Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 3, pp. 1403–1411, DOI 10.12694/scpe.v26i3.4111

IMPROVEMENT AND OPTIMIZATION OF MACHINE LEARNING ALGORITHMS BASED ON INTELLIGENT COMPUTING

JINDI FU*, XIAOJIE LIU†, PENGKAI MA‡, AND CHANGXIN SONG§

Abstract. In this paper, a sensor cloud data intrusion detection framework is proposed. The framework uses parallel discrete optimization techniques for feature refining and incorporates machine learning principles to improve sensing cloud security. Firstly, a set of optimal feature evaluation criteria is established, and a parallel discrete optimization feature extraction system is built to reduce the data dimension and strengthen the stability of feature processing. Then, a widely used discrete optimization algorithm is developed, proving its global convergence. The optimal feature set is obtained through parallel screening feature subsets. Finally, using these features and distributed fuzzy cluster analysis, the intrusion behavior of the sensing cloud is accurately detected. This method incorporates the concept of intelligent iterative evolution and self-regulating clustering strategy, which not only overcomes the local optimal trap that the conventional fuzzy clustering algorithm may encounter but also realizes the automatic adjustment of the number of clusters. The experimental data show that the intrusion detection algorithm performs excellently in providing accurate intrusion determination results. Compared with other detection algorithms, the accuracy of anomaly detection and the reduction of missing detection rate is significantly improved. In addition, the algorithm shows anti-interference solid ability and can maintain stable performance in noisy environments.

Key words: Intelligent computing; Sensing cloud data; Discrete optimization algorithm; Machine learning; Algorithm improvement; Intrusion detection.

1. Introduction. The purpose of intrusion detection is to detect all kinds of attack intentions quickly and accurately and to respond to them quickly. Abuse recognition and anomaly detection are two standard attack methods. Anomaly detection using machine learning technology can effectively identify potential attacks, so it has received more and more attention. Given the massive perception networks existing in large-scale and high-dimensional perception networks, feature extraction technology is used in the literature [1] to extract highly discriminative feature subsets from the perception networks. Scholars maximize the recognition accuracy to reduce the complexity of the problem. In literature [2], the Markov chain method extracts feature subsets efficiently by combining the maximum information coefficient with the symmetric uncertainty criterion. Literature [3] proposes a criterion based on information increment. Irrelevant and redundant features are gradually eliminated, and the target classification is realized. The above feature extraction method needs to be completed in steps, so it is challenging to meet the real-time processing requirements in the processing process. So, scholars began to use neural networks, rough sets, support vector machines, cluster analysis and other machine learning features to realize the recognition of intrusion behavior. In the literature [4], the FCM algorithm is applied to sensor networks, and an improved fuzzy clustering method, AGFCM, is proposed accordingly. Literature [5] proposes a multi-attribute fusion model based on a genetic algorithm. Literature [6] introduces the learning technique of eliciting C-means to improve its ability to identify sample sets correctly. However, it is easily affected by the initial cluster center, local extreme value, cluster number presetting, etc. Literature [7] established a graphical signal model according to the directional characteristics of each sensor. The smoothness ratio of the image is obtained by processing the image with a low-pass filter. It evaluates the anomalies by statistical inspection of the data in the network and combined with the decision threshold. However, this algorithm needs to train many samples, which significantly disadvantages improving the algorithm's operation speed. Literature [8] uses K-Means to cluster data in the network and uses the K-nearest neighbor algorithm to transmit standard

*Keyi College, Zhejiang Sci-Tech University, Shaoxing 312369, China
†Shanghai Urban Construction Vocational College, Shanghai 201415, China
‡Hefei University of Technology, Xuancheng 242000, China
§Shanghai Urban Construction Vocational College, Shanghai 201415, China (Corresponding author, songcx321@163.com)

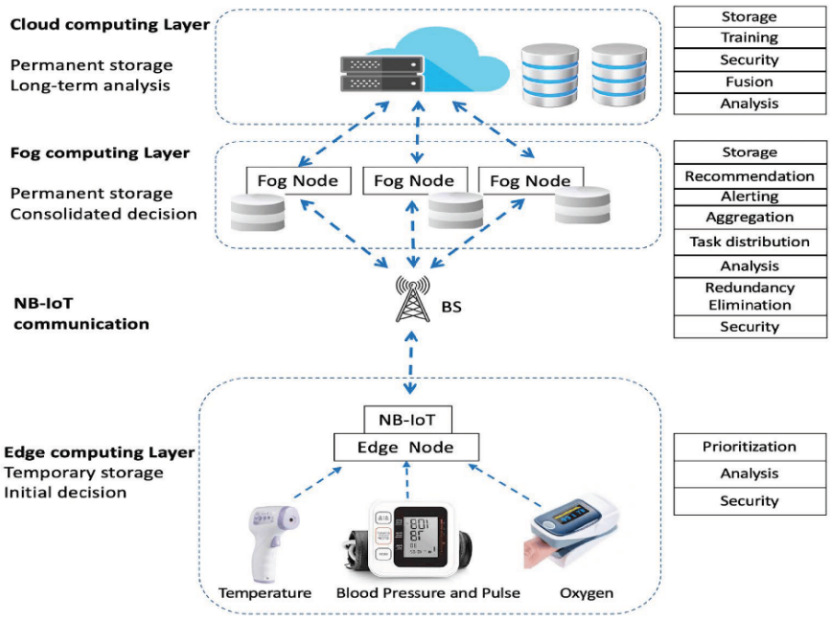

Fig. 2.1: Sensor Cloud architecture based on fog computing.

clustering information to lower-level nodes to partition abnormal data. However, this algorithm focuses on the study of spatial correlation of data and does not effectively combine the statistical characteristics of traffic with it. This results in low accuracy of anomaly recognition. This paper establishes a new perceptual cloud cluster model by combining distributed fuzzy clustering and parallel discrete optimal methods to achieve effective and reliable intrusion detection.

2. Perception-oriented fog source cloud system structure. A cloud-sensing architecture based on fog computers is established. Cloud computing has enormous computing functions, which can conduct data mining and analysis for massive data in massive perception networks. This provides a reference for the behavior of various users [9]. The "fog" technology is used to realize network computing extending from "cloud" to "edge." The system comprises multiple mobile nodes with certain computing functions to form a virtual network. It can process, store and manage data autonomously or assist the cloud. Figure 2.1 is a schematic of a perceptive cloud architecture based on fog computing.

2.1. Optimal feature extraction method based on parallel discretization.

2.1.1. Extracting feature subsets. There are n samples $\{u_i\}_{i=1}^{n}$ in the data set S, and each sample u_i can be interpreted by m samples of $G_i = (g_{i1}, g_{i1}, \cdots, g_{im})$. And divide the data set S into z categories $Z = \{z_j\}_{j=1}^{z}$. The goal of feature extraction is to extract a subspace composed of t features from m feature sets, so that it has similar recognition performance to the original data. The feature extraction vector P is:

$$\begin{aligned} &P = (p_1, \cdots, p_j, \cdots, p_m) \\ &p_j \in \{0, 1\}, P^T 1 = t \end{aligned} \tag{2.1}$$

$p_j = 1$ means that the corresponding feature is extracted. If not, $p_j = 0$. $G_i P^T = \sum_{j=1}^{t} \hat{g}_{ij}$ is obtained after the feature of vector P of u_i is extracted, where E is the corresponding characteristic description after the feature of u_i is extracted [10]. The feature extraction matrix $(\hat{g}_{i1}, \hat{g}_{i2}, \cdots, \hat{g}_{it})$ is determined. There is a corresponding relationship between the extraction matrix E and P, and the element containing only t lines in the extraction

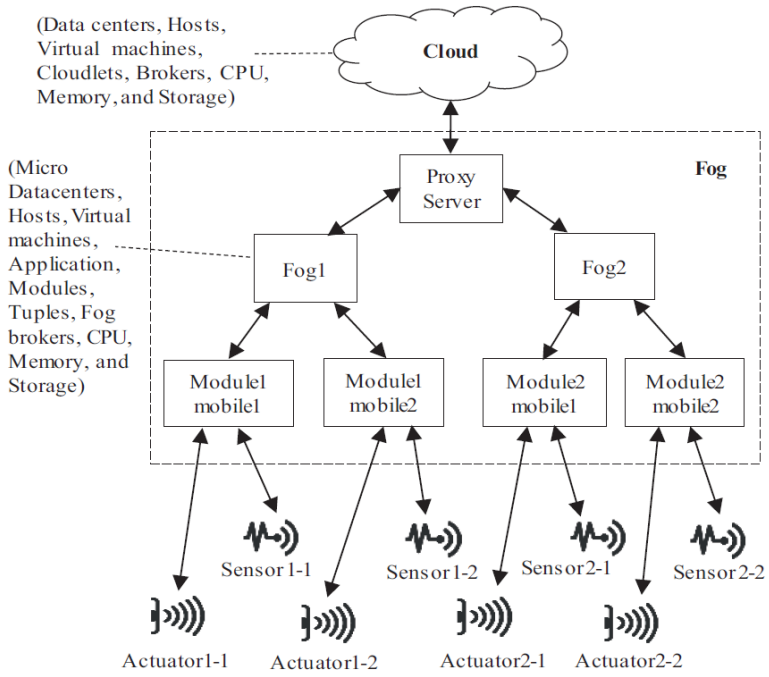

Fig. 2.2: Schematic diagram of feature subset extraction.

matrix E is set to 1 :

$$E_{m\times\Sigma} = \left(P^T, P^T, \cdots, P^T\right) = \begin{bmatrix} p_1 & p_1 & \cdots & p_1 \\ p_2 & p_2 & \cdots & p_2 \\ \vdots & \vdots & & \vdots \\ p_m & p_m & \cdots & p_m \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \|E\|_{2,0} = t \tag{2.2}$$

The eigensubset $\hat{G} = \{\hat{g}_1, \hat{g}_2, \cdots, \hat{g}_t\}$ is extracted by the feature extraction matrix E. Figure 2.2 depicts a specific graph of the E extraction S feature (image cited in Low-latency and energy-efficient scheduling in fog-based IoT applications). Visualization $\hat{G} = \{\hat{g}_1, \hat{g}_2, \cdots, \hat{g}_t\}$ reflects the non-o elements of vector P corresponding to the characteristic value of the sample, then the characteristic subset of the data sample can be extracted [11]. To better evaluate the feature subset G, the evaluation index $\phi(S)$ of the best feature subset is determined.

The best feature subset evaluation index $\phi(S)$ is:

$$\phi(S) = \min_P \left\| \frac{\varphi^T(\Lambda E)(\Lambda E)^T\varphi}{n^2} - C \right\|_G^2 \tag{2.3}$$

where $C = (c_{ij})_{z\times x}$ is a similar matrix between groups. Λ is a matrix of $n \times m$. Where $\varphi = (\lambda_{ij})_{n\times x}$ is the association coefficient between the attribute and the class. $\lambda_{ij} \in [0, 1]$ represents the correlation between sample u_i and class z_j. Use the method of maximum information to solve the λ_{ij} problem. Variables are expressed in terms of $U = \{u_i, i = 1, 2, \cdots, N\}$ and $V = \{v_i, i = 1, 2, \cdots, N\}$, and defined in terms of mutual information about U and V

$$MI(U, V) = \sum_{u\in U}\sum_{v\in V} p(u, v) \lg \frac{p(u, v)}{p(u)p(v)} \tag{2.4}$$

where $p(u, v)$ is the combination possibility of U and V.$p(u), p(v)$ is a function of the marginal probability. The value range of U and V is divided into two parts: c and d. The combined space of U and V is divided into

$c \times d$ grids, and then the histogram is used to estimate $p(u), p(v)$, and the estimate $MI(U,V)_{c,d}$ of $MI(U,V)$ is obtained. According to the different types of class $c \times d$ grid, the maximum mutual information $MI(U,V)_{c,d}^{\max}$ is introduced in Class $c \times d$ grid division. The maximum information factor is defined as:

$$MI(U,V) = \max_{c \times d \leq D(N)} \left\{ \frac{MI(U,V)_{c,d}^{\max}}{\log(c,d)} \right\} \tag{2.5}$$

$D(N)$ represents the maximum value of the grid, which is generally represented by $D(N) = N^{0.6}$. Select $\lambda_{ij} = MIC\,(u_i, z_j)$ when determining the more significant amount of information.

Proof that all the features in $\Lambda_{n \times m}$ are composed of a standardized central program, namely $\sum_{i=1}^{n} g_{ij} = 0, \sum_{i=1}^{n} g^2 ij = 1$. If $C' = n^2 C, D = \varphi^T (\Lambda E)(\Lambda E)^T \varphi$, then:

$$\begin{aligned} &\phi(S) = \min_P \frac{1}{n^2} \left\| D - C' \right\|_G^2 = \\ &\min_P \frac{1}{n^2} \operatorname{tr}\left[(D - C')\,(D - C) \right] \Rightarrow \\ &\phi(S) = \min_P \frac{1}{n^2} \operatorname{tr}\left(D^T D + C'^T C' - 2C'D \right) \end{aligned} \tag{2.6}$$

Since $C'TC'$ is a constant matrix, the minimum of $\phi(S)$ requires both $\min_P \operatorname{tr}\left(D^T D\right)$ and $\min_P \operatorname{tr}\left(C^T D\right) \cdot \min_P \operatorname{tr}\left(D^T D\right)$ is represented as follows:

$$\begin{aligned} &\min_P \operatorname{tr}\left(D^T D\right) = \sum_{i,j=1}^{t} \left[\left(\hat{g}_i^T \varphi\right) \left(\hat{g}_j^T \varphi\right) \right]^2 = \\ &\sum_{i,j=1}^{t} \left(\hat{g}_i^T \left(\varphi \varphi^T \right) \hat{g}_j \right) \Rightarrow \min_P \operatorname{tr}\left(D^T D\right) = \\ &\sum_{i,j=1}^{t} \sum_{h}^{z} \left(\langle \hat{g}_i, R_h \rangle \times \langle \hat{g}_j, R_h \rangle \right)^2 \Rightarrow \\ &\min_P \operatorname{tr}\left(D^T D\right) = \sum_{i,j=1}^{t} \sum_{h}^{=} n^4 \delta_{R_h}^4 \xi_{g_i, R_h}^2 \xi_{\hat{g}_j, R_h}^2 \end{aligned} \tag{2.7}$$

$R_h (h = 1, 2, \cdots, z)$ is the element of the h line corresponding to the matrix $\varphi_{nx} =$ of class $z_h; \delta_{R_h}^2$ is the standard deviation of $z_h; \xi_{\hat{g}_i, R_h}$ $\left(\xi_{\hat{g}_j, R_h}\right)$ is the Pearson correlation between $\hat{g}_i\,(\hat{g}_j)$ and z_h. $\sum_h^z \xi_{\hat{g}_i, R_h}^2 \xi_{\hat{g}_j, R_h}^2$ reflects the redundancy of the corresponding characteristics of $\hat{g}_i$ and $\hat{g}_j$, and the minimum of $\min_P \sum_h^z \xi_{\hat{g}_i, R_h}^2 \xi_{\hat{g}_j, R_h}^2$ is required for $\min_P \operatorname{tr}\left(D^T D\right) = \sum_{i,j=1}^{t} \sum_h^z n^4 \delta_{R_k}^4 \xi_{\hat{g}_i, R_i}^2 \xi_{\hat{g}_j, R_h}^2$ to be optimal. That is, each element has the least redundancy. Where $\max_P \operatorname{tr}\left(C'T\right)$ is:

$$\begin{aligned} &\max_p \operatorname{tr}\left(C^\tau D\right) = (\Lambda E)^T \varphi C' \phi^T (\Lambda E) = \\ &\sum_{i=1}^{t} \hat{g}_i^T \left(\varphi C' \varphi^\tau \right) \hat{g}_i \Rightarrow \max_p t^r \left(C^{-T} D \right) = \sum_{i=1}^{t} \hat{g}_i^T \left(\sum_{i=1}^{2} \sum_{i=1}^{2} R_e c_o R_i^T \right) \hat{g}_i \end{aligned} \tag{2.8}$$

$\sum_{i=1}^{2} \sum_{l=1}^{2} R_e c_{er} R_i^T$ fully embodies the similarity between attributes and the correlation between attributes and categories. This preserves the original category associations to a maximum extent. The feature extraction vector P and matrix E correspond [12]. After all the positions of $p_j = 1$ in P are determined, a specific representation of E can be obtained. Then, feature extraction is carried out.

2.1.2. Construct the model of feature subset extraction. The iterative, evolutionary algorithm is adopted to solve the optimal problem by imitating the life activity of organisms or the physical property change of materials. This project proposes a discrete optimization method suitable for the existing intelligent optimization methods [13]. It is used to solve $\phi(S)$ to obtain the optimal feature extraction vector P. This project intends to use the parallel algorithm and Q-value discrete optimal method to classify $\mathrm{Q}-P_i, i=1,2,\cdots,Q$ multidimensional data to obtain more robust and better classification effect. This can improve the robustness and reliability of the algorithm. This paper adopts P_{bost} language and distributed fuzzy clustering method. An initial particle population $H=\{U_i\}_{i=1}^{N}$ with N scales is randomly generated in the solution space L^m. Each particle $U_i=(u_{nl},u_{i2},\cdots,u_{im})$ is a possible solution that evolves repeatedly until it reaches an optimal solution.

In this discrete optimization algorithm, the correction method for particle $U_i(t)$ is as follows:

$$U_i(t+1)=\begin{cases} U_i(t)+\Delta, l_1\geq\kappa \\ U_i(t), l_1<\kappa \end{cases}$$
$$\Delta=\begin{cases} \eta_1\otimes\left(U_i(t)\leftrightarrow U_i(t)\right), l_2\leq\alpha_1 \\ \eta_2\otimes\left(U_i(t)\leftrightarrow U_{b=2}(t)\right)+l_2\otimes \\ \left(U_i(t)\leftrightarrow U_8(t)\right), \alpha_1<l_2\leq\alpha_2 \\ \eta_3\otimes\left(U_i(t)\leftrightarrow U_j(t)\right), \alpha_2<l_2<1, i\neq j \end{cases} \tag{2.9}$$

where κ,α_1,α_2 is the Update control probability. l_1,l_2 is any random number in the interval $(0,1)$; $C\leftrightarrow D$ stands for individual C research on D; Where $\eta_1\otimes(\leftrightarrow)$ represents the size of the degree of learning, and the larger the value of η_1, the more significant the role of particle D on the evolution direction of C. To improve the algorithm's convergence, this project plans to divide H into several subgroups $H_j, j=1,2,\cdots,O$. Each subpopulation adaptively sets the size of α_1,α_2 and η_1,η_2,η_3 according to the fitness of particles, and then automatically adjusts the size of particle swarm to realize the real-time adjustment of learning objectives and learning intensity of particles [14]. About the existence of subgroups H_j :

$$\begin{cases} \alpha_1\left(\eta_1\right)\propto\ln\left(\frac{\max g(U_i(t))}{g(U_{k=\alpha}(t))}\frac{t}{T_{\max}}+1\right) \\ \alpha_2\left(\eta_2,\eta_3\right)\propto\ln\left(2-\frac{\max_{U_e H_j} g(U_i(t))}{g(U_{\mathrm{bai}}\ (t))}\frac{t}{T_{\max}}\right) \end{cases} \tag{2.10}$$

where $g(\cdot)$ is the objective function value. $T_{\max}$ is the maximum number of iterations. For H_j with better fitness, the particle has a higher self-learning ability. That is, it evolves through its mutations, thus effectively expanding the deep optimization space of the algorithm. For H_j with poor fitness, the particle will tend to the optimal value of the population and the historical optimal value with a greater probability of accelerating the evolution.

2.2. Implementation of sensing cloud intrusion detection. Test case u_i^{bul} selects cluster centers closer to itself as categories, and obtains Z sample categories $S_z^{(\infty)}, z=1,2,\cdots,Z$ after all sample sets are classified [15]. Because there is usually very little data for this to happen, it is very vulnerable when there is only a small amount of data S_z^{not} . To further determine the data anomalies, the categories of suspected $S_{=}^{\mathrm{sax}}$ and ordinary correction data sets are compared sequentially. If the following conditions are met:

$$\min_{H\to Z}\left|\frac{\sum_{=}\left(u_i^{vax}-v_s^{vax}\right)}{|S_{=}^{\mathrm{lom}}|}-\frac{\sum_{H\in SS_N}\left(u_j-v_H\right)}{|S_H|}\leq\theta\right| \tag{2.11}$$

Then it is inevitable that something abnormal has occurred in S_z^{text} . Where Z' is the number of classes in the standard correction data set; v_z^{text}, v_H is the cluster center of detection data classification S_z^{text} and normal data classification S_H respectively. $|S_H|$ represents the internal data size of the class.

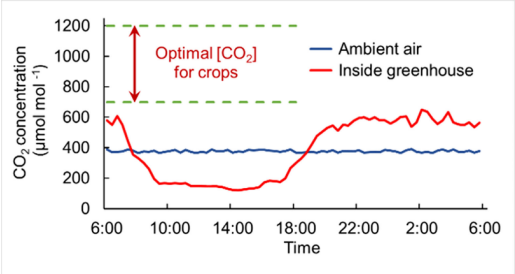

Fig. 3.1: Changes in carbon dioxide concentration values.

3. Simulation experiment.

3.1. Experimental environment. A greenhouse is taken as an example for experimental study. Conventional multi-station environmental parameter measurement system adopts cable transmission mode, which requires cable transmission when transmitting data, which increases the cost of system design, installation and later maintenance [16]. Combine discrete optimization algorithms with machine learning to optimize the wiring of communication and power supply to improve the facility's operational efficiency. The system has important theoretical and practical significance. A wireless sensor network of CO_2 concentration in the greenhouse was established using a discrete optimization algorithm and machine learning algorithm to realize online adjustment of greenhouse gas concentration. The following is the dynamic function model of wireless sensor network monitoring system:

$$x_{CO_1} = RC - B(u) \tag{3.1}$$

RC is the original CO_2 content in the greenhouse and $B(u)$ is the direct effect of photosynthetic efficiency of crops in the greenhouse on soil carbon content. The content of CO_2 showed a dynamic change with the experiment. The content of CO_2 was detected within 2 hours, and the corresponding results were obtained. Figure 3.1 shows the change in C content in the greenhouse. The greenhouse covers an area of 300 m *1000 m and is evenly arranged with 30x100 sensors in different directions. Then MATLAB is used as the test platform. Image signal processing in reference [1] is compared with hierarchical clustering in reference [2].

3.2. Analysis of experimental results. The detection rate refers to the probability of accurately detecting an anomaly, while the false alarm refers to the probability of being detected. A good anomaly detection algorithm should have high and small false favorable rates to ensure the final result's accuracy [17]. The discreteness of a node is the probability of finding an anomaly when a node is in a good state. This project takes the abnormal ratio of nodes in the network as the test target. The abnormal rate ranges from 5% to 25%. The effect of the anomaly ratio of the three algorithms on anomaly detection is shown in FIG. 3.2 and FIG. 3.3.

It can be seen from Figure 3.2 and Figure 3.3 that when the number of detection rates is 25, the method of the invention can obtain a detection rate of 94.48%. The detection rate of graph signal processing was 89.17%. The stratified polymerization method was 84.58%. The detection rate of the three algorithms decreases gradually with the increase of the node dispersion rate, while the false alarm increases continuously [18]. However, the accuracy of the proposed algorithm is significantly higher than that of graph processing and hierarchical clustering because the algorithm uses the discrete optimization algorithm in the cloud computing environment to remove excess information in the data and thus improve the accuracy of anomaly detection. Next, the amount of node energy lost when three methods are used to detect data anomalies is analyzed. As shown in Figure 3.4, node power consumption gradually increases with test times [19]. The proposed algorithm requires the lowest power consumption, followed by hierarchical clustering, and the last is graph signal processing. It is proved that the algorithm proposed in this paper can complete real-time monitoring of sensing data with the lowest power consumption.

Figure 3.5 compares the data anomaly detection time of the three methods. The number of tests is 300, and the comparison of abnormal data detected by the three methods is shown in Figure 3.5 as the average time

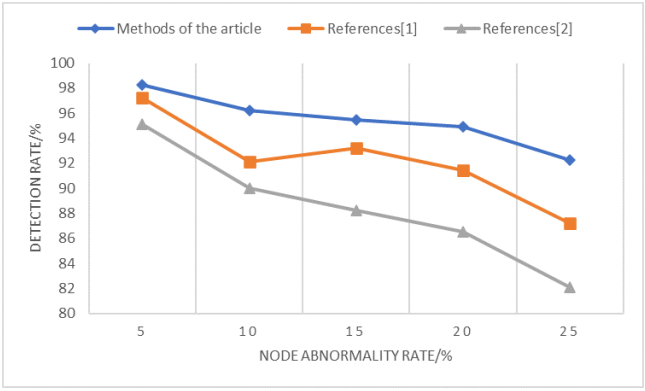

Fig. 3.2: Comparison of data anomaly calculation detection rates.

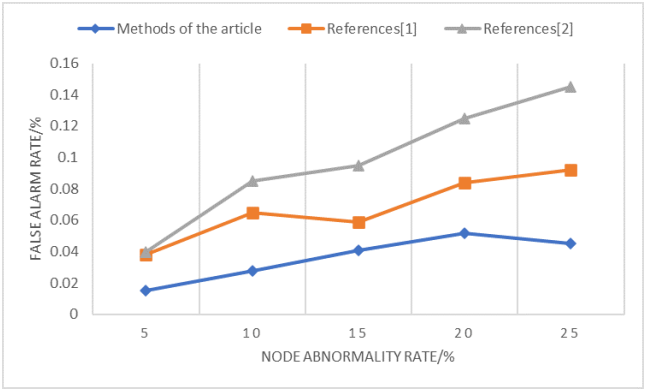

Fig. 3.3: Comparison of false alarm rates of data anomaly detection.

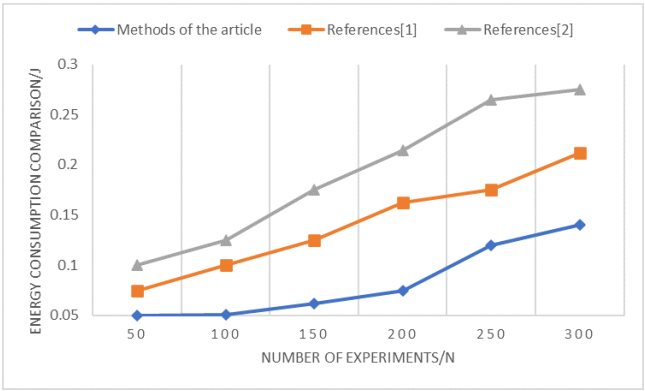

Fig. 3.4: Comparison of energy consumption of abnormal detection nodes.

of each cycle. The algorithm in this paper does not change the anomaly detection time of data and has good stability. Its detection time is minimal and it has a good application prospect.

Fig. 3.5: Comparison of data anomaly detection time.

4. Conclusion. In this paper, an innovative sensor cloud intrusion detection algorithm is proposed, which cleverly combines the characteristics of discrete optimization algorithm (DOA) and machine learning technology and effectively copes with the challenges of large data scale, high dimension and variable intrusion behavior in the sensor cloud environment. The optimal feature combination is extracted by parallel feature subset screening, which further enhances the detection ability of the algorithm. Finally, using distributed fuzzy clustering technology and intelligent iterative evolution ideas, the algorithm realizes the automatic partition of cluster numbers while avoiding local optimization, thus significantly improving the efficiency and accuracy of intrusion detection. By defining the optimal feature evaluation index and constructing the parallel discrete optimization feature extraction framework, this paper successfully reduces the data dimension and improves the robustness of feature extraction, thus laying a solid foundation for subsequent intrusion detection.

REFERENCES

[1] Shen, Y., Liu, Y., Tian, Y., & Na, X. (2022). Parallel sensing in metaverses: Virtual-real interactive smart systems for "6S" sensing. IEEE/CAA Journal of Automatica Sinica, 9(12), 2047-2054.
[2] Lu, H., Zong, Q., Lai, S., Tian, B., & Xie, L. (2021). Flight with limited field of view: A parallel and gradient-free strategy for micro aerial vehicle. IEEE Transactions on Industrial Electronics, 69(9), 9258-9267.
[3] Cong, P., Zhou, J., Chen, M., & Wei, T. (2020). Personality-guided cloud pricing via reinforcement learning. IEEE Transactions on Cloud Computing, 10(2), 925-943.
[4] Balaji, K., Sai Kiran, P., & Sunil Kumar, M. (2023). Power aware virtual machine placement in IaaS cloud using discrete firefly algorithm. Applied Nanoscience, 13(3), 2003-2011.
[5] Srivastava, A., & Kumar, N. (2023). Multi-objective binary whale optimization-based virtual machine allocation in cloud environments. International Journal of Swarm Intelligence Research (IJSIR), 14(1), 1-23.
[6] Cong, P., Zhang, Z., Zhou, J., Liu, X., Liu, Y., & Wei, T. (2021). Customer adaptive resource provisioning for long-term cloud profit maximization under constrained budget. IEEE Transactions on Parallel and Distributed Systems, 33(6), 1373-1392.
[7] Dong, W., Lao, Y., Kaess, M., & Koltun, V. (2022). ASH: A modern framework for parallel spatial hashing in 3D perception. IEEE transactions on pattern analysis and machine intelligence, 45(5), 5417-5435.
[8] Naghdehforoushha, M., Fooladi, M. D. T., Rezvani, M. H., & Sadeghi, M. M. G. (2022). BLMDP: A new bi-level Markov decision process approach to joint bidding andtask-scheduling in cloud spot market. Turkish Journal of Electrical Engineering and Computer Sciences, 30(4), 1419-1438.
[9] Yang, H., & Carlone, L. (2022). Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization. IEEE transactions on pattern analysis and machine intelligence, 45(3), 2816-2834.
[10] Wang, X., Han, S., Yang, L., Yao, T., & Li, L. (2020). Parallel internet of vehicles: ACP-based system architecture and behavioral modeling. IEEE Internet of Things Journal, 7(5), 3735-3746.
[11] Mishra, N., & Singh, R. K. (2020). DDoS vulnerabilities analysis and mitigation model in cloud computing. Journal of Discrete Mathematical Sciences and Cryptography, 23(2), 535-545.
[12] Cong, P., Xu, G., Wei, T., & Li, K. (2020). A survey of profit optimization techniques for cloud providers. ACM Computing Surveys (CSUR), 53(2), 1-35.
[13] Zhou, B., Pan, J., Gao, F., & Shen, S. (2021). Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight. IEEE Transactions on Robotics, 37(6), 1992-2009.

[14] Zhang, Y., Zhou, Y., Lu, H., & Fujita, H. (2020). Traffic network flow prediction using parallel training for deep convolutional neural networks on spark cloud. IEEE Transactions on Industrial Informatics, 16(12), 7369-7380.
[15] Chakraborty, S., Saha, A. K., & Chhabra, A. (2023). Improving whale optimization algorithm with elite strategy and its application to engineering-design and cloud task scheduling problems. Cognitive Computation, 15(5), 1497-1525.
[16] Tang, Q., Fei, Z., Li, B., & Han, Z. (2021). Computation offloading in LEO satellite networks with hybrid cloud and edge computing. IEEE Internet of Things Journal, 8(11), 9164-9176.
[17] Chhabra, A., Singh, G., & Kahlon, K. S. (2021). Multi-criteria HPC task scheduling on IaaS cloud infrastructures using meta-heuristics. Cluster Computing, 24(2), 885-918.
[18] Gualtieri, M., & Platt, R. (2021). Robotic pick-and-place with uncertain object instance segmentation and shape completion. IEEE robotics and automation letters, 6(2), 1753-1760.
[19] Swathi, V. N. V. L. S., Kumar, G. S., & Vathsala, A. V. (2023). Cloud Service Selection System Approach based on QoS Model: A Systematic Review. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2), 05-13.

Edited by: Hailong Li
Special issue on: Deep Learning in Healthcare
Received: Jun 8, 2024
Accepted: Jul 15, 2024