Hindawi
Scientific Programming
Volume 2021, Article ID 9972589, 9 pages
https://doi.org/10.1155/2021/9972589

Research Article

K-Modes Clustering Algorithm Based on Weighted Overlap Distance and Its Application in Intrusion Detection

Yawen Dai,[1] Guanghui Yuan,[2] Zhaoyuan Yang ,[3] and Bin Wang[4]

[1]*Institute of Rail Transit, Tongji University, Shanghai 201804, China*
[2]*School of Economics and Management, Shanghai University of Political Science and Law, Shanghai 201701, China*
[3]*School of Finance and Business, Shanghai Normal University, Shanghai 201418, China*
[4]*School of Humanities, Shanghai University of Finance and Economics, Shanghai 200433, China*

Correspondence should be addressed to Zhaoyuan Yang; 1853871@tongji.edu.cn

Received 3 April 2021; Revised 4 May 2021; Accepted 18 May 2021; Published 25 May 2021

Academic Editor: Shah Nazir

In order to better apply the K-modes algorithm to intrusion detection, this paper overcomes the problems of the existing K-modes algorithm based on rough set theory. Firstly, for the problem of K-modes clustering in the initial class center selection, an initial class center selection algorithm Ini_Weight based on weighted density and weighted overlap distance is proposed. Secondly, based on the Ini_Weight algorithm, a new K-modes clustering algorithm WODKM based on weighted overlap distance is proposed. Thirdly, the WODKM clustering algorithm is applied to intrusion detection to obtain a new unsupervised intrusion detection model. The model detects the intrusion by dividing the clusters in the clustering result into normal clusters and abnormal clusters and analyzing the weighted average density of the object x to be detected in each cluster and the weighted overlapping distance of x and each center point. We verified the intrusion detection performance of the model on the KDD Cup 99 dataset. The experimental results of the current study show that the proposed intrusion detection model achieves efficient results and solves the problems existing in the present-day intrusion detection system to some extent.

1. Introduction

For the threat of network security, there are many corresponding network security defence technologies. For example, traditional network security defence technologies include firewall, antivirus software, digital signature, digital authentication, and data encryption. Traditional network security defence technology is only a static defence method, and it is difficult to effectively protect our network security [1]. Intrusion detection helps systems count the number of errors in the system and analyze the causes of system vulnerabilities, detecting known intrusions and alarms, auditing of new anomalous behaviors, and integrity assessment of critical data files.

In view of the problems existing in existing intrusion detection systems, data mining techniques have been widely used in the field of intrusion detection in recent years. Among them, cluster-based unsupervised intrusion detection has caused a lot of attention. At present, many clustering algorithms have been applied to intrusion detection. As an effective extension of the K-means algorithm, the K-modes algorithm can effectively deal with categorical data. It inherits the efficient features of the K-means algorithm, and the algorithm is simple, easy to implement, and widely used in many fields. Therefore, the K-modes algorithm has a very broad application prospect in the field of intrusion detection. However, the research on applying K-modes algorithm to intrusion detection is still rare [2, 3].

In order to better apply the K-modes algorithm to intrusion detection, this paper solves the problems of the existing K-modes algorithm based on rough set theory and applies the improved algorithm to intrusion detection [4]. Firstly, for the problem of K-modes clustering in the initial class center selection, an initial class center selection algorithm Ini_Weight based on weighted density and

weighted overlap distance is proposed. Secondly, based on the Ini_Weight algorithm, a new K-modes clustering algorithm WODKM based on weighted overlap distance is proposed. Thirdly, the WODKM clustering algorithm is applied to intrusion detection to obtain a new unsupervised intrusion detection model UIDM_WODKM.

The model detects the intrusion by dividing the clusters in the clustering result into normal clusters and abnormal clusters and analyzing the weighted average density of the object x to be detected in each cluster and the weighted overlapping distance of x and each center point. We verified the intrusion detection performance of the model on the KDD Cup 99 dataset.

The organization of the paper is as follows: Section 2 describes the intrusion detection system. Section 3 briefly discusses the K-modes clustering algorithm based on weighted overlap distance. The application of K-modes clustering algorithm based on weighted overlap distance in intrusion detection is given in Section 4. The paper is concluded in Section 5.

2. Intrusion Detection

Intrusion detection is a proactive network security defence strategy that can make up for the shortcomings of traditional static security policies and thus becomes a reasonable complement to traditional static defence strategies such as firewalls [5]. System administrators' security management capabilities have been extended through auditing, monitoring, intrusion identification and response, and reducing the workload of system administrators.

Through the intrusion detection system, system administrators can grasp the status of the network system in real time, including the status of programs, files, and devices, and help network system administrators to develop accurate and complete strategies [6]. In addition, the intrusion detection system dilutes the restrictions on professionals in network security, making it easy for nonprofessional personnel to manage network systems. In addition, the intrusion detection system can respond to the discovered network intrusion, illegal operation, timely and proactive response, and achieve the purpose of active defence [7].

Intrusion detection system (IDS) is a collection of network security systems or systems that detect abnormal behaviors in the system or changes in network status and can alert or take proactive responses [8]. It differs from other network security systems in that it is also an active security protection technology [9–11]. Intrusion detection in an intrusion detection system is mainly divided into the following steps:

(1) Collect message: intrusion detection first collects the corresponding system and network information. The main content includes the content of network traffic, system health, status, and behavior of user connection activities.

(2) Signal analysis: the information collected above is generally analyzed by three technical means: pattern matching, statistical analysis, and integrity analysis. The first two methods are used for real-time intrusion detection, while the integrity analysis is used for postmortem analysis.

(3) Real-time recording, alarm, and response: finally, the intrusion detection system can timely respond to detected intrusions or other network attacks. This includes logging intrusion information and alerting the network administrator.

At present, there are many kinds of intrusion detection systems. According to different classification standards, we classify the current intrusion detection system as follows.

2.1. Classify according to the Principle of Detection

(1) Anomaly detection: anomaly detection first summarizes the characteristics of normal behavior and then judges whether the user behavior is intrusive according to the user's activities or the use of resources in the system. Anomaly detection needs to establish a model to determine the normal user behavior and activities before detection and then to determine which qualified behaviors can be marked as "exceptions" and to make corresponding processing.

(2) Abuse detection: misuse detection first establishes the feature library based on the behavioral characteristics of the abnormal operation collected. When the observed user activity or system resource usage matches the records in the feature library, it is considered an intrusion. When the intrusion behavior matches the normal user activity, the system produces a false alarm; when the feature library does not store the features matching with a new attack behavior, the system produces a false alarm. Therefore, the focus of abuse detection is how to update the feature library automatically to reduce the false alarm rate.

(3) Mixed detection: hybrid detection combines the advantages of the above two methods, mainly based on the normal data flow of the system to detect intrusion behavior. And before making a decision, we not only analyze the normal behavior of the system but also observe the suspicious intrusion behavior. Therefore, the results of mixed detection are more comprehensive. It is reliable.

2.2. Classify according to Data Sources

(1) Host-based intrusion detection system: host-based intrusion detection system obtains data from the host where the system runs. It mainly protects the security of the host where the system is located. Usually, this system is installed on the host which needs to be checked, mainly on the relatively critical system files and executable files.

(2) Network-based intrusion detection system: with the development of network technology, the host-based

intrusion detection system has been difficult to meet the needs of network security, and the network-based intrusion detection system came into being. The data source of this kind of system analysis is the data packet on the network. It mainly analyzes and detects the intrusion behavior on the corresponding network segment according to the data flow, data packet, and corresponding protocol on the network.

2.3. Classify according to the Architecture of the Intrusion Detection System

(1) Centralized intrusion detection system: centralized intrusion detection systems usually have multiple auditors on different hosts, but only one central server is used for intrusion detection. In this way, the audit program sends the collected data trace to the central server, which then analyzes and detects the data. The scalability and configurability of such intrusion detection systems are relatively poor.

(2) Distributed intrusion detection system: the distributed intrusion detection system (DIDS) sends the corresponding tasks to several different host-based IDS by the central detection server, and these host-based IDS are nonhierarchical. Each of them is responsible for monitoring suspicious activities on the corresponding host. This kind of intrusion detection system has relatively high scalability and security, but its maintenance cost is high and the workload of the host is relatively heavy.

3. K-Modes Clustering Algorithm Based on Weighted Overlap Distance

The schematic of the intrusion detection system is shown in Figure 1.

3.1. K-Modes Clustering Algorithm. The K-means algorithm is a clustering algorithm commonly used in the field of data mining. However, this algorithm can only process numeric data, but cannot process subtype data [12]. In response to this problem, Huang et al. further proposed the K-modes clustering algorithm.

Clustering is the process of dividing dataset into a plurality of groups or clusters composed of similar objects [13] The similarity between objects in the same group is made as high as possible, and the similarity of objects between different groups is as low as possible. Clustering is an unsupervised machine learning method that does not have any prior knowledge of the datasets that need to be classified [14]. Datasets are automatically divided into groups or clusters based solely on similarity metrics. Try to make the similarity between samples in the same group high and the sample similarity between different groups is low. The groups in the cluster do not need to be defined in advance and are automatically divided according to the inherent similarity of the data according to the actual characteristics of the data [13]. The cluster analysis system inputs the criteria for measuring the similarity between data and the dataset that needs to be classified, and the output is the result of the well-classified class. The additional result of cluster analysis is a comprehensive description of each group, which can be used to provide a more in-depth analysis of the characteristics of the dataset [15].

The K-means clustering algorithm is a popular clustering algorithm with simple and efficient characteristics, but it can only deal with numerical data. The K-modes clustering algorithm is an extension of the K-means algorithm, which can deal with class attribute data. It inherits the characteristics of the K-means clustering algorithm which is efficient and easy to implement, so it is widely used in various fields.

Distance measure (or similarity measure) is the most basic and important part of the K-modes clustering algorithm. As mentioned earlier, the existing geometric distance measurement for numeric data is not suitable for categorical data. It is necessary to design some new distance measurement mechanisms according to the characteristics of categorical data. At present, in the K-modes clustering algorithms, most of them use simple overlap distance to measure the dissimilarity between any two objects X and Y; that is, the number of attributes with different values of X and Y is taken as the distance between X and Y. The process of clustering and segmentation is shown in Figure 2.

When calculating the overlap distance between any two objects, we mainly get the final result by comparing the values of the two objects on each attribute. If we use the same way to deal with all the attributes in the information table without distinguishing them, this approach obviously does not conform to objective reality and eventually will lead to the clustering results deviation, thus affecting the performance of the K-modes algorithm.

The K-modes algorithm uses a simple overlapping distance measure to calculate the distance between objects [4, 16]. The specific definition of the distance measure is as follows: for any two objects X and y, X and Y in the universe U, the distance $d(x, y) = \sum_{a \in A} \delta_a(x, y)$, which for any $a \in A$, $\delta_a(x, y)$ denotes a simple overlap distance between X and Y on the conditional property a, that is, if $f(x, a) = f(y, a)$, $\delta_a(x, y) = 0$; otherwise, $\delta_a(x, y) = 1$.

The schematic of the K-means algorithm is shown in Figure 3.

Obviously, the traditional K-modes clustering algorithm has the following problems [17–20]:

(1) Select the initial center point by random selection: choosing the initial center point in this way will lead to the instability of clustering results. If we want to get relatively good clustering results, we need to repeatedly execute the K-modes clustering algorithm and select the best group of data from it. For intrusion detection, it is inappropriate to select the initial center point by using the above method because the data to be processed by the network intrusion detection system is usually very huge. Repeated K-modes

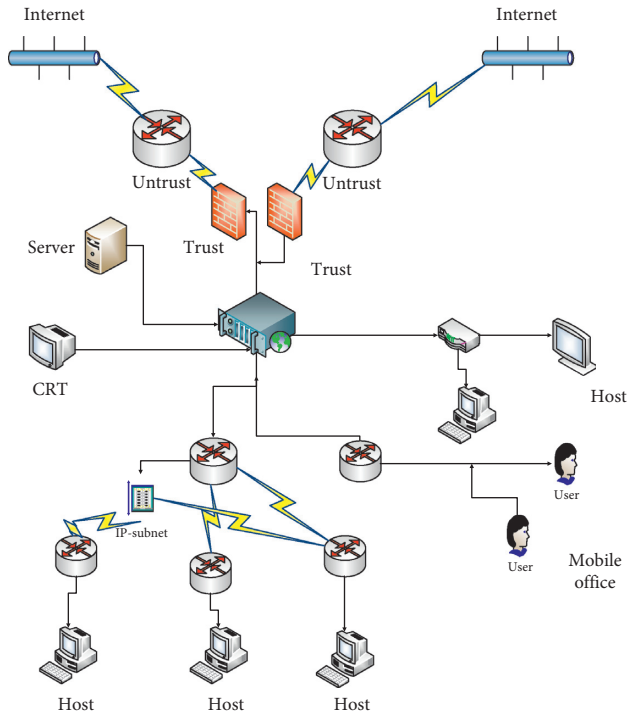

Figure 1: Intrusion detection detecting system.

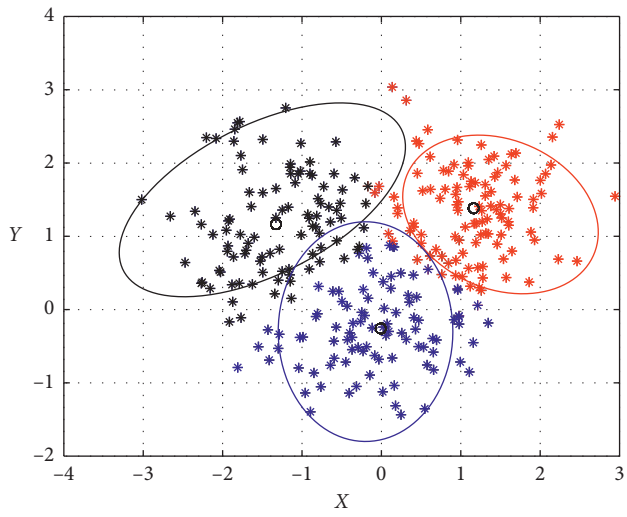

Figure 2: Process of clustering and segmentation.

clustering algorithms on massive, high-dimensional network data are often not feasible in time.

(2) Traditional K-modes algorithms usually use simple overlap distance to measure the dissimilarity between objects X and y. Simple overlapping distance is simple and intuitive, but there are still some problems in its practical application. For example, it assumes that all attributes play the same role in calculating the distance between objects; that is, each attribute has a weight equal to 1. However, in many practical cases, the impact of different attributes on distance calculation is likely to be different.

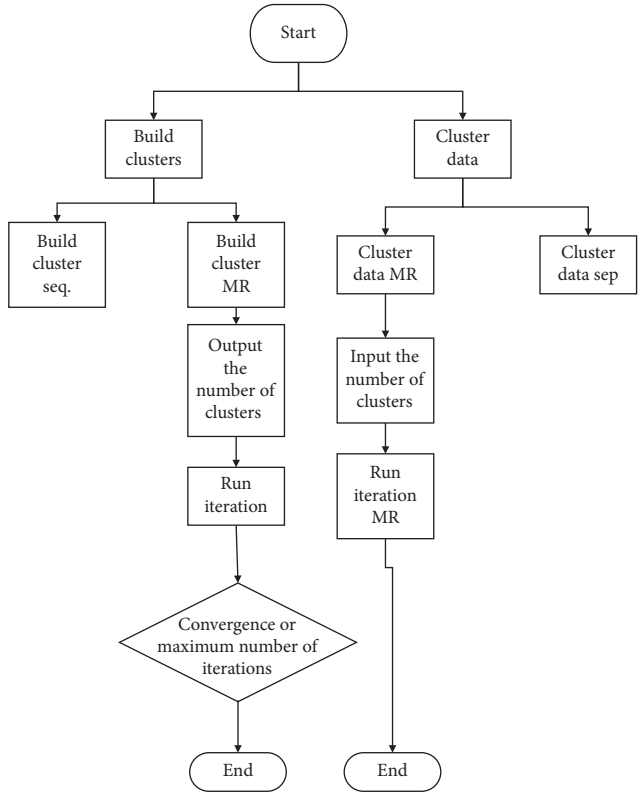

Figure 3: K-modes algorithm.

Therefore, it is necessary to assign a weight value to each attribute and through the size of the weight value to reflect the contribution of each attribute to distance calculation.

3.2. Weighted Overlap Distance. Aiming at the problems existing in the traditional K-modes clustering algorithm, this paper proposes a new K-modes clustering algorithm WODKM based on weighted overlap distance. Different from the traditional K-modes clustering algorithm, the WODKM algorithm uses the initial class center selection algorithm Ini_Weight based on weighted density and weighted overlap distance proposed in Chapter 3 to select the initial center point. In addition, the WODKM algorithm uses a weighted overlap distance metric instead of a simple overlap distance metric when calculating the distance between objects and the distance between the object and the center point (Algorithm 1).

The specific steps of the WODKM algorithm are in Algorithm 1.

In the worst case, the time complexity of steps (1)–(5) in the Ini_Weight algorithm is O ($|A|2 \times |U|$), The time complexity of steps (6) – (10) is O ($K2 \times |A| \times |U|$), where K is the number of clusters. Therefore, in the worst case, the time complexity of the algorithm is O ($K2 \times |A| \times |U| + |A|2 \times |U|$), and the space complexity is O ($|A| \times |U|$). Obviously, the time complexity of Ini_Weight is linearly relative to the number of objects.

To evaluate the performance of the Ini_Weight algorithm, we experimented with the following four subtype UCI datasets. The basic information of the above four datasets is shown in Table 1.

We compare the Ini_Weight algorithm with four existing initial center selection methods: (1) random method [21]; (2) Cao's algorithm [22]; (3) Wu's algorithm [23]; and (4) Khan's algorithm. The main steps of the experiment are as follows: first, different initialization algorithms are used to select the initial center; secondly, K-modes clustering is performed based on the initial center selected in the previous step (using the classical K-modes algorithm proposed by Huang for clustering); finally, compare the clustering results corresponding to different initialization algorithms. Table 2 shows the K-modes clustering results for different initialization algorithms on the four datasets Soybean, Zoo, Breast, and Mushroom.

There is

$$\begin{aligned} \mathrm{PR} &= \frac{\sum_{i=1}^{K} a_i/a_i + b_i}{K}, \\ \mathrm{RE} &= \frac{\sum_{i=1}^{K} a_i/a_i + c_i}{K}, \\ \mathrm{AC} &= \frac{\sum_{i=1}^{K} a_i}{|U|}. \end{aligned} \tag{1}$$

From Table 2, we can see that the Ini_Weight algorithm performs better than the random method because the AC, PR, and RE of Ini_Weight outperform the random method on the three datasets of Soybean, Breast, and Mashroom. On Zoo, although Ini_Weight's PR is slightly worse than the random method, it is significantly superior to the random method in AC and RE. In addition, the clustering results obtained by random methods are different each time, and our method can get stable clustering results.

Although Cao's method and Wu's method are also superior to random methods, Ini_Weight is superior to both methods. On Soybean, Breast, and Mashroom, Ini_Weight yields AC, PR, and RE higher than or equal to these two methods. On Zoo, although Ini_Weight's PR is better than Cao, Wu is low, but it is higher than those 2 methods on AC and RE. In addition, Ini_Weight performs better than Khan's because on any data set, Ini_Weight yields higher AC, PR, and RE than Khan's.

We can also see that Ini_Weight is always higher than other methods on RE. This shows that our method can strictly control the objects in a cluster without being wrongly assigned to other clusters. In addition, our method produces poorer PR on Zoo than Cao, Wu, and random methods, but better on AC. This is because PR and AC are two different performance metrics. In the clustering results obtained by Ini_Weight, one of the clusters has zero accuracy (i.e., no objects in Zoo are allocated to this cluster correctly), so that the PR obtained by Ini_Weight is lower than that by Cao, Wu, and random methods. However, although the accuracy of the cluster is zero, the number of objects belonging to this cluster in Zoo is very small, so Ini_Weight is still higher on AC than Cao, Wu, and random methods. We also compared the accuracy of several algorithms, as shown in Figure 4.

4. Application of K-Modes Clustering Algorithm Based on Weighted Overlap Distance in Intrusion Detection

We used the network test environment shown in Figure 5 for testing.

Under normal circumstances, Host 5 uses Tcpdump to collect 10 minutes of network packets every 1 h. The data collection was performed six times in succession. The first five sets of data were merged to form the standard normal behavior set dataNormal, and the sixth set of datasets was the normal behavior set dataNatural. Then the distributed denial of service attack tool trino is installed on host Host 1, and trino uses master to control host Host 2 and host Host 3 to launch an attack on host Host 5. The network packet of 10 min is also collected to form the abnormal behavior set data Abnoraml (Algorithm 2).

The clustering process is shown in Figure 6.

During the experiment, we used four attack types. For each of the four types of attacks, random and non-returning are performed according to different proportions. Each type of attack extracts a certain amount accordingly, and the corresponding category attribute is also removed. The attack datasets of D, P, R, and U are, respectively, obtained, and the attack records of DOS, Probe, R2L, and U2R types are, respectively, stored in D, P, R, and U. The proportion of the four attack types corresponding to the extraction and the corresponding number of records are shown in Table 3.

In the experiment, we set the threshold μ to the proportion of abnormal behavior in the entire training set. In order to verify the performance of the UIDM_WODKM

Input: Information table IS = (*U*, *A*, *V*, *f*), where *U* = {*x*1, ..., *x*n}, *A* = {*a*1, ..., *a*m}; the number of clusters *k* expected.
Output: *k* initial center points.
Input: Information table IS = (*U*, *A*, *V*, *f*), where *U* = {*x*1, ..., *x*n}, *A* = {*a*1, ..., *a*m}; the number of clusters *k* expected.
Output: *k* initial center points.
Initialization: Let $C = \Phi$, where C is the initial set of center points that have been selected
(1) Calculate the division U/IND (A-{a}) and U/IND ({a}) respectively by counting sorting;
(1.1) Calculate the information entropy *E* (A-{a}) of IND (A-{a});
(1.2) Calculate the importance of the attribute a Sig (a), and thus obtain the weight of a weight (a);
(1.3) For any $x \in$ U, calculate |[x]{a}| according to the division U/IND ({a}), and
(2) Calculate WDens (*x*) for any $x \in$ U;
(3) Select the object *y* with the largest weighted average density from U as the first initial center, and $C = C\{y\}$;
(4) If |C| < *k*, go to step (5), otherwise go to step (10);
(5) Assume that $C = \{c1, c2, ..., cq\}$, repeated for any $x \in$ U -C
(5.1) Calculate the weighted overlapping distance wd (x, ci) of *x* and ci, where ci ∈ C, $1 \le i \le q$;
(5.2) Calculate Pos_ Center (x);
(6) Select the object *y* that is the most likely to be the initial center from U-C as the new initial center.
And let $C = C\{y\}$;
(7) If |C| < *k*, go to step (5), otherwise go to step (8);
(8) Return *k* initial centers in *C*.

Algorithm 1: The initial center selects the Ini_Weight algorithm.

Table 1: UCI datasets.

Datasets	Soybean	Zoo	Breast	Mushroom
Number of categories	4	7	2	2
Number of objects	47	101	699	8124
Number of attributes	35	16	9	22
With or without missing values	No	No	Yes	Yes

Table 2: Initialization results of different algorithms.

Datasets	Clustering	Initializing methods				
		Random	Cao	Wu	Khan	Ini_Weight
Soybean	AC	0.8356	0.8812	0.8812	0.8911	0.9406
	PR	0.8186	0.8702	0.8702	0.7224	0.7676
	RE	0.6123	0.6714	0.6714	0.7716	0.8143
Zoo	AC	0.8356	0.8812	0.8812	0.8911	0.9406
	PR	0.8186	0.8702	0.8702	0.7224	0.7676
	RE	0.6123	0.6714	0.6714	0.7716	0.8143
	AC	0.8461	0.9113	0.9113	0.9127	0.9385
Breast	PR	0.8700	0.9292	0.9292	0.9318	0.9479
	RE	0.7833	0.8773	0.8773	0.8783	0.9167
	AC	0.7318	0.8754	0.8754	0.8815	0.8858
Mushroom	PR	0.7520	0.9019	0.9019	0.8975	0.9080
	RE	0.7278	0.8709	0.8709	0.8780	0.8817

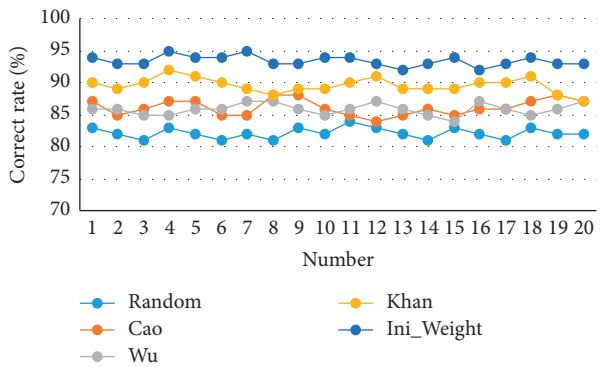

Figure 4: Comparison of the accuracy of different algorithms.

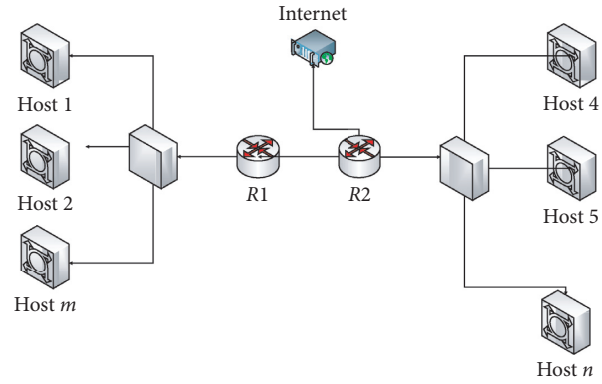

Figure 5: Test network topology.

Input: Information table IS = (*U*, *A*, *V*, *f*), where *U* = {*x*1,..., *x*n}, *A* = {*a*1,..., am}; the number of clusters *k* expected.
Output: A collection of *k* clusters.
Step 1: according to the previously proposed Ini_Weight algorithm, select *k* objects from the universe *U* as the initial center point: *z*1, ..., *z*k.
Step 2: calculate the weighted overlap distance between each object *x* in *U* and each center point zi, and divide the object *x* into the cluster represented by the nearest center point.
Step 3: for each current cluster *c*, recalculate the center point of *c* based on the frequency of the object's value on each attribute in *c*.
Step 4: repeat steps 2 and 3 until the value of the objective function does not change. After each iteration, let *t* ++.
Step 5: return the clustering results.

Algorithm 2: The clustering algorithm used in the experiment.

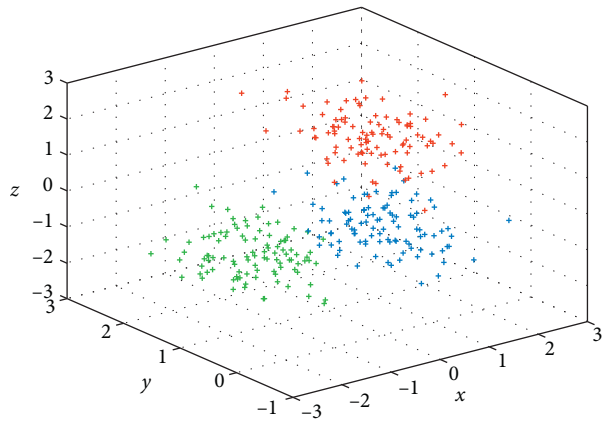

Figure 6: Class process in the experiment.

model in intrusion detection, we compared it with the FCM clustering algorithm and the clustering algorithm proposed by Wu. The FCM algorithm is a fuzzy clustering algorithm based on the objective function, which uses Euclidean distance to measure the similarity of objects. The specific experimental results are shown in Figure 7.

In Figure 5, DR (detection rate) represents the detection rate, and FPR (false positive rate) represents the false alarm rate. By comparing the detection results of the three intrusion detection algorithms listed in the table, the following conclusions can be drawn. The UIDM_WODKM model is better for Probe, U2R, and R2L attacks than the FCM and Wu algorithms. The detection effect of the DOS attack is worse than the FCM algorithm, but it is better than the Wu algorithm. In addition, the overall performance of the UIDM_WODKM model is better than the traditional FCM algorithm and Wu algorithm, especially the UIDM_WODKM model has a much lower false alarm rate than the FCM algorithm. Since Wu's algorithm does not analyze the false alarm rate, we cannot compare it.

Table 3: The percentage and number of four-attack categories.

Attack type	Extraction ratio (%)	Extraction number
DOS	0.311	1217
Probe	24.84	1020
R2L	50	563
U2R	100	52

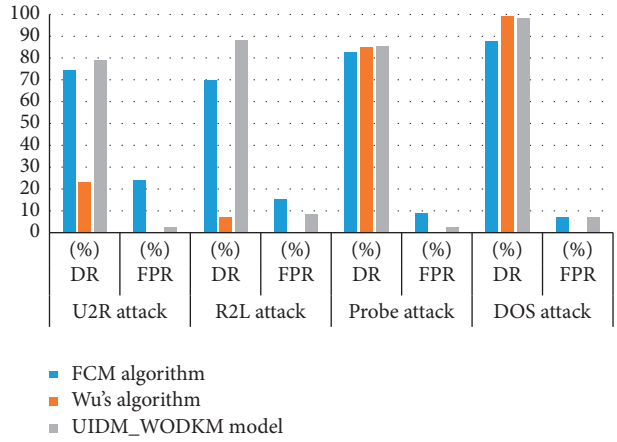

Figure 7: Comparison of results of different algorithms.

5. Conclusion

In this paper, a new distance metric and density metric are proposed based on the concepts of attribute importance and rough entropy in rough set theory, the weighted overlap distance and the weighted average density are used, and an initial class center selection algorithm Ini_Weight based on the weighted density and the weighted overlap distance is proposed [21]. The Ini_Weight algorithm takes full account of the different importance of each attribute so that important attributes are given a larger weight, and unimportant attributes are given a smaller weight. The Ini_Weight algorithm distinguishes the influence of different attributes on the clustering result by the weight of the weight, thus solving the problem of the traditional K-modes algorithm in the initial center point selection. We have carried out experiments on multiple subtype UCI datasets. The experimental results show that the performance of the Ini_Weight algorithm is better than the existing initial center point selection method, which effectively improves the accuracy of the initial center point selection.[24]

Data Availability

All data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] H. Sandberg, S. Amin, and K. H. Johansson, "Cyberphysical security in networked control systems: an introduction to the issue," *IEEE Control Systems*, vol. 35, no. 1, pp. 20–23, 2015.

[2] P. Maji, S. K. Pal, and A. Skowron, "Preface: pattern recognition and mining," *Natural Computing*, vol. 15, no. 3, pp. 355–357, 2016.

[3] D. G. Ferrari and L. N. De Castro, "Clustering algorithm selection by meta-learning systems: a new distance-based problem characterization and ranking combination methods," *Information Sciences*, vol. 301, pp. 181–194, 2015.

[4] R. J. Kuo, Y. Potti, and F. E. Zulvia, "Application of meta-heuristic based fuzzy K-modes algorithm to supplier clustering," *Computers & Industrial Engineering*, vol. 120, pp. 298–307, 2018.

[5] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: a survey," *Journal of Big Data*, vol. 2, no. 1, p. 3, 2015.

[6] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2670–2679, 2015.

[7] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: an intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-based Systems*, vol. 78, pp. 13–21, 2015.

[8] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, 2017.

[9] J. Jabez and B. Muthukumar, "Intrusion Detection System (IDS): anomaly detection using outlier detection approach," *Procedia Computer Science*, vol. 48, pp. 338–346, 2015.

[10] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez, "Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: a feasibility study," *IEEE Systems Journal*, vol. 9, no. 1, pp. 31–44, 2015.
[11] U. Ravale, N. Marathe, and P. Padiya, "Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function," *Procedia Computer Science*, vol. 45, pp. 428–435, 2015.
[12] S. A. M. Ramona, C. M. Pompiliu, and S. L. Constantin, "Attainment of K-means algorithm using hellinger distance," *Economic Sciences Series*, vol. 17, no. 2, pp. 324–329, 2017.
[13] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 132–164, 2015.
[14] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
[15] S. Haben, C. Singleton, and P. Grindrod, "Analysis and clustering of residential customers energy behavioral demand using smart meter data," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 136–144, 2016.
[16] L. Chen, S. Wang, K. Wang, and J. Zhu, "Soft subspace clustering of categorical data with probabilistic distance," *Pattern Recognition*, vol. 51, pp. 322–332, 2016.
[17] D. L. Huerta-Muñoz, R. Z. Ríos-Mercado, and R. Ruiz, "An iterated greedy heuristic for a market segmentation problem with multiple attributes," *European Journal of Operational Research*, vol. 261, no. 1, pp. 75–87, 2017.
[18] G. Gan and M. K.-P. Ng, "k -means clustering with outlier removal," *Pattern Recognition Letters*, vol. 90, pp. 8–14, 2017.
[19] K. A. Prabha and N. K. K. Visalakshi, "Particle swarm optimization based K-prototype clustering algorithm," *IOSR Journal of Computer Engineering*, vol. 17, pp. 56–62, 2015.
[20] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm and Evolutionary Computation*, vol. 16, pp. 1–18, 2014.
[21] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998.
[22] F. Cao, J. Liang, and L. Bai, "A new initialization method for categorical data clustering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10223–10228, 2009.
[23] S. Wu, Q. S. Jiang, and Z. X. Huang, "A new initialization method for clustering categorical data," in *Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Nanjing, China, May 2007.
[24] G. Wang, X. a. Ma, and H. Yu, "Monotonic uncertainty measures for attribute reduction in probabilistic rough set model," *International Journal of Approximate Reasoning*, vol. 59, pp. 41–67, 2015.