2024 9th International Conference on Automation, Control and Robotics Engineering (CACRE) | 979-8-3503-5030-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CACRE62362.2024.10635055

Novel Methods for Smart Grid Intrusion Detection System Using Feature Selection Based on Improved Gravitational Search Algorithm

Jiahao Li
College of Information Science and Technology
Shihezi University
Shihezi, China
921922959@qq.com

Dingyi Jia
College of Information Science and Technology
Shihezi University
Shihezi, China
j_dingyi1999@163.com

Tao Luo
College of Information Science and Technology
Shihezi University
Shihezi, China
luotao.chn@outlook.com

Jie Zhou*
College of Information Science and Technology
Shihezi University
Shihezi, China
jiezhou@shzu.edu.cn
*Corresponding author

***Abstract*—The smart grid architecture, which represents a deep integration of information technology and power systems, brings many conveniences to people. However, due to the highly open communication network and complex information interaction environment, it also faces more security risks. Existing intrusion detection algorithms based on machine learning cannot cope with the increasing features in the Energy Internet. To address this issue, this paper proposes the Improved Gravitational Search Algorithm (IGSA) for feature selection. Our core idea is to utilize IGSA for efficient feature selection, reducing the learning cost of machine learning methods and improving detection accuracy. Furthermore, to enhance the algorithm's global search capability and robustness, a novel elite selection strategy and adaptive mutation strategy are introduced. Experimental results on three public datasets demonstrate that IGSA improves detection accuracy by an average of 11.14% compared to other feature selection methods.**

Keywords—smart grid, intrusion detection, feature selection

I. INTRODUCTION

The intellectualization and automation of the power grid have impacted an increasing number of related electrical devices and smart applications, which while bringing convenience, have also amplified the threat of cyberattacks on the grid. Furthermore, cyberattacks have become more diversified and complex, posing significant challenges to cyberspace security. To address these issues, the introduction of intrusion detection technology can effectively protect the smart grid. Currently, a mainstream approach for intrusion detection involves the use of machine learning methods for attack identification [1,2]. The application of machine learning algorithms to intrusion detection models can somewhat reduce the tedious work involved in model establishment and optimize the impact of inaccurate modeling on smart grid security research. At present, intrusion detection algorithms based on classic machine learning models such as Support Vector Machine (SVM) [3] and K-Nearest Neighbor (KNN) [4] are gradually being proposed and applied. However, as the feature dimensions and attack categories continue to increase, relying solely on machine learning models becomes insufficient to meet the basic requirements of intrusion detection tasks. To improve the detection accuracy of machine learning-based intrusion detection models, many scholars have employed feature selection methods in conjunction with machine learning for intrusion detection [5,6].

Due to the importance of feature selection, scholars have proposed various feature selection algorithms, which can be broadly categorized into two types: 1) Filter feature selection methods [7,8] and 2) Wrapper feature selection methods [9,10]. Filter methods utilize the inherent properties between features, such as mutual information, with typical algorithms including the Fisher score algorithm. Wrapper methods, on the other hand, employ classifiers to evaluate the quality of selected features, with common algorithms including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC). Although Filter methods are simpler to implement, they overlook the role of features in classifiers. Therefore, this paper opts for the Wrapper method. The main contributions are listed as follows:

1) In this study, an Gravitational Search Algorithm (IGSA) algorithm is introduced that integrates the elitist strategy and adaptive mutation strategy. This integrated approach not only preserves the global search capability of the genetic algorithm but also enhances the precision of local search through simulated annealing techniques. The elitist strategy ensures the inheritance of superior genes, while the adaptive mutation strategy improves the flexibility and adaptability of the algorithm, enabling it to more effectively address complex optimization problems.

2) The IGSA and K-Nearest Neighbor (KNN) algorithms are aptly combined to construct a novel intrusion detection system. By optimizing the parameter selection of KNN through IGSA, the accuracy and efficiency of intrusion detection are significantly improved. This hybrid method fully utilizes the global optimization capabilities of IGSA and the classification accuracy of KNN, providing a novel and effective solution for intrusion detection in smart grids.

3) This study not only presents a novel approach for intrusion detection but also verifies its performance in practical applications through detailed experiments. Real-world network attack datasets are tested and compared with traditional intrusion detection methods. Experimental results demonstrate

significant advantages in key metrics such as detection accuracy and F1 score.

II. Related Work

In recent years, numerous wrapper feature selection methods have been proposed. Gu [9] introduced a feature selection classifier based on genetic algorithms (GA). They compared GA on real datasets with traditional methods such as greedy search, demonstrating the superiority of their algorithm. Since then, a growing number of scholars have focused on implementing GA algorithms for feature selection [10-12]. Particle Swarm Optimization (PSO), as a swarm intelligence optimization technique, has also been applied to feature selection methods [13,14]. Tran [15] conducted a series of representative PSO-based studies, designing various initialization strategies, fitness functions, and search mechanisms to obtain high-quality feature subsets.

Besides the aforementioned GA and PSO-based feature selection algorithms, several other interesting feature selection methods have been proposed, such as Artificial Bee Colony (ABC) [16-19], Ant Colony Optimization (ACO) [20], and Differential Evolution (DE) [21]. However, these algorithms tend to suffer from issues such as easily falling into local optimums and slow convergence rates. The IGSA algorithm proposed in this paper offers advantages of fast convergence and strong global search capabilities, effectively addressing the challenges faced by GA and PSO.

III. Improved Gravitational Search Algorithm

The Gravitational Search Algorithm treats all particles as objects with mass capable of frictionless movement. Each particle is influenced by the gravitational pull of other particles in the solution space, resulting in acceleration towards particles with greater mass. Since the mass of a particle is correlated with its fitness value, particles with higher fitness values have greater mass. Consequently, particles with lower mass gradually approach the optimal solution in the optimization problem as they move towards particles with higher mass. A key characteristic of the Gravitational Search Algorithm is that particles do not rely on environmental factors to perceive the situation in their environment. Instead, they share optimization information through the interaction of gravitational forces among individuals. Therefore, without the influence of environmental factors, particles can still perceive the global situation and search the environment effectively.

Assuming a *D*-dimensional search space containing *N* particles, the position of the *i*-th particle is denoted as:

$$X_i = (x_i^1, x_i^2, \ldots, x_i^k, \ldots, x_i^d); i = 1, 2, \ldots, N \tag{1}$$

where x_i^k represents the position of the *i*-th particle in the *k*-th dimension.

A. Calculation of Inertial Mass

In the GSA algorithm, the mass of each particle is closely correlated with the fitness value obtained through particle information. During the *t*-th iteration, the mass of particle X_i is denoted as $M_i(t)$, as shown in formula (2). Since the mass *M* is calculated based on its corresponding fitness value, particles with greater mass are closer to the optimal position in the entire computational space, naturally exerting a stronger gravitational pull on other objects.

$$\begin{cases} m_i(t) = \dfrac{fit_i(t) - worst(t)}{best(t) - worst(t)} \\ M_i(t) = m_i(t) / \sum_{j=1}^{N} m_j(t) \end{cases} \tag{2}$$

In the formula, $fit_i(t)$ represents the fitness value of particle Xi, which is calculated using the objective function in the algorithm. $best(t)$ denotes the best value achieved during the *t*-th iteration, while $worst(t)$ represents the worst value at that iteration. $M_i(t)$ is the normalized particle mass, obtained by calculating the percentage of each particle's mass relative to the total mass of all particles.

For maximization problems, the formulas for calculating $best(t)$ and $worst(t)$ are as follows:

$$\begin{cases} best(t) = \max\limits_{i \in \{1,2,\ldots,N\}} fit(t) \\ worst(t) == \min\limits_{i \in \{1,2,\ldots,N\}} fit(t) \end{cases} \tag{3}$$

Based on the law of universal gravitation, the gravitational force exerted by particle j on particle *i* in the *k*-th dimension during the *t*-th iteration can be calculated using formula (4).

$$F_{ij}^k(t) = G(t) \frac{M_i(t) M_j(t)}{\| X_i(t), X_j(t) \|} (x_j^k(t) - x_i^k(t)) \tag{4}$$

In the formula, $G(t)$ represents the gravitational constant, as specified in formula (5), while $\| X_i(t), X_j(t) \|$ denotes the Euclidean distance between particle *i* and particle *j*.

$$G(t) = G_0 \cdot e^{-\alpha t / T} \tag{5}$$

In the formula, G_0 represents the initial gravitational constant with a value of 100, α is a constant with a value of 20, and *T* denotes the maximum number of iterations.

The total force acting on particle X_i in the *k*-th dimension equals the randomly weighted sum of forces exerted by all other particles.

$$F_i^k(t) = \sum_{j=1, j \neq 1}^{N} r \cdot F_{ij}^k(t) \tag{6}$$

In the formula, r represents a random number between [0,1], indicating the random weight.

The acceleration of a particle can be calculated according to Newton's second law of motion, as shown in formula (7).

$$a_i^k(t) = \frac{F_i^k(t)}{M_i(t)} \tag{7}$$

This acceleration, calculated based on the resultant force acting on the particle and its mass, is used to update the particle's velocity and position according to formula (8).

$$\begin{cases} v_i^k(t+1) = r \cdot v_i^k(t) + a_i^k(t) \\ x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \end{cases} \tag{8}$$

In the formula, x_i^k represents the k-dimensional component of the position of the *i*-th particle, and v_i^k denotes the *k*-dimensional component of the velocity of the *i*-th particle.

B. Elite Strategy

The elite strategy refers to selecting the best-performing particles among the group to participate in the force and acceleration updates, aiming to prevent a decrease in the precision of convergence results caused by poorly performing particles. To avoid falling into local optima, the influence of more particles is considered during the initial iterations. However, as the iterations progress, the number of elite particles gradually decreases, meaning that poorly performing particles are phased out, and only the best-performing particles are involved in the updates towards the end of the iterations. Consequently, the number of elite particles is negatively correlated with the number of algorithm iterations, as determined by formula (9).

$$num = round(N \cdot (num_{last} + (1 - \frac{t}{T}) \cdot (\frac{1 - num_{last}}{100}) \tag{9}$$

C. Adaptive Mutation

Through analysis of the GSA algorithm, it becomes evident that as the number of iterations increases, the magnitude of particle position shifts significantly diminishes. Hence, an adaptive mutation strategy is proposed to enhance the algorithm. Simultaneously, the mutation trigger rate should decrease as the iterations progress. Initially, a higher mutation rate can be employed to augment population diversity and strengthen the algorithm's global search capabilities. However, in later stages of the algorithm, reducing the mutation rate prevents the destruction of superior individuals, potentially leading to better precision. The expression for the mutation trigger function is given by:

$$TR_i^t = e^{(-\frac{\dim}{2} \times \frac{1}{T})} \times (r + \varepsilon) \tag{10}$$

In the formula, TR_i^t represents the mutation trigger value for the i-th particle during the *t*-th iteration. When $TR_i^t > 0.5$, mutation is triggered, and performing mutation operations on the particle at this point can achieve better results. Here, r is a random number between [0,1], dim represents the dimension of the solution space, and ε is a small constant, taken as 0.1 in this context.

The formula for updating the particle position through uniform mutation is shown in equation (11)

$$x_i(t) = (1 + r)x_i(t) \tag{11}$$

where *r* represents a uniformly distributed constant in the range of [0, 1].

D. Evaluation Fitness

The objective of IGSA is to reduce the number of features, in conjunction with the KNN classifier, to enhance the accuracy of intrusion detection.

The evaluation metric for this experiment is accuracy: Accuracy is a metric used to evaluate the performance of classification models. It represents the proportion of correctly predicted samples by the model out of the total number of samples. A higher accuracy indicates stronger predictive capabilities of the model.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Here, *TP* represents true positives, *TN* stands for true negatives, *FP* denotes false positives, and *FN* indicates false negatives.

IV. Experiment

The equipment and software used in this experiment include an Intel 12700H CPU, 32GB of RAM, MATLAB 2022, and the Windows 11 operating system.

For this experiment, we utilized the KDDCUP99 dataset (1999), the CICIDS dataset (Canadian Institute for Cybersecurity (CIC), 2018), and the UNSW-NB15 dataset (Moustafa and Slay, 2015).

To validate the performance of the IGSA, this paper conducts comparative experiments using the basic Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC) algorithms. The population size *N* and the maximum iteration count *itermax* are set as common parameters for each algorithm. Specifically, *N* is set to 100, *itermax* is set to 300, and each algorithm is independently run 30 times. In addition, the selected feature subsets are all evaluated for performance on a KNN classifier with $k = 2$.

A. Performance Evolution

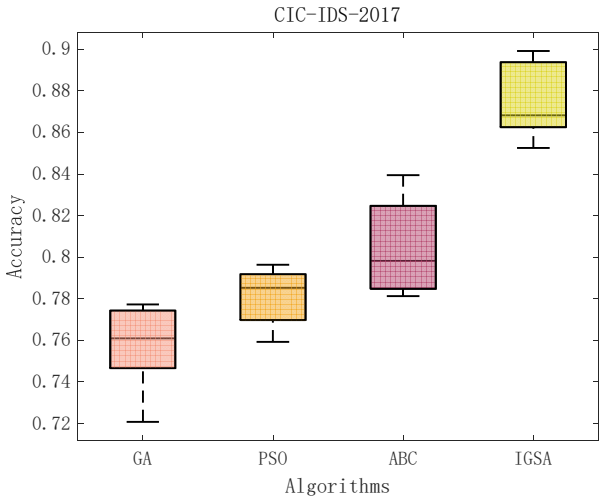
Table 1 presents a performance comparison of different feature selection methods on the KDDCup99, CIC-IDS-2017, and UNSW-NB15 datasets. The evaluation is primarily based on two key metrics: accuracy and F1-score. On the KDDCUP99 dataset, the ISGA algorithm demonstrates significant advantages, improving accuracy by 10.20%, 8.97%, and 13.49% compared to GA, PSO, and ABC, respectively. In terms of F1-score, ISGA also excels, showing an improvement range of 10.10% to 15.73% compared to the other three algorithms. On the CIC-IDS-2017 dataset, ISGA's performance advantage is even more pronounced. Its accuracy increases by 16.96%, 13.79%, and 8.87% relative to GA, PSO, and ABC, respectively. In the comparison of F1-scores, ISGA's improvement is even more significant, especially when

compared to GA, with an increase of up to 20.03%. For the UNSW-NB15 dataset, ISGA also outperforms other algorithms in all evaluation metrics. Its accuracy is 9.48%, 7.17%, and 11.36% higher than GA, PSO, and ABC, respectively. In terms of F1-score, ISGA's improvement relative to these three algorithms ranges from 10.80% to 14.18%. Based on the experimental results from the three datasets, it is clear that ISGA significantly outperforms the three algorithms (GA, PSO, ABC) in terms of both accuracy and F1-score. This superiority may be attributed to ISGA's balanced global search and local exploitation capabilities in searching for optimal solutions, or its algorithm design may be inherently more suitable for handling such complex problems. Especially on the CIC-IDS-2017 dataset, ISGA's improvement in F1-score relative to GA exceeds 20%, further demonstrating ISGA's excellent performance in handling specific problems.

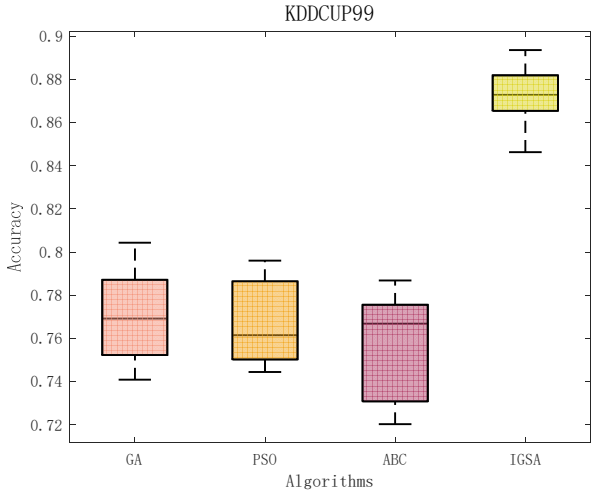
Fig. 1 analyzes the Accuracy scores of various feature selection methods across multiple experimental rounds, examining their respective stability. For instance, the accuracy performance of GA on the CIC-IDS-2017 dataset exhibits a significant distance between the upper and lower bounds of its box plot, with a relatively low median line, indicating a considerable range of fluctuation in Accuracy across multiple tests, representing unstable performance of GA. In contrast, ABC on the CIC-IDS-2017 dataset shows an even greater spread within the box plot, indicating a wider variation in its Accuracy, representing an even more unstable performance. Among the three datasets of KDDCUP99, CIC-IDS-2017, and UNSW-NB15, the boxes representing the performance of IGSA consistently occupy the highest positions overall, with the smallest span. Compared to the other three methods, IGSA demonstrates higher Accuracy, indicating superior and more stable performance.

TABLE I. PERFORMANCE COMPARISON

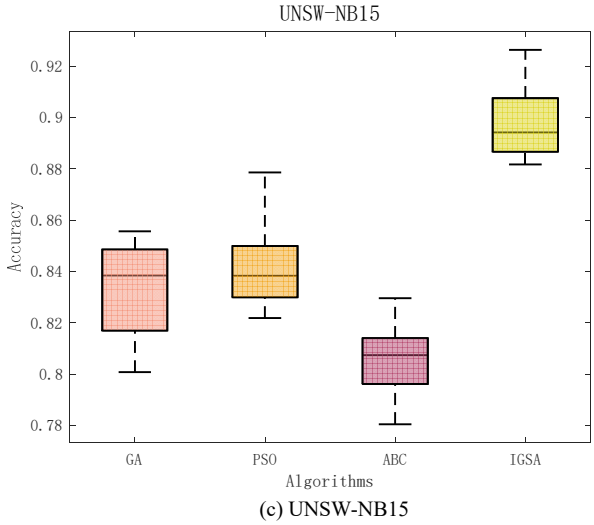
Dataset	***Performance***	***GA***	***PSO***	***ABC***	***IGSA***
KDDCUP99	Accuracy	0.794	0.803	0.771	0.875
	Precision	0.748	0.773	0.737	0.857
	Recall	0.778	0.792	0.752	0.865
	F1-score	0.763	0.782	0.744	0.861
CIC-IDS-2017	Accuracy	0.756	0.777	0.812	0.884
	Precision	0.721	0.742	0.782	0.865
	Recall	0.737	0.757	0.796	0.877
	F1-score	0.729	0.749	0.789	0.871
UNSW-NB15	Accuracy	0.833	0.851	0.819	0.912
	Precision	0.764	0.814	0.792	0.883
	Recall	0.809	0.800	0.774	0.906
	F1-score	0.786	0.807	0.783	0.894



(a) KDDCUP99



(b) CIC-IDS-2017



(c) UNSW-NB15

Fig. 1. Accuracy performance of feature selection algorithms on datasets

CONCLUSION

In this study, an Improved Gravitational Search Algorithm (IGSA) is proposed for selecting relevant features in the intrusion detection system of smart grids, aiming to enhance the accuracy of intrusion detection. By integrating IGSA with the K-Nearest Neighbor (KNN) classifier, a novel approach is created, which outperforms traditional feature selection methods. On three intrusion detection datasets, IGSA achieves a higher level in terms of both accuracy and F1 score. In summary, the IGSA-based feature selection method improves the accuracy and stability of intrusion detection models, highlighting its potential in enhancing the overall security and reliability of smart grid systems.

ACKNOWLEDGMENT

This paper was funded by the project of National Key Research and Development Program of the Corps: Research and application demonstration of information technology based on source-grid-load-storage and multi-energy complementarity, grant number: 2023AB021; National Key Research and Development Program of the Corps: Key Technology Research and Application for High Penetration New Energy Grid Dispatch, grant number: 2023AB010; National Key Research and Development Program of the Corps: Research and application demonstration of key technologies for panoramic monitoring, perception and intelligent control of large-scale distributed photovoltaic bases, grant number: S2023AA5383; National Key Research and Development Program of the Corps: R&D and demonstration of the intelligent supervision platform for the inspection and testing institutions of the Corps based on big data and artificial intelligence, grant number: S2024AB048; Young scientific and technological innovation talents: key technological innovation and application of integrated energy intelligent management and control and precise low-carbon dispatching, grant number: 2023TSYCCx0120; Research on the application of "Internet +" customer service based on the comprehensive energy big data of the XPCC, grant number: KXO0760202.

REFERENCES

[1] Kumar, Chandan, and Md Sarfaraj Alam Ansari, "An explainable nature-inspired cyber attack detection system in Software-Defined IoT applications," Expert Systems with Applications, vol. 250, pp. 123853, 2024.

[2] Almotairi, Ayoob, et al. "Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models," Systems Science & Control Engineering, vol. 12.1 pp. 2321381, 2024.

[3] Mohammadi, Mokhtar, et al. "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," Journal of Network and Computer Applications, vol. 178 pp. 102983, 2021.

[4] Chen, Hongtao, et al. "Intrusion Detection Model Based on KNN-AE-DNN," International Conference on Artificial Intelligence and Security. Cham: Springer International Publishing, 2022.

[5] Li, Jing, et al. "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," Journal of Big Data, vol. 11.1, pp. 36, 2024.

[6] Ngo, Vu-Duc, et al. "Machine learning-based intrusion detection: feature selection versus feature extraction," Cluster Computing, pp. 1-15, 2023.

[7] Solorio-Fernandez, Saul, José Fco Martínez-Trinidad, and J. Ariel Carrasco-Ochoa, "A supervised filter feature selection method for mixed data based on spectral feature selection and information-theory redundancy analysis," Pattern Recognition Letters, vol. 138, pp. 321-328, 2020.

[8] Ba J, Wang P, Yang X, et al. "Glee: A granularity filter for feature selection," Engineering Applications of Artificial Intelligence, vol. 122, pp. 106080, 2023.

[9] Gu, Shenkai, Ran Cheng, and Yaochu Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," Soft Computing, vol. 22, pp. 811-822, 2018.

[10] Liang, Jing, et al, "A multimodal multiobjective genetic algorithm for feature selection," 2022 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2022.

[11] Lee, Seung-Ju, et al. "Genetic algorithm-based feature selection for depression scale prediction," Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019.

[12] Izabela, Rejer, and Lorenz Krzysztof. "GAAMmf: genetic algorithm with aggressive mutation and decreasing feature set for feature selection," Genetic Programming and Evolvable Machines, vol. 24.2, pp.10, 2023.

[13] B. Tran, B. Xue, M. Zhang, and S. Nguyen, "Investigation on particle swarm optimisation for feature selection on high-dimensional data: Local search and selection bias," Connection Sci., vol. 28, no. 3, pp. 270‑294, 2016.

[14] B. Tran, B. Xue, and M. Zhang, "A new representation in PSO for discretization-based feature selection," IEEE Trans. Cybern., vol. 48, no. 6, pp. 1733‑1746, Jun. 2018.

[15] Tran, Binh, Bing Xue, and Mengjie Zhang. "Variable-length particle swarm optimization for feature selection on high-dimensional classification," IEEE Transactions on Evolutionary Computation vol. 23.3, pp. 473-487, 2018.

[16] Mohammadi, F. Ghareh, and M. Saniee Abadeh, "Image steganalysis using a bee colony based feature selection algorithm," Engineering Applications of Artificial Intelligence, vol. 31, pp. 35-43, 2014.

[17] Sarac Essiz, Esra, and Murat Oturakci, "Artificial bee colony‑based feature selection algorithm for cyberbullying," The Computer Journal, vol. 64.3, pp. 305-313, 2021.

[18] Lin, Yanhong, et al. "An improved artificial bee colony for feature selection in QSAR," Algorithms, vol. 14.4, pp. 120, 2021.

[19] Hanbay, K, "A new standard error based artificial bee colony algorithm and its," 2022.

[20] Moradi, Parham, and Mehrdad Rostami. "Integration of graph clustering with ant colony optimization for feature selection," Knowledge-Based Systems, vol. 84, pp. 144-161, 2015.

[21] Varghese, Nelson Vithayathil, et al. "Binary hybrid differential evolution algorithm for multi-label feature selection," IEEE international conference on systems, man, and cybernetics (SMC). IEEE, 2020.