Malicious Attack Detection Algorithm of Internet of Vehicles based on CW-KNN

Peng-Shou Xie, Cheng Fu, Tao Feng, Yan Yan, and Liang-Lu Li

(Corresponding author: Cheng Fu)

School of Computer and Communications, Lanzhou University of Technology
287 Lan-gong-ping Road, Lanzhou, Gansu 730050, China
(Email: 452708186@qq. com)

(Received Nov. 29, 2019; revised and accepted Mar. 21, 2020)

Abstract

With the wide application of multiple wireless communication technologies, vehicle nodes realize the connection of various networks such as WiFi, Bluetooth, 802.11p, LTE-V2X, and 5G. The attacker accesses the car's internal network through wireless communication,install malware for malicious attacks, these malicious attacks interfere with normal vehicle communication, spoofing or tapmer information, which will seriously threaten the security of the Internet of Vehicles. Therefore, this paper studies the main threats of malicious attacks on the Internet of Vehicles, extracts their malicious attack features, weights these features in combination, and proposed CW-KNN, which is a malicious attack detection algorithm suitable for Internet of Vehicles. Simulation experiments prove the effectiveness of the proposed algorithm.

Keywords: Combined Weight; CW-KNN; Internet of Vehicles; Malicious Attack Detection; Malware

1 Introduction

In the United States, the research work on the Internet of Vehicles (IoV) is based on Wireless Access in Vehicular Environment (WAVE) in Dedicated Short Range Communications(DSRC). The use of WAVE requires the construction of a dedicated service base station of IoV, this has greatly limited the popularity of IoV. But in China, in the 5G environment, vehicle nodes in the IoV rely on cellular wireless communication technology to communicate, and the related information is presented to the user through the upper-layer application. Huawei has established an LTE-V network and developed a communication chip. By loading a SIM card into a car, real-time communication services between cars can be achieved.

IoV is a part of wireless communication, wireless communication is generally integrated in vehicle systems, the CW-KNN detection algorithm proposed in this paper can also be integrated to protect the safe of IoV. Attackers installing malware can cause significant threats to IoV. The malicious attacks in this paper are active attacks, and the main threats are the following three aspects.

Denial of Service (DoS): Malware can interfere or block communication, causing vehicle nodes to fail to establish communication within the receiving range;

Spoofing: IoV's application technology requires accurate and timely access to application data. The attacker faked the relevant information and sent it, causing the vehicle to receive the wrong information, causing the driver to make abnormal behaviors, posing a certain threat to driving.

Tapmer: Malware can tamper information, each vehicle in IoV can be used as a terminal or relay node, information sent or received by them may be tampered, this will bring more scams and cause huge losses to the user.

It turns out that tapmer is easier than spoofing. Overall,malware will affect the normal function of the system, seriously affect driving safety, and even cause traffic accidents.

In terms of security of IoV, [15] proposed data falsification attack detection using hashes for enhancing network security and performance by adapting contention window size to forward accurate information to the neighboring vehicles in a timely manner. [20] in order to analyze the virus propagation under the road environment mixed with Cooperative Adaptive Cruise Control (CACC) vehicles andcommon vehicles, considering the interaction among traffic flow, information flow and virus propagation, CACC vehicle virus infection probability is calculated and the dynamic model of virus propagation is built. [22] aimed at the problem of security under the internet of vehicles environment, combining K area with fake names anonymous technology, a kind of improved Privacy Preservation Algorithm-Internet of Vehicles (PPA-IOV) privacy protection algorithm is formed. at the same time, researchers have also conducted related research on protocol and model strategies [8, 24, 25]. In terms of malicious attack detection, [1] proposed a solution to the problem of detecting semantic attacks in data based on hybrid automata implementation state constraints. [12] proposed a network intrusion detection model based on K-nearest neighbor(KNN)algorithm of extreme learning machine Extreme Learning Machine (ELM)feature mapping. [9] proposed a semi-supervised fuzzy kernel clustering algorithm based on quantum artificial fish group.

Although researchers have recently proposed many detection methods [2–7, 11, 13, 14, 17–19], these detection methods are not very suitable for the IoV. In the above, we have proposed the main threats of the IoV, which have corresponding attack features. Traditional malicious attack detection methods treat the feature contributions of the samples as the same, and do not weight the features from these threats. The direct use in the IoV will reduce the detection accuracy.

The main technical contributions of this paper are as follows. First, a specific method for establishing a simulated attack dataset of IoV is proposed, which can provide support for further research on the detection technology of the malicious attack of IoV. Second, the Combination Weight-KNN (CW-KNN) detection algorithm is proposed, which makes up for the lack of a malicious attack detection method in IoV.

2 Building a Simulated Attack Dataset of IoV

2.1 Feature Selection

The KDD CUP 99 [16]dataset marks each network connection as normal or abnormal. These anomaly types are further subdivided into 4 categories and a total of 39 attack types. A total of 22 attack types appeared in the training set, while the remaining 17 appeared only in the testing set. The criterion for evaluating intrusion detection is the ability to detect unknown attack types. KDD CUP 99 can well test the generalization power and applicability of the classification algorithm. It is also a recognized standard data set in the field of anomaly intrusion detection.

As the real-world malicious attack data set of IoV cannot be obtained, we improved KDD CUP 99 to obtain the simulation data set for experiments. The specific process is as follows.

The first step is to prune the original data set. There are 41 features in original KDD CUP 99 dataset. If all 41 features are used, this will lead to inaccurate and time-consuming results. Therefore, it is necessary to specifically remove some redundant features or low-important features. For example, "num_outbound_cmds" and "is_hot_login", The values are the same and they are all 0, So delete them.

The second step is to obtain the corresponding features of the malicious attack of the IoV. We studied the main threats to the IoV, and got the corresponding features. Some of these features are shown in Table 1.

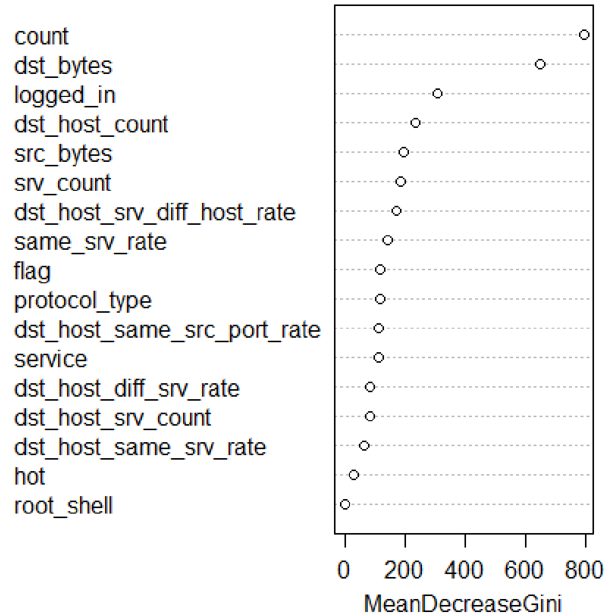

Figure 1: Feature contribution

Table 1: Feature contribution

The main malicious attacks on IoV	Features
Malicious code implantation	protocol_type, service, src_bytes, srv_count, count *etc.*
Spoofing	hot, root_shell, logged_in, num_access_files, flag *etc.*
Tamper	is_hot_login, is_guest_login, num_failed_logins *etc.*
Denial of service	src_bytes, dst_host_count, dst_host_srv_count *etc.*
Signal playback	dst_host_same_srv_rate, dst_host_same_src_port_rate *etc.*

The third step is to further optimize the selection of features. In order to avoid feature selection being too subjective in the previous section, and to make the selection persuasive, the Random Forest was used to evaluate the feature importance. Random forest can find out the degree of contribution of each feature to each tree, then take the average value, and finally compare the degree of contribution between features. the degree of contribution is usually measured using the Gini index as an evaluation indicator. as shown in Figure 1.

Finally, after many experiments, we selected 17 features, as shown in Table 2. We use the data set created by these 17 features as the simulation dataset for experiments.

Table 2: Final selected feature

Number	Feature name	Description	Types
1	protocol_type	Network protocol type	Discrete
2	service	The network service type of the target's host	Discrete
3	flag	Connected to a normal or incorrect state	Discrete
4	src_bytes	The number of bytes of data from source host to target host	Continuous
5	dst_bytes	The number of bytes of data from target host to source host	Continuous
6	hot	Number of times to access system sensitive files and directories	Continuous
7	logged_in	Successful login or not	Discrete
8	root_shell	Get superuser privileges or not	Discrete
9	count	The number of connections to the same target host as the current connection in the last two seconds	Continuous
10	srv_count	The number of connections with the same service as the current connection in the past two seconds	Continuous
11	same_srv_rat	Percentage of connections with the same service as the current connection in the last two seconds of a connection with the same target host	Continuous
12	dst_host_count	Of the top 100 connections,the number of connections with the same target host as the current connection	Continuous
13	dst_host_srv _count	Of the top 100 connections,the number of connections with the same target host and the same service as the current connection	Continuous
14	dst_host_same _srv_rate	Of the top 100 connections, percentage of connections with the same target host and the same service as the current connection	Continuous
15	dst_host_diff _srv_rate	Of the top 100 connections, percentage of connections with the same target host as the current connection but different services	Continuous
16	dst_host_same _src_port_rate	Of the top 100 connections, the percentage of connections with the same target host and the same source port as the current connection	Continuous
17	dst_host_srv _diff_host_rate	Of the top 100 connections, the current connection has the same target host and the same service. the percentage of connections with different source hosts from the current connection	Continuous

2.2 Data Preprocessing

To make the experiment more accurate, the data needs to be pre-processed before the experiment.

Numeric: One-hot encoding for the some features. for example,encoding "tcp", "udp", "icmp" as "0", "1", "2".

Standardization: Sij is the value normalized by the Xij value, as shown in Equations (1), (2), and (3).

$$S_{ij} = \frac{X_{ij} - AVG_j}{STAD_j} \quad (1)$$

$$AVG_j = \frac{X_{1j} + X_{2j} + \cdots + X_{nj}}{n} \quad (2)$$

$$STAD_j = \frac{|X_{1j} - AVG_j| + \cdots + |X_{nj} - AVG_j|}{n} \quad (3)$$

Normalization: The data is uniformly mapped to the interval $[0, 1]$, and N_{ij} is the normalized value of the X_{ij} value, as shown in Equation (4), Equation (5), and Equation (6).

$$N_{ij} = \frac{S_{ij} - X_{\min}}{X_{\max} - X_{\min}} \quad (4)$$

$$X_{\min} = \min\{S_{ij}\} \quad (5)$$

$$X_{\max} = \max\{S_{ij}\} \quad (6)$$

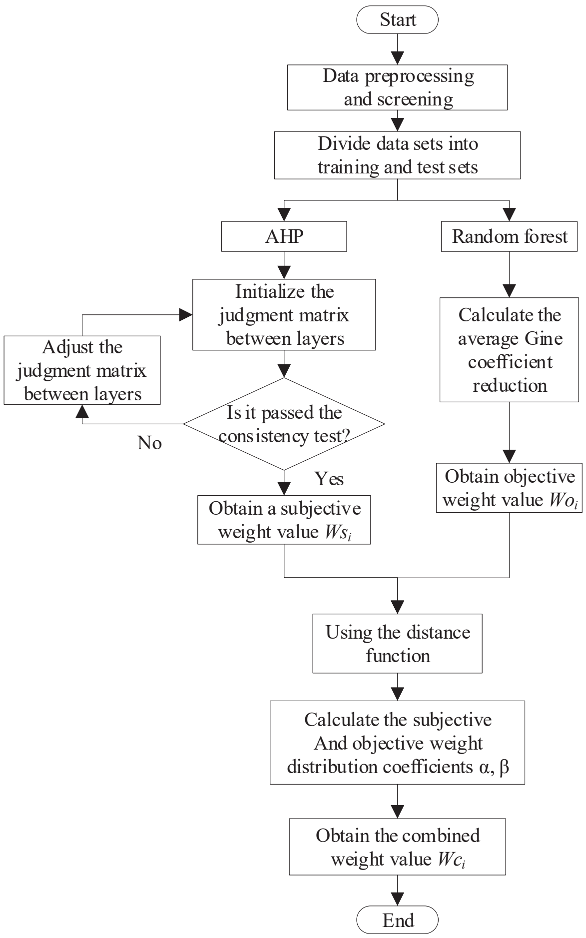

Figure 2: Weight calculation total flow chart

3 Building Malicious Attack Detection Algorithm of IoV based on CW-KNN

3.1 Weight Calculation

The main work of this section is to weight the KNN algorithm using combined weights, the purpose is to get the CW-KNN algorithm. In Part 2, 17 main malicious attack features of the IoV were selected. In this section, combined weights are given to these 17 features. We use the Analytic Hierarchy Process (AHP) to calculate subjective weights, and use random forests to calculate objective weights, then the distance function method is used to calculate combined weights. This not only reflects people's intuitive understanding of malicious attacks, but also reflects the authenticity of objective data, and also can make the results more accurate. The overall calculation process is shown in Figure 2.

(1) Calculating Subjective Weights

The first step is to use AHP to calculate subjective weights. AHP is a decision analysis method that combines qualitative and quantitative methods to solve multi-objective complex problems. It is widely used in various fields.

AHP model is established according to Table 2. As shown in Figure 3. But the established AHP model needs to pass the consistency check [23], details as follows. The calculation method of CI is shown in Equation (7).

$$CI = \frac{\lambda_{\max} - n}{n-1} \tag{7}$$

n is the dimension of the matrix, the value of RI is shown in Table 3.

Table 3: the value of RI

n	1	2	3	4	5	6
RI	0	0	0.58	0.9	1.12	1.24

The consistency ratiois CR, as shown in Equation (8).

$$CR = \frac{CI}{RI} \tag{8}$$

If $CR < 0.1$, passes the consistency check; Begin to calculate the subjective weight of 17 features. The judgment matrix [23] of the Criterion B_j ($j = 1, 2, 3, 4$) to the Goal A is as shown in Equation (9).

$$A = \begin{bmatrix} 1 & 2 & 2 & \frac{1}{2} \\ \frac{1}{2} & 1 & 1 & \frac{1}{2} \\ \frac{1}{2} & 1 & 1 & \frac{1}{2} \\ 2 & 2 & 2 & 1 \end{bmatrix} \tag{9}$$

The maximum eigenvalue is $\lambda_{\max}$. From the Equation $A\mu = \lambda^*_{\max}\mu$, $\lambda_{\max} = 4.0604$ can be calculated, the eigenvectors of B_j ($j = 1, 2, 3, 4$) is [0.2775, 0.3925, 0.1650, 0.1650].

$CR = 0.0226 < 0.1$ is calculated. Through the consistency check. The Weight of [B_1, B_2, B_3, B_4] is [0.2775, 0.3925, 0.1650, 0.1650].

The judgment matrix of the sub-criteria C_1-C_5 versus B_1 is as shown in Equation (10).

$$B_1 = \begin{bmatrix} 1 & 1 & \frac{1}{3} & \frac{1}{4} & \frac{1}{4} \\ 1 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 3 & 4 & 1 & \frac{1}{2} & \frac{1}{2} \\ 4 & 4 & 2 & 1 & 1 \\ 4 & 4 & 2 & 1 & 1 \end{bmatrix} \tag{10}$$

$\lambda_{\max} = 5.0552$ of the B_1 can be calculated, and the eigenvectors of C_i ($i = 1, 2, 3, 4, 5$) is [0. 0751, 0. 0709, 0.2028, 0.3256, 0.3256].

$CR = 0.0123 < 0.1$ is Calculated. Through the consistency check. the weight of [C_1, C_2, C_3, C_4, C_5] is [0. 0751, 0. 0709, 0.2028, 0.3256, 0.3256].

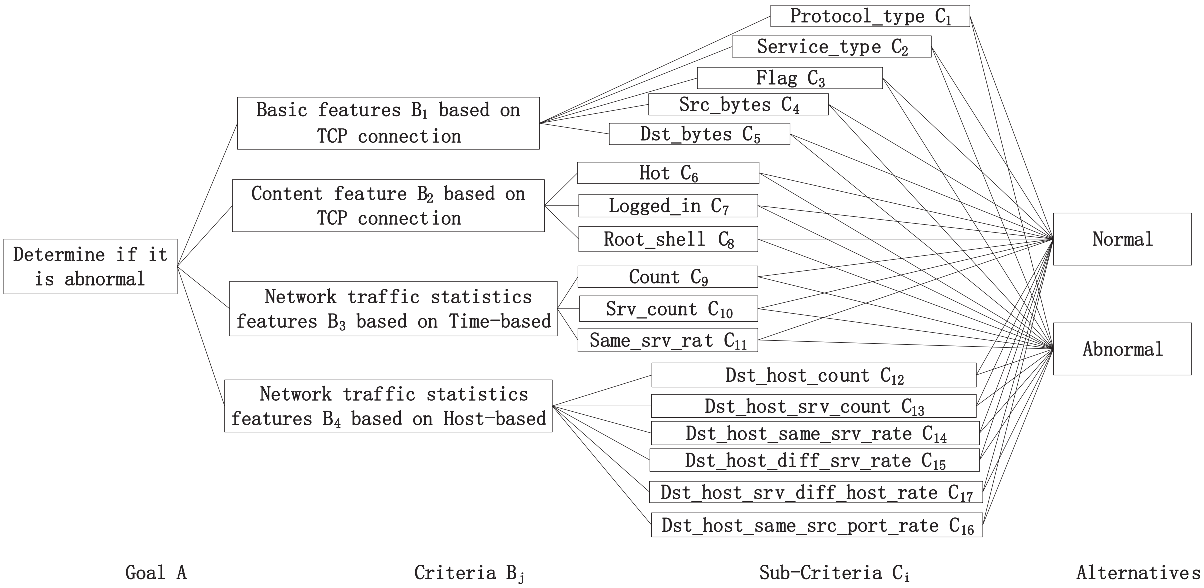

Figure 3: AHP model

The judgment matrix of the sub-criteria C_6-C_8 versus B_2 is as shown in Equation (11).

$$B_2 = \begin{bmatrix} 1 & 6 & 3 \\ \frac{1}{6} & 1 & \frac{1}{3} \\ \frac{1}{3} & 3 & 1 \end{bmatrix} \tag{11}$$

$\lambda_{\max} = 3.0183$ of the B2 can be calculated, and the eigenvectors of C_i $(i = 6, 7, 8)$ is [0.6548, 0. 0953, 0.2499].

$CR = 0.0176 < 0.1$ is Calculated. Through the consistency check. the weight of [C_6, C_7, C_8] is [0.6548, 0. 0953, 0.2499].

The judgment matrix of the sub-criteria C_9-C_{11} versus B_3 is as shown in Equation (12).

$$B_3 = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{2} \\ 3 & 1 & 3 \\ 2 & \frac{1}{3} & 1 \end{bmatrix} \tag{12}$$

$\lambda_{\max} = 3.0536$ of the B_3 can be calculated, and the eigenvectors of C_i $(i = 9, 10, 11)$ is [0.1571, 0.2493, 0.5936].

$CR = 0.0516 < 0.1$ is Calculated. Through the consistency test, the weight of [$C9$, C_{10}, C_{11}] is [0.1571, 0.2493, 0.5936].

The judgment matrix of the sub-criteria C_{12}-C_{17} versus B_4 is as shown in Equation (13).

$$B_4 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & \frac{1}{3} \\ 2 & 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & \frac{1}{2} \\ 3 & 3 & 1 & 2 & 3 & 2 \\ 3 & 3 & \frac{1}{2} & 1 & 2 & 2 \\ 2 & 2 & \frac{1}{3} & \frac{1}{2} & 1 & \frac{1}{3} \\ 3 & 2 & \frac{1}{2} & \frac{1}{2} & 3 & 1 \end{bmatrix} \tag{13}$$

$\lambda_{\max} = 6.2454$ of the B_4 can be calculated, and the eigenvectors of C_i $(i = 12, 13, 14, 15, 16, 17)$ is [0. 0660, 0. 0890, 0.3144, 0.2333, 0.1851, 0.1121].

$CR = 0.0390 < 0.1$ is Calculated. Through the consistency test, the weight of [C_{12}, C_{13}, C_{14}, C_{15}, C_{16}, C_{17}] is [0. 0660, 0. 0890, 0.3144, 0.2333, 0.1851, 0.1121].

The total consistency check of AHP model is as follows.

$$\begin{aligned} CI &= \sum_{j=1}^{4} B_j^* Cl_j \\ &= 0.2775 * \frac{5.0552 - 5}{5 - 1} + 0.3925 * \frac{3.0183 - 3}{3 - 1} \\ &+ 0.1650 * \frac{3.0536 - 3}{3 - 1} + 0.1650 * \frac{6.2454 - 6}{6 - 1} \\ &= 0.0198 \end{aligned}$$

$$\begin{aligned} RI &= \sum_{j=1}^{4} B_j^* RI_j \\ &= 0.2775 * 1.12 + 0.3925 * 0.58 + 0.1650 * 0.58 \\ &+ 0.1650 * 1.24 = 0.83875 \end{aligned}$$

The result is "$CR = CI/RI = 0.0236 < 0.1$", so the total consistency check is passed.

Subjective weight is defined as W_{S_i}. The calculation method of W_{S_i} is shown in Equation (14). And summary in Table 4.

$$W_{S_i} = \begin{cases} c_i * B_1;\ i = 1, 2, 3, 4, 5 \\ c_i * B_2;\ i = 6, 7, 8 \\ c_i * B_3;\ i = 9, 10, 11 \\ c_i * B_4;\ i = 12, 13, 14, 15, 16, 17 \end{cases} \tag{14}$$

(2) Calculation of Objective Weights

The second step uses a random forest to calculate objective weights. Random forests are not prone to overfitting

Table 4: Subjective weights

B layer	B_1	B_2	B_3	B_4	W_{S_i}
c layer	0.2775	0.3925	0.165	0.165	
C_1	0. 0751	\	\	\	0. 0208
C_2	0. 0709	\	\	\	0. 0197
C_3	0.2028	\	\	\	0. 0563
C_4	0.3256	\	\	\	0. 0904
C_5	0.3256	\	\	\	0. 0904
C_6	\	0.6548	\	\	0.257
C_7	\	0. 0953	\	\	0. 0374
C_8	\	0.2499	\	\	0. 0981
C_9	\	\	0.1571	\	0. 0259
C_{10}	\	\	0.2493	\	0. 0411
C_{11}	\	\	5936	\	0. 098
C_{12}	\	\	\	0. 066	0. 0109
C_{13}	\	\	\	0. 089	0. 0147
C_{14}	\	\	\	0.3144	0. 0519
C_{15}	\	\	\	0.2333	0. 0385
C_{16}	\	\	\	0.1851	0. 0305
C_{17}	\	\	\	0.1121	0. 0185

and have a high tolerance for outliers and noise. In this paper, the creation of the random forest model is performed in the R Language environment. It can provide some integrated tools, such as the "RandomForest" and "caret" toolkits required for this modeling.

In this paper, an another important reason for choosing a random forest is that the random forest can calculate the importance value of each variable. Random forest provides two basic variable importance values: Mean Decrease Gini and Mean Decrease Accuracy. this paper used Mean Decrease Gini as an objective weight. Some feature weights calculated by the random forest are shown in Figure 4.

```
training finished
 1) count                          0.179618
 2) ecr_i                          0.145816
 3) dst_host_srv_diff_host_rate    0.092629
 4) icmp                           0.071231
 5) same_srv_rate                  0.067123
 6) dst_bytes                      0.056923
 7) udp                            0.055820
 8) dst_host_count                 0.047003
 9) serror_rate                    0.039553
10) srv_count                      0.036703
```

Figure 4: Some features and weights

Objective weight is defined as W_{O_i}, Repeat the experiment 10 times and take the average, The serial number in W_{O_i} corresponds to Table 2. Objective weights calculated by Random Forest as shown in Table 5.

Table 5: Typical states of SEIR model

W_{O_1}	W_{O_2}	W_{O_3}	W_{O_4}
0. 0368	0. 0286	0. 0461	0. 0814
W_{O_5}	W_{O_6}	W_{O_7}	W_{O_8}
0. 0982	0. 0184	0. 0982	0. 0002
W_{O_9}	$W_{O_{10}}$	$W_{O_{11}}$	$W_{O_{12}}$
0.2087	0. 0532	0. 0627	0. 0859
$W_{O_{13}}$	$W_{O_{14}}$	$W_{O_{15}}$	$W_{O_{16}}$
0. 0266	0. 0266	0. 0327	0. 0384
$W_{O_{17}}$			
0. 0573			

(3) Calculation of Combined Weights

The third step uses the distance function method to calculate the combined weight. Because KNN is based on distance, and the distance function method introduces the concept of distance function, therefore, this paper choosed distance function method for combined weighting. The distance function method is used to reduce the difference between subjective and objective weights, so that the subjective and objective weights are organically combined,and this also makes the combination weights statistically significant.

Make W_{C_i} as the combined weight, α is the coefficient of subjective weighting, β is the coefficient of objective weight,as shown in Equation (15).

$$W_{C_i} = \alpha W_{S_i} + \beta W_{O_i} \tag{15}$$

The distance function expressions [10] is shown in Equation (16).

$$\mathrm{d}\left(W_{S_i}, W_{O_i}\right)=\sqrt{\frac{1}{2} \sum_{i=1}^{n}\left(W_{S_i}-W_{O_i}\right)^2} \tag{16}$$

To reduce the difference, make the distribution coefficient equal to the distance function, as shown in Equation (17).

$$\mathrm{d}\left(W_{S_i}, W_{O_i}\right)^2=(\alpha-\beta)^2 \tag{17}$$

The value of α and β is calculated, as shown in Equation (18). and $\alpha+\beta=1$.

$$\begin{aligned} \alpha &= \sqrt{\frac{1}{8} \sum_{i=1}^{n}\left(W_{S_i}-W_{O_i}\right)^2}+\frac{1}{2} \\ &= \sqrt{\frac{1}{8} * 0.11368}+\frac{1}{2} \\ &= 0.12+0.5=0.62 \end{aligned} \tag{18}$$

$\beta=1-0.62=0.38$, α and β can be substituted into the Equation (15) to calculate the combination weight of each feature, as shown in Table 6.

3.2 Improve KNN Algorithm

The direct use of KNN in the IoV will reduce the accuracy, because KNN uses Euclidean distance to consider the contribu- tion of all features in the sample as the same, and does not weight features,Therefore, this section is to improve the KNN algorithm. The combined weights calculated in Table 6 are brought into the weighted distance to obtain the CW-KNN classification algorithm. The specific process is as follows.

(1) Weight the Distance

Different features have their corresponding weights. Bring the combined weight W_{C_i} into the Euclidean distance, obtaine the weighted distance of two arbitrary samples x and y, as shown in Equation (19).

$$\mathrm{d}(x, y)=\sqrt{\sum_{i=1}^{n}\left(x_i-y_i\right)^2 W_{C_i}} \tag{19}$$

(2) Building CW-KNN

The main classification decision rule in CW-KNN is a majority vote. The process is as follows.

Algorithm 1 CW-KNN

Input: training dataset $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_i, y_i)\}$; k is the number of neighbors;

Output: The category y to which the instance x belongs;

1: Begin

2: Calculate combination weighted Euclidean distance

$$\mathrm{d}(x, y)=\sqrt{\sum_{i=1}^{n}\left(x_i-y_i\right)^2 W_{C_i}}$$

3: Find the k points closest to x in the training set D,

4: The neighborhood of x covering the k points is denoted as $Nk(x)$

5: In $Nk(x)$, ater majority vote, determine category to which instance x belongs;

6: End

4 Simulation Experiment

4.1 Experimental Benchmarks and Methods

This paper used python3 to perform binary classification experiments on CW-KNN. The experimental benchmark is to use the confusion matrix to analyze from four aspects: Accuracy, Precision, Recall, and F1. In order to verify the efficiency of CW-KNN, it will be compared with many different types of detection methods. Specifically, it includes KNN without combined weighting, SVM(Support Vector Machine) based on machine learning, FCD-KNN [21] based on Related to the Distance of Attribute Values, Adaboost based on ensemble learning and Random Forest based on tree.

The experiment is divided into two parts. The first part is the comparison between CW-KNN and the other two KNN algorithms. The second part is the comparison between CW-KNN and other types of classification algorithms.

Considering the factors of calculation time and memory consumption, in this paper, 10% training set and extracts part of the testing set are finally used for experiments, as shown in Table 7:

4.2 Comparison within KNN

This section reserch on the effect of different values of K on CW-KNN, and compared with the other two KNN algorithms. The value of K is the nearest neighbor number,and it is the most important value in Knn. The value of K will directly affect the quality of classification. The combined weight set by CW-KNN is shown in Table 6. K takes 3 to 10 and $K \in Z$, the experimental results are shown in Figure 5.

From Figure 5 it can be seen that when $K=7$, the accuracy of all the KNN algorithms is the same. When $k=8$, the accuracy of FCD-KNN and CW-KNN is the same. When $k \neq 7$ or $\neq 8$, the accuracy of CW-KNN is higher than KNN and FCD-KNN.

In order to reduce the influence of the values of K on experimental results, this paper set $K=7$, and get the ROC curves of the three kind of KNN algorithms, As shown in Figure 6.

The experiments in this section prove that the accuracy of CW-KNN is higher than KNN and FCD-KNN.

Table 6: Feature combination weight table

Feature number and name $i = 1, 2, \cdots, 17$	**Subjective weight** W_{S_i}	**Objective weight** W_{O_i}	**Combination weight** W_{C_i}
1. protocol_type	0. 0208	0. 0368	0. 0267
2. service	0. 0197	0. 0286	0. 023
3. flag	0. 0563	0. 0461	0. 0524
4. src_bytes	0. 0904	0. 0814	0. 087
5. dst_bytes	0. 0904	0. 0982	0. 0934
6. hot	0.257	0. 0184	0.1663
7. logged_in	0. 0374	0. 0982	0. 0605
8. root_shell	0. 0981	0. 0002	0. 0609
9. count	0. 0259	0.2087	0. 0954
10. srv_count	0. 0411	0. 0532	0. 0457
11. same_srv_rat	0. 098	0. 0627	0. 0846
12. dst_host_count	0. 0109	0. 0859	0. 0394
13. dst_host_srv_count	0. 0147	0. 0266	0. 0192
14. dst_host_same_srv_rate	0. 0519	0. 0266	0. 0423
15. dst_host_diff_srv_rate	0. 0385	0. 0327	0. 0363
16. dst_host_same_src_port_rate	0. 0305	0. 0384	0. 0335
17. dst_host_srv_diff_host_rate	0. 0185	0. 0573	0. 0332

Table 7: Sample distribution of dataset

Num	**Type**	**Number of samples**	
		Training	**Testing**
0	normal	97278	118835
1	abnormal	396743	29371

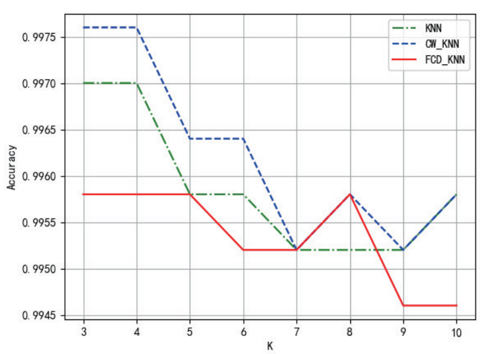

Figure 5: Accuracy with different K values

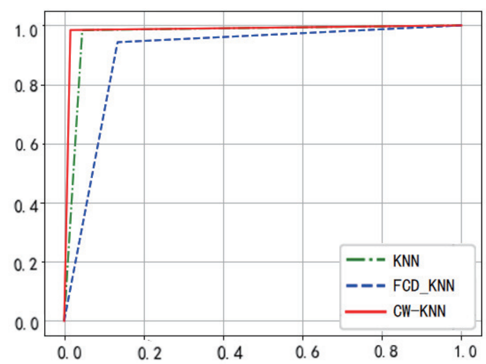

Figure 6: ROC graph of three KNN algorithms

4.3 Comparison Between CW-KNN and Other Classification Algorithms

This section focuses on the measurement of CW-KNN benchmarks,and compared with the other five classification methods.

The value of K of all KNN is set to 7, the other classification algorithm parameters are Python3 original parameters. Obtain the confusion matrix of 6 classification algorithms through experiments, as shown in Tables 8-13.

Comparison of multiple classification results, As shown in Table 14.

From Table 14, it can be seen that the value of F1 of CW-KNN is higher than other classification algorithms, which illustrates CW-KNN is superior in comprehensive performance. Second, CW-KNN has improved in Precision, which shows that CW-KNN has better detection ability than other classification algorithms. however, CW-KNN is inferior to SVM and Adaboost in terms of Accuracy, this is also an issue that needs to be addressed in the next step. In sumary,the experiment proves that the CW-KNN proposed in this paper has better classification effect in binary classification.

Table 8: Confusion matrix of KNN

KNN		prediction	
		normal	abnormal
actual	normal	117910	925
	abnormal	1	29370
Precision		0.994	
Recall		0.992	
Accuracy		0.995	
F1		0.984	

Table 9: Confusion matrix of Random Forest

Random Forest		prediction	
		normal	abnormal
actual	normal	118129	706
	abnormal	18	29353
Precision		0.977	
Recall		0.999	
Accuracy		0.995	
F1		0.988	

Table 10: Confusion matrix of Adaboost

Adaboost		prediction	
		normal	abnormal
actual	normal	118316	519
	abnormal	125	29246
Precision		0.983	
Recall		0.996	
Accuracy		0.996	
F1		0.989	

Table 11: Confusion matrix of FCD-KNN

FCD-KNN		prediction	
		normal	abnormal
actual	normal	118683	152
	abnormal	1	29370
Precision		0.995	
Recall		0.998	
Accuracy		0.995	
F1		0.991	

5 Conclusion

Few researchers currently optimize the classification algorithm for IoV, and the KNN without combined weighting does not consider the difference of sample attribute contribution. Therefore, this paper proposed CW-KNN algorithm for IoV. First of all, we selected the featurs of main threats according to IoV, built a simulated attack dataset of IoV, then calculated the combined weight of each feature, and finally brought the combined weight into the KNN for classification. The experimental results show that the CW-KNN has higher efficiency.

The shortcoming of this paper is that the accuracy of CW-KNN is lower than SVM and Adaboost, this will be the next problem to be solved. With the increase of new types of malicious attacks of IoV, dimensions of data will also increase, KNN is based on distance, so it is not good for multi-dimensional data processing, which may lead to a decline in accuracy. Random forest is better at processing multi-dimensional data, so the next step is to bring the combined weights to the Random Forest for research to improve the accuracy.

Acknowledgement

This research is supported by the National Natural Science Foundations of China under Grants No.61862040, No.61762059 and No. 61762060. The authors gratefully acknow-ledge the anonymous reviewers for their helpful comments and suggestions.

References

[1] S. Adepu and A. Mathur, "From design to invariants: Detecting attacks on cyber physical systems," in *IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C'17)*, pp. 533–540, 2017.

[2] R. C. Baishya, N. Hoque, and D. K. Bhattacharyya, "Ddos attack detection using unique source IP deviation," *International Journal Network Security*, vol. 19, no. 6, pp. 929–939, 2017.

[3] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in iot using SDN," in *The 27th International Telecommunication Networks and Applications Conference (ITNAC'17)*, pp. 1–6, 2017.

[4] J. Y. Chen and X. Z. Xu, "Research on network attack detection based on self-adaptive immune computation," *Computer Science*, vol. 45, no. 6A, pp. 364–370, 2018.

[5] F. Chen, Z. Ye, C. Wang, L. Yan, and R. Wang, "A feature selection approach for network intrusion detection based on tree-seed algorithm and k-nearest neighbor," in *IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS'18)*, pp. 68–72, 2018.

[6] C. Guo, Y. Ping, N. Liu, and S. S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[7] T. Jeyaprakash and R. Mukesh, "A survey of mobility models of vehicular adhoc networks and simulators," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 94–101, 2015.

Table 12: Confusion matrix of SVM

SVM		prediction	
		normal	abnormal
actual	normal	118548	287
	abnormal	170	29201
Precision		0.990	
Recall		0.994	
Accuracy		0.997	
F1		0.992	

Table 13: Confusion matrix of CW-KNN

CW-KNN		prediction	
		normal	abnormal
actual	normal	118548	287
	abnormal	170	29201
Precision		0.997	
Recall		0.998	
Accuracy		0.995	
F1		0.993	

[8] X. Jian, W. J. Li, H. Y. Geng, and Y. B. Zhai, "An anti-dos attack rfid security authentication protocol in the internet of vehicles," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, no. 2, pp. 114–119, 2019.

[9] G. Li, "Research on network intrusion detection model based on quantum artificial fish school and fuzzy kernel clustering algorithm," *Software Engineering*, vol. 22, no. 6, pp. 33–37, 2019.

[10] T. H. Li, J. Xue, and X. Wei, "Application of combinedweigh mehod and comprehensive index method based on cask theory in ecological waterway assessment of yangtze river," *Journal of Basic Science and Engineering*, vol. 27, no. 1, pp. 36–49, 2019.

[11] J. P. Liu, W. X. Zhang, and Z. H. Tang, "Adaptive network intrusion detection based on fuzzy rough set-based attribute reduction and gmm-lda-based optimal cluster feature learning," *Control and Decision*, vol. 34, no. 2, pp. 243–251, 2019.

[12] M. R. Mohamed, A. A. Nasr, I. F. Tarrad, and M. Z. Abdulmageed, "Exploiting incremental classifiers for the training of an adaptive intrusion detection model," *International Journal Network Security*, vol. 21, no. 2, pp. 275–289, 2019.

Table 14: Comparison of multiple classification results

methods	Benchmarks			
	F1	Accuracy	Precision	Recall
CW-KNN	0.993	0.995	0.997	0.998
KNN	0.984	0.995	0.994	0.992
FCD-KNN	0.991	0.995	0.995	0.998
Adaboost	0.989	0.996	0.983	0.996
SVM	0.992	0.997	0.99	0.994
Random Forest	0.988	0.995	0.977	0.999

[13] E. U. Opara and O. A. Soluade, "Straddling the next cyber frontier: The empirical analysis on network security, exploits, and vulnerabilities," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 10–18, 2015.

[14] K. K. Ravulakollu, Amrita, "A hybrid intrusion detection system: Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers," *International Journal of Network Security (IJNS'18)*, vol. 20, no. 1, pp. 41–55, 2018.

[15] D. B. Rawat, M. Garuba, L. Chen, and Q. Yang, "On the security of information dissemination in the internet-of-vehicles," *Tsinghua Science and Technology*, vol. 22, no. 4, pp. 437–445,2017.

[16] J. D. Ren, X. Q. Liu, and Q. Wang, "An multi-level intrusion detection method based on KNN outlier detection and random forest," *Journal of Computer Research and Development*, vol. 56, no. 3, pp. 566–575, 2019.

[17] Y. Ren, S. Wang, X. Zhang, and M. S. Hwang, "An efficient batch verifying scheme for detecting illegal signatures," *International Journal Network Security*, vol. 17, no. 4, pp. 463–470, 2015.

[18] T. A. Tchakoucht and M. Ezziyyani, "Building a fast intrusion detection system for high-speed-networks: probe and dos attacks detection," *Procedia Computer Science*, vol. 127, pp. 521–530, 2018.

[19] J. Wang and L. L. Yang, "Multitier ensemble classifiers for malicious network traffic detection," *Journal on Communications*, vol. 39, no. 10, pp. 155–165, 2018.

[20] L. Wei, Y. P. Wang, and H. K. Qin, "An algorithm of the privacy security protection based on location service in internet of vehicles," *Automotive Engineering*, vol. 41, no. 3, pp. 252–258, 2019.

[21] H. H. Xiao and Y. M. Duan, *Improved of KNN Algorithm Based on Related to the Distance of Attribute Values*. PhD thesis, 2013.

[22] P. S. Xie, T. X. Fu, and H. J. Fan, "An algorithm of the privacy security protection based on location service in the internet of vehicles," *International Journal of Network Security*, vol. 21, no. 4, pp. 556–565, 2019.

[23] A. M. Yang and F. Gao, "Cloud computing security evaluation and countermeasure based on AHP-fuzzy comprehensive evaluation," *Journal on Communications*, vol. 37, no. Z1, pp. 104–110, 2016.

[24] H. Zhao, D. Sun, H. Yue, M. Zhao, and S. Cheng, "Dynamic trust model for vehicular cyber-physical systems," *International Journal Network Security*, vol. 20, no. 1, pp. 157–167, 2018.

[25] H. Zhao, H. Yue, T. Gu, and W. Li, "CPS-based reliability enhancement mechanism for vehicular emergency warning system," *International Journal of Intelligent Transportation Systems Research*, vol. 17, no. 3, pp. 232–241, 2019.

Biography

Peng-shou Xie was born in Jan.1972. He is a professor and a supervisor of master student at Lanzhou University of Technology. His major research field is Security on Internet of Things. E-mail: xiepsh_lut@163. com

Cheng Fu was born in Jun.1991. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 452708186@qq. com

Tao Feng was born in Dec.1970. He is a professor and a supervisor of Doctoral student at Lanzhou University of Technology. His major research field is modern cryptography theory, network and information security technology. E-mail: fengt@lut. cn

Yan Yan was born in Oct.1980. She is a associate professor and a supervisor of master student at Lanzhou University of Technology. Her major research field is privacy protection, multimedia information security. E-mail: yanyan@lut. cn

Liang-lu Li was born in Jun.1992. She is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 1141685642@qq. com