017 and BotIot. Ananthi [19] et al. used the RFE algorithm for feature selection on dataset KDD99 and deployed a deep neural network for binary classification after selecting the necessary features through RFE.

Most of the existing studies of feature selection methods consider correlation and redundancy factors; they often focus on dependencies between individual features and the target class. However, a feature may exhibit an average correlation with the target class but may lose relevance when interacting with

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

other features. Therefore, this study proposes an FI-mRMR algorithm that considers feature interactions for a more comprehensive feature selection approach.

3. FI-mRMR Algorithm Design

In the feature selection process, the mRMR (max-relevance and min-redundancy) algorithm plays an important role. It operates on the principle of identifying the features in the original feature set that exhibit the highest relevance to the final output (max-relevance) while maintaining the least relevance among the features (min-redundancy). The objective of feature selection is to identify a subset S of features with m features that exhibit maxi-mum dependence on the target classification c. The formula can be expressed as follows:

$$\max D(S,c), D = I(x_i, i=1,...,m; c) \tag{1}$$

The maximum correlation formula can be expressed as follows:

$$\max D(S,c), D = \frac{1}{|S|}\sum_{x_i \in S} I(x_i;c) \tag{2}$$

The minimum redundancy formula can be expressed as follows:

$$\min \mathrm{R}(S), R = \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i;x_j) \tag{3}$$

where X_i is the i_{th} feature, c is a category variable, and S is a subset of features.

Combining maximum relevance D with minimum redundancy R results in the mRMR algorithm defined by the operator Φ(D, R). This can be expressed in the simplest additive integration method as follows:

$$\max \Phi(D,R), \Phi = D - R \tag{4}$$

In practice, incremental search methods are used to identify near-optimal features. Assuming an existing feature set Sm-1, the goal is to find the m_{th} feature from the remaining features X-Sm-1 and maximize Φ(D, R) through feature selection. The incremental algorithm optimization formula can be expressed as follows:

$$mRMR = \max_{x_{j \in X - S_{m-1}}} [I(x_j;c) - \frac{1}{m-1}\sum_{x_i \in S_{m-1}} I(x_j;x_i)] \tag{5}$$

While the mRMR algorithm only considers correlation and redundancy, the feature set obtained through the maximum correlation formula depicts the dependency between individual and target features. However, some individual features may only exhibit aver-age or lower dependencies. If these features are combined, i.e., after feature interaction, a high dependency on the target feature is created. By introducing feature interaction, the proposed algorithm identifies redundant features by calculating the degree of interaction between features through conditional mutual information. Then, the combined redundant features are filtered again using the minimum redundancy formula.

In this study, the proposed algorithm extends the mRMR algorithm by introducing feature interaction, hence named feature interaction max-relevance and min-redundancy (FI-mRMR). In the FI-mRMR algorithm, let $X=x_1,x_2,...,x_m$ be the feature set of the dataset K with n instances. The algorithm aims to generate a subset F of H features, where H≤m and F⊆X. As shown in Equation (6), the algorithm was initiated using the maximum relevance formula to obtain a feature set Fmax, which is an ordered set of mutual information be-tween the input and target features from high to low. The Fmax set considers only the correlation between individual features and the target feature but not the interactions between features. The formula can be expressed as follows:

$$F_{\max} = \max D(F_{\max}, c), D = \frac{1}{|F_{\max}|}\sum_{x_i \in X} I(x_i;c) \tag{6}$$

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

Next, the feature-feature interactions in Fmax are computed through conditional mutual information. By subtracting the conditional mutual information from the mutual in-formation, a feature set F_{fi}, representing feature interactions, was obtained. In this case, a positive interaction indicates that the dependency of the interacting feature on the target feature is higher than the dependency of the individual feature on the target feature, and vice versa for negative interactions. Thus, F_{fi} is a set of positively interacting feature sets and can be expressed as follows:

$$F_{fi} = F_{fi} \cup \{x_i, x_j\}, (x_i, x_j) \in F_{\max} \tag{7}$$

$$I(x_i; c \mid x_j) - I(x_i; c) > 0 \tag{8}$$

Finally, the algorithm filters redundant features in F_{fi} using the minimum redundancy formula to obtain the feature set F. The formula can be expressed as follows:

$$F = \min R(F), R = \frac{1}{|F|^2} \sum_{x_i, x_j \in X} I(x_i; x_j) \tag{9}$$

The FI-mRMR algorithm can be written as follows:

FI-mRMR Algorithm

Input:Feature set X={x1,...,xm},class labels C={y1,...,yn},number of features to be selected H,H≤m

Output: Selected feature subset F, $F \subseteq X$

```
//Load processed dataset
train_df = pd.read_csv("process_train.csv")
train_df = train_df.sample(2000)//sample
train_df = train_df.astype(int)//Converting data types to integers
train_x = train_df.drop(['labels'], axis=1)
train_y = train_df['labels'].values
//Feature selection
features_num = train_x.shape[1]//Number of features
selected_features = set()//Using collections to avoid duplicate features
//Calculate and rank the mutual information between each feature and the target variable
mi_list = mutual_info_classif(train_x, train_y)
mi_indices_sorted = np.argsort(mi_list)[::-1]
Ffi = set()
F = set()
//      Select top features_num/2 features as Ffi based on mutual information
for index in mi_indices_sorted[:features_num // 2]:
    Ffi.add(index)
//Compute conditional mutual information for each feature and select
for i in Ffi:
    for j in Ffi:
        if j <= i:// Avoiding double counting
            continue
        mic = drv.information_mutual_conditional(train_x.iloc[:, i].values, train_x.iloc[:, j].values, train_y)
        if mic > mi_list[i] and mic > mi_list[j]:
            F.add(j)
if not F:
    F.add(mi_indices_sorted[0])
print("Selected features based on mutual information and conditional mutual information:")
print(sorted(list(F)))
```

4. Experimental Results and Analysis

4.1. Experimental Equipment

The experiments were conducted using a laptop (Model: LAPTOP-6QBDGDR4) equipped with an

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

AMD Ryzen 5 5600H processor @ 3.30 GHz (single processor), 16GB RAM, 64-bit operating system, and 512GB hard disk. Feature selection was executed using PyCharm and Python 3.6, while graphs were generated using Origin.

4.2. Dataset and IDS Model

4.2.1. Dataset and Data Preprocessing

The NSL-KDD [20, 21] dataset is a widely used intrusion detection benchmark that represents a revised version of the original KDDCUP99 dataset [22]. It addresses the limitations of KDDCUP99 by eliminating redundant records, rationalizing the number of in-stances, and maintaining the diversity of samples [23]. Each record in the dataset contains 43 features, with 41 pertaining to the traffic input and the last two denoting labels (normal or attack) and scores (severity of the traffic input itself). Tables 1 and 2 present the types of attack and data distribution for the NSL-KDD dataset.

The presence of redundant features not only affects the training results but also re-duces the training speed of the model. Therefore, it is necessary to downscale the irrelevant dimensions of the original dataset [24]. At the same time, filtering out the features that have less impact on the results can effectively reduce computational overhead and prevent the interference of irrelevant features. Upon the application of the feature selection method proposed in this study, the 41 features in the NSL-KDD dataset were reduced to approximately 13, as shown in Table 3. Among them, DOS attack refers to making the tar-get network hosts or applications inaccessible or unusable; Probe attack refers to probing the target network, hosts, or applications to obtain information about their topology, ser-vices, or vulnerabilities; U2R attack refers to obtaining super-user access to local hosts by exploiting the vulnerabilities of the target hosts; R2L attack refers to accessing the information and resources of the attacked system from outside the network by taking ad-vantage of the deficiencies of the network security mechanism.

The CIC-IDS-2017 dataset [25, 26], a collaborative project between the Communica-tions Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC), evaluates 11 datasets that have been available since 1998 and shows that most of them (e.g., the classic KDDCUP99, NSLKDD, etc.) are outdated and unreliable. Some of these datasets lack traffic diversity and capacity, some do not cover a wide range of known attacks, while others anonymize packet payload data, which does not reflect current trends. Some also lack feature sets and metadata.

It has data collected up to 5 p.m. on Friday, July 7, 2017, for a total of five days. Mon-day was a normal day and includes only normal traffic. The realized attacks include Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attacks, Exfiltration, Botnets, and DDoS. They were executed on Tuesday, Wednesday, Thursday, and Friday mornings and afternoons, respectively, as shown in Table 4. Among them GoldenEye, Slowloris, hulk, Slow-HTTPTest, LOIC, HOIC are security testing tools used to simulate Dos attacks. The number of features in CIC-IDS-2017 dataset after using the algorithm of this paper is reduced from 80 to 10 as shown in Table 5.

Table 1: Types of attacks on the NSL-KDD dataset.

Classes:	DoS	Probe	U2R	R2L
Sub-Classes:	apache2	ipsweep	Buffer_overflow	ftp_write
	back	mscan	loadmodule	guess_passwd
	Land	nmap	perl	httptunnel
	neptune	portsweep	ps	imap
	mailbomb	saint	rootkit	multihop
	pod	satan	sqlattack	named
	processtable		xterm	phf
	smurf			sendmail
	teardrop			Snmpgetattack
	udpstorm			Spy
	worm			snmpguess
				warezclient
				warezmaster
				xlock
				xsnoop
Total:	11	6	7	15

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

Table 2: Data distribution of the NSL-KDD dataset.

Dataset	Number of Records:					
	Total	Normal	DoS	Probe	U2R	R2L
KDDTrain+20%	25192	13449(53%)	9234(37%)	2289(9.16%)	11(0.04%)	209(0.8%)
KDDTrain+	125973	67343(53%)	45927(37%)	11656(9.11%)	52(0.04%)	995(0.85%)
KDDTest+	22544	9711(43%)	7458(33%)	2421(11%)	200(0.9%)	2654(12.1%)

Table 3: Dataset after feature selection (NSL-KDD).

Feature Name:	Description	Type	Value Type
Duration	Length of time duration of the connection	Continuous	Integers
Protocol-Type	Protocol used in the connection	Categorical	Strings
Service	Destination network service used	Categorical	Strings
Flag	Status of the connection – Normal or Error	Categorical	Strings
Src-Bytes	Number of data bytes transferred from source to destination in single connection	Continuous	Integers
Dst-Bytes	Number of data bytes transferred from destination to source in single connection	Continuous	Integers
Land	If source and destination IP addresses and port numbers are equal then, this variable takes value 1 else 0	Binary	Integers
Wrong-Fragment	Total number of wrong fragments in this connection	Discrete	Integers
Hot	Number of “hot” indicators in the content such as: entering a system directory, creating programs and executing programs	Continuous	Integers
Num-Failed-Logins	Count of failed login attempts	Continuous	Integers
Logged-In	Login Status : 1 if successfully logged in; 0 otherwise	Binary	Integers
Num-Compromised	Number of “compromised” conditions	Continuous	Integers
Is-Guest-Login	1 if the login is a “guest’’ login; 0 otherwise	Binary	Integers

Data preprocessing stands as the most time-consuming and fundamental step in da-ta mining, considering that real data often originates from different platforms and may exhibit noise, redundancy, incompleteness, and inconsistency [27]. Therefore, it is important to convert the raw data into a format suitable for analysis. The preprocessing steps include data filtering, data numericalization, and data discretization.

(1) Data filtering: Given the heterogeneous nature of the platform, the raw data inevitably contains anomalies and redundant instances that can negatively affect classification accuracy. To address this issue, it is important to remove these records from the dataset before the commencement of experimentation. We can achieve the purpose of data filtering by removing unwanted content such as labels, special symbols, numbers, etc. from the data through techniques such as regular expressions, string matching and filtering;

(2) Data numericalization: Eliminating differences in data scale and size is essential to ensure comparison occurs under uniform scales or orders of magnitude. Numericalization ensures that data with larger values do not disproportionately influence the model’s convergence in machine learning. This makes numercalization essential in handling data with different attributes on a single platform that contains both numeric and non-numeric values [28]. For instance, features such as "protocol type", "flag", and "service" in the NSL-KDD dataset are non-numeric. Through numercalization, the non-numeric features were transformed using a unique thermal encoding, which converts the original 41-dimensional features of the NSL-KDD dataset into 122-dimensional features [29]. We can use min-max normalization to numericalize the data. For each attribute, let minA and maxA be the minimum and maximum values of attribute A. An original value x of A is mapped to a value x' in the interval [0,1] by min-max normalization with the formula: new data = (original data - minimum value)/(maximum value - minimum

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

value). Numericization of data allows the values of indicators to be at the same order of magnitude, thus facilitating comprehensive analysis and comparison of indicators in different units or orders of magnitude;

(3) Data discretization: This involves mapping a finite number of individuals in an infinite space to a finite space. The process helps to conserve computational resources, improve computational efficiency, and enhance the stability and accuracy of the model [30]. Also, data discretization is essential for continuous data: it converts data value distribution from continuous attributes to discrete attributes, which generally contain two or more value domains [31]. The result of discretization of continuous data can be classified into two categories, classification of continuous data into sets of specific intervals and classification of continuous data into specific classes. We can achieve discretization of continuous data using methods such as quantile method, distance interval method, frequency interval method, clustering method and chi-square filtering. The distribution of the data value domain will change from continuous to discrete attributes after processing.

Table 4: CIC-IDS-2017 Dataset Record Date and Attack Type.

Date of Recording	Type of Attack
Thursday-01-03-2018	Benign, Infiltration(permeability)
Friday-02-03-2018	Benign, Bot(botnet attack)
Wednesday-14-02-2018	Benign, SSH-Bruteforce, FTP-BruteForce, (BruteForce- violent attack)
Thursday-15-02-2018	Benign, DoS-GoldenEye, DoS-Slowloris
Friday-16-02-2018	Benign, DoS attack-hulk, DoS attacks-SlowHTTPTest
Thuesday-20-02-2018	Benign, DDoS attacks-LOIC-HTTP, DDoS-LOIC-UDP
Wednesday-21-02-2018	Benign, DDOS-LOIC-UDP,DDOS-HOIC
Wednesday-21-02-2018	Benign, Brute Force -Web, Brute Force -XSS, SQL Injection
Friday-23-02-2018	Benign, Brute Force -Web, burte Force -XSS, SQL Injection
Wednesday-28-02-2018	Benign, Infiltration

Table 5: Dataset after feature selection (CIC-IDS-2017).

Feature Name:	Description
fl_dur	Flow duration
tot_fw_pk	Number of packets in the positive direction
tot_bw_pk	Number of packets up in reverse
tot_l_fw_pkt	Total forward packet size
fw_pkt_l_max	The maximum size of the package is positive
fw_pkt_l_min	Package in positive upward minimum size
fw_pkt_l_avg	The average size of packets in the forward direction
fw_pkt_l_std	Size of the forward standard deviation of data packets
bw_urg_flag	Number of times the URG flag is set in the reverse packet
bw_hdr_len	The total number of bytes used for backward-oriented packet headers

4.2.2. IDS in Fog Nodes

Figure 1 shows how the modules function in the fog node for feature selection. The working principle can be described as follows:

(1) Attribute extractor: This module is responsible for capturing network traffic, where large amounts of high-dimensional traffic data are transmitted from IoT devices to this module in the fog node. This involves storing traffic information as features that describe the behavior of the ongoing network activities, resulting in a primitive feature dataset;

(2) Feature selection: Feature dimensionality reduction of the original feature set using a feature selection algorithm to obtain a subset of features that are highly correlated with the input target;

(3) Attack classifier: This module is responsible for identifying attack traffic in IoT net-works and performing classifier attack detection on the filtered feature set.

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

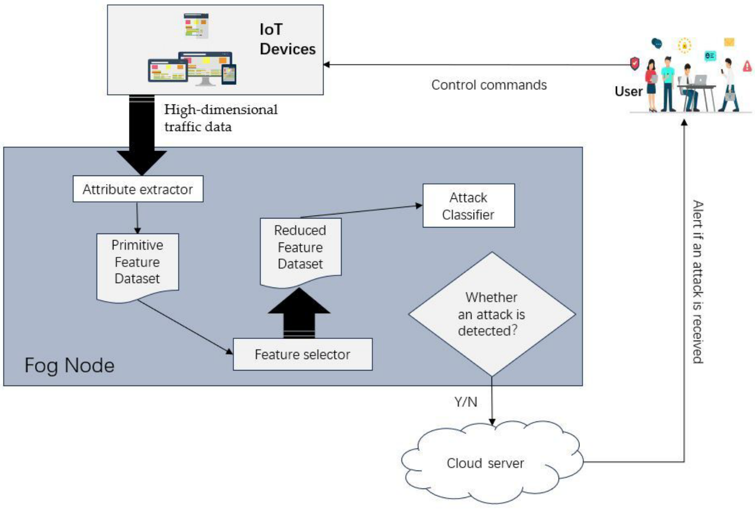

Figure 1: Fog node intrusion detection model.

4.3. Performance Analysis

To evaluate the performance of the IDS using the proposed FI-mRMR feature selection algorithm, a preliminary comparison was conducted between the proposed feature selection method and the original method without feature selection. Tables 6 and 7 show the detection results using the original dataset and the parsimonious dataset on the NSL-KDD dataset, respectively. Tables 8 and 9 present the detection results using the original dataset and the parsimonious dataset on the CIC-IDS-2017 dataset.

Table 6: Detection results for each classifier using the original dataset (NSL-KDD).

Classifier:	Accuracy(%)	Precision(%)	Recall(%)	Response Time (S)
RF	92.35	94.16	93.83	19.34
DT	94	94.8	93.68	10.86
KNN	95.19	95.37	95.12	54.29
XGB	95.35	95.56	95.22	23.77
MLP	96.9	96.93	96.63	37.51

Table 7: The results of each classifier after reduced algorithm were detected(NSL-KDD).

Classifier:	Accuracy(%)	Precision(%)	Recall(%)	Response Time(S)
RF	96.67	95.87	97.61	7.71
DT	96.35	95.95	95.26	3.11
KNN	97.57	96.65	96.55	12.75
XGB	97.61	96.89	96.94	9.79
MLP	97.95	97.36	98	20.36

Table 8: Detection results for each classifier using the original dataset(CIC-IDS-2017).

Classifier:	Accuracy(%)	Precision(%)	Recall(%)	Response Time (S)
RF	96.16	99.47	80.87	1018
DT	98.22	95.05	95.61	194
KNN	95.65	87.9	90.43	1351
XGB	95.88	92.96	93.12	1487
MLP	96.26	95.34	94.81	1550

Table 9: The results of each classifier after reduced algorithm were detected(CIC-IDS-2017).

Classifier:	Accuracy(%)	Precision(%)	Recall(%)	Response Time (S)
RF	96.2	99.56	81.19	447
DT	98.38	95.39	96.84	23
KNN	98.74	96.63	98.35	210
XGB	97.84	94.62	95.12	397
MLP	98.12	98.27	97.97	445

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

The comparative analysis of Tables 6, 7 and 8, 9 shows that the FI-mRMR algorithm pro-posed in this paper selects features with high relevance and low redundancy, which im-proves the accuracy and precision of the evaluation indexes. This proves the effectiveness of the proposed feature selection method. For the dataset NSL-KDD, the number of features is reduced from 41 to 13. For the dataset CICIDS-2017 the number of features is reduced from 80 to 10. The time required for feature selection using the FI-mRMR proposed in this paper is reduced by more than 45% compared to the original method, which greatly re-duces the response time of the classifier and thus reduces the overall time cost. The effectiveness of the algorithm in this paper is demonstrated.

Subsequently, the performance of the FI-mRMR algorithm was compared with three alternative feature selection techniques: mRMR, CCMI, MRI and GFS. The approximated dataset was used as input to the classifier to compare the detection accuracy, precision, and classifier response time of each algorithm across different classifiers.

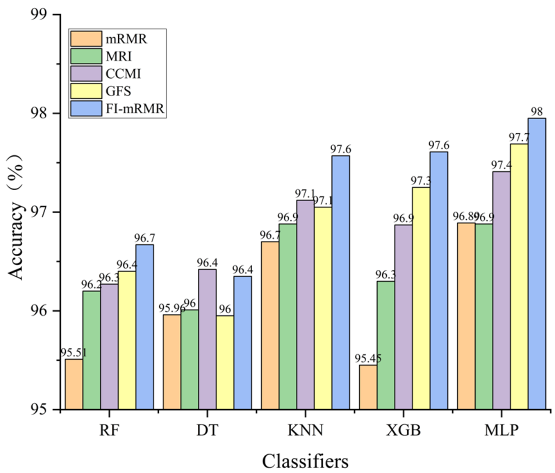

Figure 2: Comparison of the accuracy of the algorithms on different classifiers.

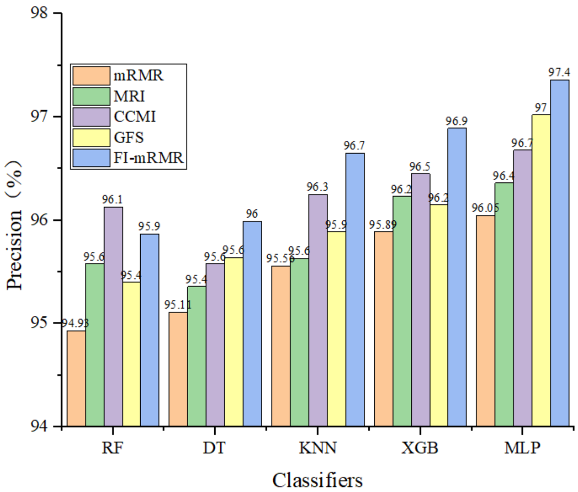

Figure 3: Comparison of the precision of the algorithms on different classifiers.

Figure 2 shows that across the same classifiers, the proposed FI-mRMR algorithm model consistently outperforms other feature selection algorithm models, achieving up to 98% accuracy in the MLP classifier. Additionally, Figure 3 shows that the precision of the FI-mRMR algorithm surpasses that of other algorithms, except for the RF classifier, which is slightly lower than that of the CCMI algorithm. The MLP classifier achieves the highest precision of approximately 97.4%. These comparisons prove the efficiency of the proposedalgorithm and validate the proposed IDS. Therefore, the FI-mRMR algorithm exhibits superior performance compared to other feature selection methods.

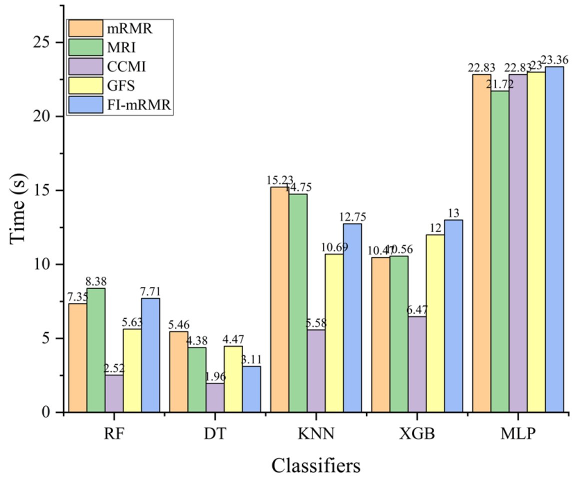

Figure 4: Response time for each classifiers.

Fig. 4 shows that the response time of the proposed algorithm on the first four classifiers is lower than that of the mRMR algorithm and MRI algorithm, but it is still worse compared to the CCMI, GFS algorithms. On the MLP classifier, all algorithms have longer response times, with the FI-mRMR algorithm having the longest response time. Although the FI-mRMR algorithm obtained higher accuracy, further optimization is needed to im-prove the response time.

5. Conclusions

This study proposed a novel FI-mRMR feature selection algorithm that incorporates feature interaction to eliminate the inaccuracies and time-consuming processes involved in detecting intrusion in fog computing. The algorithm considered not only correlation and redundancy but also the combination between features. By integrating the maximum correlation minimum redundancy feature selection with feature interaction into the mRMR algorithalgorithm, intrusion detection in fog computing was achieved, which im-proved the precision and accuracy of the detection. In the future, we will focus on further reducing the classifier response time while maintaining precision and accuracy. Striking a balance between these factors will contribute to the broader applicability and efficiency of the proposed intrusion detection system in real-world fog computing scenarios.

Acknowledgements

This work was supported by the National Natural Science Foundation of China Youth Fund (62202145).

References

[1] ALANI M M. IoTProtect: A Machine-Learning Based IoT Intrusion Detection System [Z]. 2022 6th International Conference on Cryptography, Security and Privacy (CSP). 2022: 61-5.10.1109/csp 55486. 2022.00020

[2] ASULBA B A, SCHUMACHER N, SOUTO P F, et al. Impact of Training Set Size on Resource Usage of Machine Learning Models for IoT Network Intrusion Detection [Z]. 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT). 2023: 330-7.10.1109/DCOSS-IoT58021.2023.00061

[3] SHEN S, HUANG L, ZHOU H, et al. Multistage Signaling Game-Based Optimal Detection Strategies for Suppressing Malware Diffusion in Fog-Cloud-Based IoT Networks [J]. IEEE Internet of Things Journal, 2018, 5(2): 1043-54.

[4] CHISHAKWE S, MOYO N, NDLOVU B M, et al. Intrusion Detection System for IoT environments using Machine Learning Techniques [Z]. 2022 1st Zimbabwe Conference of Information and Communication Technologies (ZCICT). 2022: 1-7.10.1109/zcict55726.2022.10045992

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

[5] DESHMUKH M S, BHALADHARE P R. Intrusion Detection System (DBN-IDS) for IoT using Optimization Enabled Deep Belief Neural Network [Z]. 2021 5th International Conference on Information Systems and Computer Networks (ISCON). 2021: 1-4.10.1109/iscon52037.2021.9702505

[6] FERRAG M A, SHU L, FRIHA O, et al. Cyber Security Intrusion Detection for Agriculture 4.0: Machine Learning-Based Solutions, Datasets, and Future Directions [J]. IEEE/CAA Journal of Automatica Sinica, 2022, 9(3): 407-36.

[7] FU G, LI B, YANG Y, et al. A Multi-Distance Ensemble and Feature Clustering Based Feature Selection Approach for Network Intrusion Detection [Z]. 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI). 2022: 160-4.10.1109/issi55442.2022.9963155

[8] GE Y, LI J, TIAN Y. Internet of Things Intrusion Detection System Based on D-GRU [Z]. 2022 4th International Conference on Applied Machine Learning (ICAML). 2022: 1-6.10.1109/icaml57167. 2022.00066

[9] HATTARKI R, HOUJI S, DHAGE M. Real Time Intrusion Detection System For IoT Networks [Z]. 2021 6th International Conference for Convergence in Technology (I2CT). 2021: 1-5.10. 1109/i2ct51068.2021.9417815

[10] IKHWAN S, PURWANTO P, ROCHIM A F. Comparison Analysis of Intrusion Detection using Deep Learning in IoT Networks [Z]. 2023 11th International Conference on Information and Communication Technology (ICoICT). 2023: 339-44.10.1109/ICoICT58202.2023.10262603

[11] PRISCILLA C V, PRABHA D P. A two-phase feature selection technique using mutual information and XGB-RFE for credit card fraud detection [J]. International Journal of Advanced Technology and Engineering Exploration, 2021, 8(85).

[12] PASHAEI E, PASHAEI E. Hybrid binary arithmetic optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical data [J]. The Journal of Supercomputing, 2022, 78(13): 15598-637.

[13] ZHOU H, WANG X, ZHU R. Feature selection based on mutual information with correlation coefficient [J]. Applied Intelligence, 2021, 52(5): 5457-74.

[14] QI Z, FEI J, WANG J, et al. An Intrusion Detection Feature Selection Method Based on Improved Mutual Information [Z]. 2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC). 2023: 1584-90.10.1109/itnec56291.2023.10082305

[15] WANG J, WEI J-M, YANG Z, et al. Feature Selection by Maximizing Independent Classification Information [J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(4): 828-41.

[16] NGUYEN H T, FRANKE K, PETROVIC S. Towards a Generic Feature-Selection Measure for Intrusion Detection [Z]. 2010 20th International Conference on Pattern Recognition. 2010: 1529-32.10.1109/icpr. 2010.378

[17] WANG Y, YANG H, LIU H, et al. Scaled IoT Intrusion Detection Model based on Improved PSO Algorithm Optimization [Z]. 2023 5th International Conference on Electronic Engineering and Informatics (EEI). 2023: 340-4.10.1109/eei59236.2023.10212914

[18] MITTAL S, MISHRA A K, TRIPATHI V, et al. A Comparative Analysis of Supervised Machine Learning Models for Smart Intrusion Detection in IoT Network [Z]. 2023 3rd Asian Conference on Innovation in Technology (ASIANCON). 2023: 1-6.10.1109/asiancon58793.2023.10270377

[19] ANANTHI P, RAMYA T E, JANANI R. Ensemble based Intrusion Detection System for IoT Device [Z]. 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS). 2023: 1073-8.10.1109/icscss57650.2023.10169426

[20] TAVALLAEE M, BAGHERI E, LU W, et al. NSL-KDD dataset download site. https://www.unb. ca/cic/ datasets/nsl.html

[21] TAVALLAEE M, BAGHERI E, LU W, et al. A detailed analysis of the KDD CUP 99 data set; proceedings of the IEEE International Conference on Computational Intelligence for Security & Defense Applications, F, 2009 [C].

[22] AMIN Z, KABIR A. A Performance Analysis of Machine Learning Models for Attack Prediction using Different Feature Selection Techniques [Z]. 2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science (BCD). 2022: 130-5.10.1109/bcd54882.2022.9900597

[23] ISMAIL M G, GHANY M A E, SALEM M A M. Enhanced Recursive Feature Elimination for IoT Intrusion Detection Systems [Z]. 2022 International Conference on Microelectronics (ICM). 2022: 193-6.10.1109/icm56065.2022.10005438

[24] LATHA R, BOMMI R M. Hybrid CatBoost Regression model based Intrusion Detection System in IoT-Enabled Networks [Z]. 2023 9th International Conference on Electrical Energy Systems (ICEES). 2023: 264-9.10.1109/icees57979.2023.10110148

[25] SHARAFALDIN I, LASHKARI A, GHORBANI A. CIC-IDS-2017 dataset download site. https://www. unb. ca/cic/datasets/ids-2017.html

[26] SHARAFALDIN I, LASHKARI A H, GHORBANI A A. Toward Generating a New Intrusion

ISSN 2616-5775 Vol. 7, Issue 3: 1-12, DOI: 10.25236/AJCIS.2024.070301

Detection Dataset and Intrusion Traffic Characterization; proceedings of the International Conference on Information Systems Security & Privacy, F, 2018 [C].

[27] MAJHI B, PRASTAVANA. An Improved Intrusion Dectection System using BoT-IoT Dataset [Z]. 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT). 2022: 488-92.10.1109/csnt54456.2022.9787639

[28] NAVEED M, USMAN S M, SATTI M I, et al. Intrusion Detection in Smart IoT Devices for People with Disabilities [Z]. 2022 IEEE International Smart Cities Conference (ISC2). 2022: 1-5.10.1109/isc255366.2022.9921991

[29] SINGH S, FERNANDES S V, PADMANABHA V, et al. MCIDS-Multi Classifier Intrusion Detection system for IoT Cyber Attack using Deep Learning algorithm [Z]. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). 2021: 354-60.10.1109/icicv50876.2021.9388579

[30] YANG X, LIU Q. Intrusion Detection Technology of Natural Resource Information System in The Internet of Things Environment [Z]. 2023 International Conference on Mechatronics, IoT and Industrial Informatics (ICMIII). 2023: 403-6.10.1109/icmiii58949.2023.00084

[31] WADATE A J, DESHPANDE S P. Edge-Based Intrusion Detection using Machine Learning Over the IoT Network [Z]. 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP). 2023: 1-6.10.1109/icetet-sip58143.2023. 10151535