Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 2, pp. 900–907, DOI 10.12694/scpe.v25i2.2587

# INTELLIGENT DETECTION AND ANALYSIS OF SOFTWARE VULNERABILITIES BASED ON ENCRYPTION ALGORITHMS AND FEATURE EXTRACTION

HENG LI*, XINQIANG LI†, AND HONGCHANG WEI‡

**Abstract.** Implement status detection of ship software, identify the source of faults in problematic software, and release new software versions. Based on the above requirements, the author regards the detection and control of ship software status as the core research content. Based on the actual operating environment of ship software, the functional requirements of software status detection were studied and analyzed, and a set of ship software status detection was designed and implemented, a software inspection and maintenance platform that integrates ship software operation and maintenance, as well as ship software version release and update. The author conducted practical verification of the SM3 and SM2 hybrid encryption algorithm and selected software on the ship for detection. After analyzing the experimental results, it has been proven that using a hybrid algorithm for encryption and decryption, the server can accurately obtain software information on the ship's platform, detect the software status on the ship, and locate specific problem files. For software that does not meet the standard status, the server can accurately transmit software information to the "component integration framework" and put the component in a "prohibited" scheduling state. After the server repairs the problematic software, the detection results of the software change and display as legal, while the software is in the "allowed" scheduling state in the "component integration framework".

**Key words:** Encryption algorithm, feature extraction, software vulnerabilities, intelligent detection

**1. Introduction.** With the development of integrated electronic warfare equipment, the proportion of software in equipment is becoming larger and more complex. Large software is usually jointly developed by multiple development units responsible for research and development. This cross unit joint research and development strategy improves the efficiency of large-scale software development. On the other hand, the state of system software becomes difficult to control, and the various software cannot cooperate effectively. Maintenance work after failures is also difficult to carry out. After the delivery of equipment software, unauthorized changes to the software are also difficult to control, and local software changes may lead to unstable operation of the entire system software, affecting the normal use of military equipment [9]. Therefore, there is an urgent need to carry out research on software status detection and maintenance related technologies, achieve monitoring and management of ship software status changes through technical means, solve the problem of uncontrolled ship software status changes, and ensure software quality.A software inspection and maintenance system based on component scheduling, firstly, it can solve the problem of uncontrolled status of ship software, allowing users to clearly observe the status of all controlled ship software on each ship station and discover a list of unauthorized software status changes; Secondly, for problematic software, this system can quickly locate the source of the fault, promptly identify and handle the problem, avoiding maintenance personnel from aimlessly debugging and consuming a large amount of time and cost, enabling the problematic software to quickly resume normal operation, at the same time, it also saves a lot of manpower and resources; Finally, this system provides the only legal software release and update interface, and all ship software is deployed to various ship stations through this system, ensuring the unity of the overall software status of the fleet, standardizing the software release and update methods, simplifying the operation process, and enabling developers to focus more on improving the quality and functionality of ship software. In summary, a software detection and maintenance system based on component scheduling provides an effective solution for the detection and comprehensive management of ship software, which is very necessary. In response to this research issue, Yu, L. and others have applied neural net-

*Software Engineering Department, Shijiazhuang Information Engineering Vocational College, Shijiazhuang, Hebei 050000,China
†Mechanical and Electrical Engineering Department, Shijiazhuang Information Engineering Vocational College, Shijiazhuang, Hebei, 050000, China
‡Software Engineering Department, Shijiazhuang Information Engineering Vocational College, Shijiazhuang, Hebei, 050000, China (Corresponding author)

work technology to binary similarity detection, which has become a promising research topic, and vulnerability detection is an important application field of binary similarity detection. Embedding binary code into matrices using neural networks also requires addressing feature representation issues in vulnerability detection. However, current research mostly extracts the syntax or structural features of binary codes, and uses basic blocks as the minimum analysis unit, which is relatively rough. In addition, the structural characteristics of binary functions are usually represented by dependency graphs. During the embedding process, only the neighbor information of the node is obtained, ignoring the global information of the graph. In order to address these two issues, the author proposes a dual channel feature extraction method to obtain finer grained semantic features and globally represent structural features rather than locally [18]. Duan, L. et al. in intelligent systems, attackers can use botnets to launch different network attack activities against the Internet of Things. Traditional botnet detection methods usually use machine learning algorithms, but due to the imbalance of traffic data in the network, it is difficult to detect and control botnets. The author proposes a novel and efficient botnet detection method based on the cooperation between an autoencoder neural network and a decision tree on a given network. Firstly, the deep flow detection method and statistical analysis are used as feature selection techniques to select relevant features for characterizing the communication related behavior between network nodes. Then, the self encoder neural network is used for feature selection to improve the efficiency of model construction. Finally, the Tomak recursive boundary synthesis minority oversampling technique is used to generate additional minority samples to achieve class balance, and an improved gradient enhanced decision tree algorithm is used to train and establish an anomaly traffic detection model to improve class balance [2].

Based on current research, the main functional objectives of this system are to detect the status of ship software, repair faults, and release and update software versions. Maintain and record software states that do not comply with agreed rules, and prohibit the invocation of that component in the "Component Integration Framework". Backup and save ship software with stable operation and good performance; Release a new version of the software and deploy it to ship stations, or update the old version of the software. Through a software detection and maintenance system platform based on component scheduling, users can accurately grasp the real-time status of ship software, identify problem software, locate problem sources, and strengthen the monitoring ability of ship software;Improve the efficiency of software version release and update, and save resource costs for software status changes; Accurately grasp the software version distribution of the entire fleet and individual ships, issue reasonable combat instructions based on the software status on different ships, and improve coordination and interaction between ships. This system provides comprehensive support for the stable operation of ship software, provides solutions for the management and control of ship software status, and provides technical support for the combat command system. It is of great significance for the control of fleet software and battlefield combat command.

## 2. Methods.

**2.1. SM2 elliptic curve algorithm.** The SM2 algorithm was released by the National Password Administration on December 17, 2010, the asymmetric cryptographic algorithm based on ECC (Elliptic Curve Cryptography) has the characteristics of requiring less private key bit length for operations, lower system parameter requirements, less storage space, lower broadband requirements for data transmission, and lower overall power consumption of the algorithm. Therefore, it can be widely applied to devices with relatively small system scales and severely limited resources [11]. The SM2 algorithm mainly includes three parts of applications: Digital signature algorithm, key exchange protocol, and public key encryption algorithm. With the development and progress of international cryptographic technology, the current 1024 bit RSA algorithm has faced serious security issues, and the elliptic curve cryptography algorithm has its algorithm performance advantages. It has been used as a standard for public key cryptography algorithms in many countries and regions. In order to ensure the security of domestic passwords and improve the security factor, the National Cryptography Administration began researching elliptic curve algorithms with independent intellectual property rights in 2001. After extensive research on public key cryptography algorithms with high recognition in the international cryptology community, learning from advanced cryptographic experience abroad, and absorbing the theoretical foundation of existing elliptic curve algorithms, the SM2 algorithm was successfully completed in 2004, And on December 17, 2010, the SM2 algorithm standard was announced [15]. In March 2011, Bank of China released the relevant specifications for its financial IC card, stating that it uses the SM2 elliptic curve algorithm to enhance the

Table 2.1: Comparison of breakthrough time between RSA algorithm and SM2 algorithm.

| RSA Key Strength | SM2 Key Strength | Breakthrough time (year) |
|---|---|---|
| 512 | 106 | 104, has been breached |
| 768 | 132 | 108, has been breached |
| 1024 | 160 | 1011 |
| 2048 | 210 | 1020 |

Table 2.2: Performance comparison of RSA and ECC algorithms.

| Algorithm | Signature efficiency (times/second) | Verification efficiency (times/second) |
|---|---|---|
| 1024 bit RSA | 2793 | 51225 |
| 2014 bit RSA | 456 | 15123 |
| 256 bit SM2 | 4096 | 872 |

security of IC card applications. At the same time, non-financial applications using PBOC3.0 as a reference standard also use the SM2 algorithm. Comparing and analyzing the two algorithms, SM2 and RSA, it was found that SM2 has higher security than RSA. Under the same key strength, the SM2 algorithm is more difficult to crack and takes longer to break; In terms of computational efficiency, the signature efficiency of SM2 is much higher than that of RSA, while the efficiency of verifying signatures is slower than RSA. The specific comparison between the two algorithms is shown in Tables 2.1 and 2.2:

**2.2. SM3 password hash algorithm.** The hash function, also known as the hash function, its functional feature is that it can output fixed length summary information after a series of changes and processing of an unlimited length message, and the resulting summary result is called a hash value [10, 5]. The hash function has unidirectionality, meaning that its encryption process is easily implemented in a computer, while the decryption process is not feasible in a computer. Therefore, the hash function is considered an irreversible encryption algorithm, and the generated ciphertext is treated as the "fingerprint" of the message or data block, serving as the unique identifier of the message to verify the content of the information. The one-way variation and fixed length output characteristics of hash functions make them widely used in fields such as data integrity verification, digital signatures, message authentication codes, and data cryptographic protocols.

The structure of the compression function of the SM3 algorithm is similar to that of the SHA-256 algorithm, but the SM3 algorithm incorporates many new design techniques, including adding 16 steps of all XOR operation, message doubleword intervention, and adding fast avalanche effect P-permutation, which can effectively avoid high probability local collisions and resist various cryptographic analyses [12]. Moreover, the SM3 algorithm reasonably utilizes word addition operations to form a carry plus 4-stage pipeline. Without significantly increasing hardware overhead, it adopts P-permutation to accelerate the avalanche effect of the algorithm and improve computational efficiency.The SM3 password hash algorithm adopts basic operations suitable for 32b microprocessors and 8b smart cards, which has high efficiency and wide applicability for cross platform implementation. The comparison between the SM3 algorithm and the SHA series algorithm is shown in the Table 2.3:

The software on ships is organized and operated by a series of files and components. The detection of the status of ship software is actually the detection of the content of the files and components that make up the software [8, 7]. Therefore, in order to implement this system module, the author proposes to algorithmically process all file contents of the target software, convert the file contents into file feature values through a series of operations, and package other attributes of the file together into network packets. The server compares the file attributes and file feature values of the ship software with the software standard library, and realizes the software status detection module by analyzing the matching results.

Table 2.3: Comparison of SM3 and SHA series algorithms.

| algorithm | The length of the output hash value (bits) | Data block length (bits) | Maximum length of input message (bits) | The length of a word (bits) | Number of cycles | The operator used |
|---|---|---|---|---|---|---|
| SHA-0 | 160 | 512 | 264-1 | 32 | 80 | +,and,or, xor,rotl |
| SHA-1 | 160 | 512 | 264-1 | 32 | 80 | +,and,or, xor,rotl |
| SHA-256 | 256 | 512 | 264-1 | 32 | 64 | +,and,or, shr ,xor,rotl |
| SHA-512 | 512 | 1024 | 2128-1 | 64 | 80 | +,and,or, shr,xor ,rotl |
| SM3 | 256 | 512 | 264-1 | 32 | 64 | +,and,or, shr,xor,rotl |

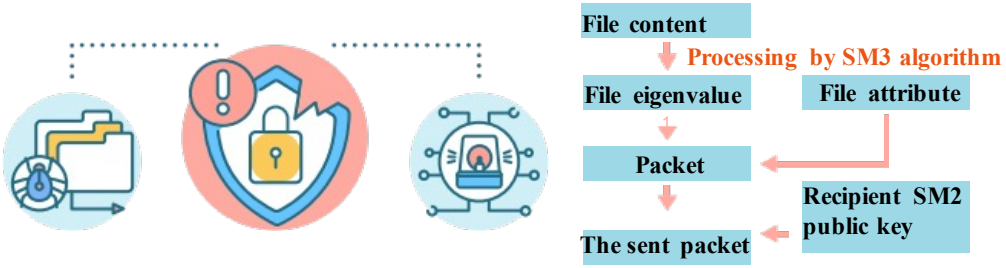


Fig. 2.1: Hybrid algorithm encryption flowchart.

**2.3. Hybrid encryption algorithm technology combining SM3 and SM2.** The author utilizes the advantages of the SM3 algorithm, such as fast computation speed, higher security, lower broadband requirements, and easy key management, in order to combine the two algorithms to obtain a more efficient, accurate, and secure encryption technology [4]. The basic principle of algorithm design is that when the server detects the software status of the ship location, the ship location first uses the SM3 password hash algorithm to process the file content, calculate the file feature values, integrate them with other attributes of the file into a message packet, and then use the SM2 algorithm to encrypt the message packet, the ciphertext containing file information is transmitted from the ship location to the server through a TCP connection. After receiving the ciphertext data sent by the ship location, the server decrypts it using the SM2 decryption algorithm, then compare the obtained file feature values with the feature values in the standard library to determine whether the file's status is legal. This encryption and decryption method not only ensures the security of message data but also improves the speed of message transmission, thereby ensuring the speed, accuracy, and effectiveness of ship software status detection.

(1) Hybrid algorithm encryption process

The encryption process of the hybrid algorithm is to use SM3 to calculate the file feature values based on the file content; The ship station packages the file feature values and other attributes into a message [19]. Encrypt the message using the SM2 public key sent by the server at the ship station; The ship station transmits the encrypted message to the server through a TCP connection. The entire encryption process is shown in Figure 2.1.

(2) Hybrid algorithm decryption process

The decryption process of the hybrid algorithm is as follows: The server decrypts the packet using the SM2 private key to obtain the file attribute information of the ship's location software; Based on the file name, file

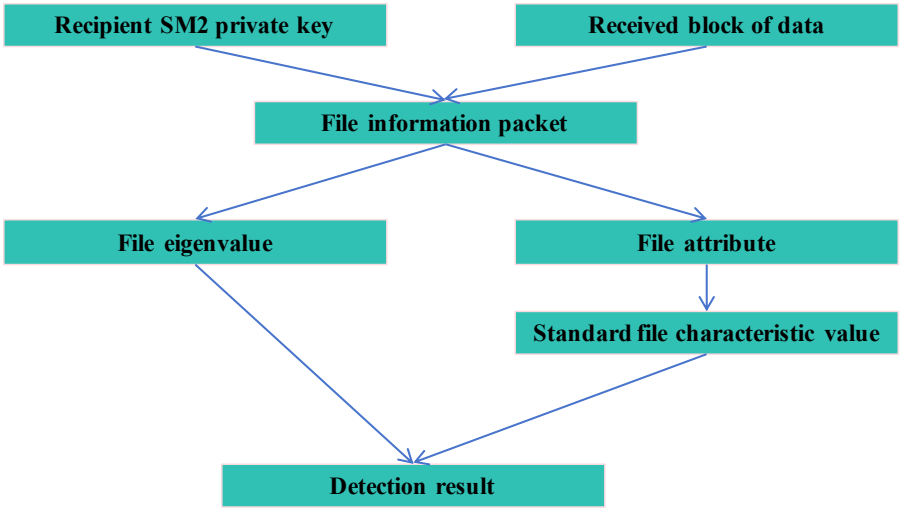


Fig. 2.2: Flow chart of hybrid algorithm decryption.

path, software name, and software version in the file attributes, locate the standard file record corresponding to the file; Compare the feature values of the ship's location file with those in the standard library, determine the status of the file, and display the detection results [16]. The entire decryption process is shown in Figure 2.2.

**3. System Design.**

**3.1. Communication design.** When designing network data communication between the ship's position (client) and the software detection and maintenance system (server), Socket UDP network communication is chosen as the interaction method for the server to issue ship software instructions and report the network status of the ship's position based on the on-site network environment of the system, at the same time, a strategy of triggering high-frequency interaction with a timer is adopted to ensure the speed, real-time performance, and stability of the data; Using SocketTCP as the communication method for software status detection and software entity transmission between servers and ship stations, while adopting flow control and message confirmation strategies to ensure the integrity of data transmission.

**3.2. Architecture design.** The logical architecture of a software inspection and maintenance system based on component scheduling is mainly divided into two levels: Business layer and service layer. The business layer mainly includes visual interfaces for various functions and the basic logic for business implementation, and is the presentation layer for interaction with users. The service layer mainly includes the network data transmission between the server and the ship station, as well as the underlying support of the server's internal database.

**3.3. Deployment design.** The system adopts a CIS structure, with servers deployed in the ship software status command room, and clients distributed on all ship stations that require detection and maintenance [17]. The server has a visual interface and complete functions such as ship and ship software status detection, fault software repair, software version release and update; The client does not have a visual interface and is a console program that starts automatically in the background. It can receive command messages sent by the server and has the function of remote data transmission with the server.

**3.4. Subsystem design.** The software inspection and maintenance system based on component scheduling mainly includes four subsystems: Ship software status detection, software standard library operation and maintenance, software entity control, and software supervision and recording [20]. The main functional objective of the ship software status detection subsystem is to extract the ship software status within the local area network based on the user input of the ship location and relevant information of the ship software. After

comparison with the software standard library, the software status detection results are displayed, mainly including three functional modules: Ship location information control, ship software status detection, and fault analysis of the problematic software. The functional objective of the software standard library operation and maintenance subsystem is mainly to provide a reference template for matching the status of ship software, which is the basis for ship software status detection and fault analysis. It mainly has the function of publishing new software status standards and maintaining software standard status; The main functional objective of the software entity control subsystem is to manage the physical files of the software, mainly including backing up stable and high-performance ship software, repairing ship software with running faults and illegal status, and upgrading and updating outdated ship software. In addition, the system has also designed a subsystem structure for software supervision and recording, with the main functional goal of recording detailed information on the detection results, fault causes, version changes, and other operations of ship software, facilitating users to access and analyze the historical operations of ship software.

## 4. Results and Analysis.

### 4.1. Ship position software detection.

(1) Ship position information control

The main function of ship location information control is to manage and control ship location information. The data items stored in the system for ship location include the name of the ship location, the network P of the ship location, and the list of software controlled by the ship location. The software list information includes the name of the software, the version of the software, and the storage path of the software in the ship location system [1]. The ship position information control module should not only provide a visual interface for ship position information, but also provide functions for adding and modifying ship position information. At the same time, for ship fleets using the same software system, a unified software status control list template should be provided to quickly declare new ship position information through the template.

(2) Ship software status detection

The server issues detection instructions to the ship location software. The ship location first traverses the file directory of the detected software to obtain the file organization structure of the software. The SM3 algorithm is used to sequentially process the file content, calculate the file feature values, and integrate them with other attributes of the file into a message package. Then, the SM2 algorithm is used to encrypt the message package, transfer the ciphertext containing file information from the ship's location to the server through a TCP connection. After receiving the ciphertext data sent by the ship's location, the server decrypts it using the SM2 decryption algorithm, and then compares the obtained file feature values with the feature values in the standard library to finally determine the status of the file and display it.

(3) Problem software fault analysis

Problem software fault analysis mainly focuses on illegal states and ship software that have malfunctioned during operation. Based on the detection of ship software status, fault analysis identifies specific problem sources, identifies files related to the fault and the time of the fault occurrence, enabling users to repair the problem software in a targeted manner, effectively analyzing the cause of the fault, and optimizing the performance of ship software.

**4.2. Software standard library operation and maintenance.** The software standard library operation and maintenance subsystem can release new software status standards, abstract software entities into software feature values, store various data of standard software status in servers, and provide reference basis for software status detection of ship stations [3, 13]. This module can adjust the details of the released software standard status, present the entire software standard status to users in a good interface interaction, facilitate users to find the source of problems with faulty software, and analyze the functional differences between different software versions.

**4.3. Software entity control.** The software entity control subsystem mainly provides functional interfaces for software entity operations, including ship software backup, ship software fault repair, and ship software version upgrade [14]. In order to improve the unity and correlation between software entities and software states, the operation of software entities is closely tied to ship software state detection and software standard library operation and maintenance, this means that each control of software entities corresponds to a

Table 4.1: Detection results of ship software based on hybrid algorithm.

| File name | File type | File size | Version Status |
|---|---|---|---|
| 1029LBandGlobePara_ sys.xml | xml | 2KB | × |
| 1029LHFGlobePara_ sys.xml | xml | 3KB | × |
| 1029LHFGlobePara _work.xml | xml | 3KB | × |
| Config.xml | xml | 2KB | × |
| Config.xml | xml | 2KB | × |
| Config.xml | xml | 2KB | × |
| DDSRecvLog.txt | txt | :1KB | × |
| config.ini | ini | 1KB | × |
| 1.500000MHz- CW- 2016 08- 10 | Wav | 1KB | ✓ |
| 1.txt | txt | 26KB | ✓ |
| 1029HFAutoSearchPara.xml | xml | 41KB | ✓ |
| 1029HFAutoSearchPara_ LF.xml | xml | 41KB | ✓ |
| 1029HFDFTestPara.xml | xml | 5KB | ✓ |
| 1029HFDemoduCtrlPara_ 1.xml | xml | 11KB | ✓ |
| 1029HFDemoduCtriPara_ 2.xml | xml | 11KB | ✓ |
| 1029HFDemoduCtrlPara _3.xml | xml | 11KB | ✓ |
| 1029HFDemoduCtriPara_ 4.xml | xml | 11KB | ✓ |
| 1029HFDemoduCtrlPara_ 5.xml | xml | 11KB | ✓ |
| 1 029HFDemoduCtrlPara 6.xml | xml | 11KB | ✓ |
| 1029HFDemoduCtrlPara _7.xml | xml | 41KB | ✓ |

synchronous change in the status of ship software, effectively avoiding the phenomenon of ship software being tampered with and users' uncontrolled management of ship software entities. Moreover, this system provides a unified interface for software entity operations, eliminating the need for users to perform software entity operations on ship platforms. This simplifies the workflow for repairing and upgrading ship software faults, effectively improves the efficiency of software entity control, greatly reduces software maintenance costs, and has constructive significance for the standardization and standardization of software entity operations.

**4.4. Software control records.** The software control record subsystem provides a visual interface for viewing all software control records of users, mainly recording controlled ship information, such as ship location network IP, ship software name and version, operation time, specific operation behavior, and related file statistics, it mainly records the software detection status of ship positions and changes in software versions. The system will record detailed information on the status of illegal software discovered during the detection process and display it to users in a visual list. Users can query specific problem sources by viewing illegal file records, strengthen the management of ship software, and ensure the stability and uniformity of software operation.

**4.5. Test results.** The author conducted practical verification of the SM3 and SM2 hybrid encryption algorithm and selected software on the ship for detection [6]. After analyzing the experimental results, it has been proven that using a hybrid algorithm for encryption and decryption, the server can accurately obtain software information on the ship's platform, detect the software status on the ship, and locate specific problem files. For software that does not meet the standard status, the server can accurately transmit software information to the "component integration framework" and put the component in a "prohibited" scheduling state. After the server repairs the problematic software, the detection results of the software change and display as legal, while the software is in the "allowed" scheduling state in the "component integration framework". The detection results of ship software status are shown in Table 4.1:

**5. Conclusion.** In the software detection and maintenance system designed by the author, ship software status detection is the basic function of the system for subsequent management and maintenance of ship software. In order to ensure the accuracy and uniqueness of the target software detection results, this system traverses the file directory of the target software and uses the SM3 password hash algorithm to calculate the file feature

values. If there are any changes to the file content, the calculated feature values will definitely not be the same, so by matching the feature values in the software standard library, it is possible to determine whether the content of the file has been tampered with, thereby obtaining the detection results of the ship software. And in response to the disadvantage of low security of a single encryption algorithm, this system adopts a mixed encryption scheme of SM3 and SM2 to encrypt the transmission of network messages, ensuring sufficient security of the software detection process and accurate detection results. We have designed a software inspection and maintenance system based on component scheduling, mainly including communication design, deployment design, architecture design, etc, chosen the Qt compiler as the research and development environment for the system, and SQLite as the storage medium for storing software feature values on the server. We have designed a network message structure for data exchange, and implemented a software detection and maintenance system that includes the above solutions. This not only effectively improves the detection efficiency of ship software status, but also ensures the safety of the detection process and the accuracy of the detection results, moreover, it can significantly shorten the repair time of faulty software, save the cost of ship software maintenance, strengthen centralized management of software release and updates, and improve the control ability of ship software.

## REFERENCES


[1] K. Chen, L. Hu, M. Yao, L. Qian, and Y. Zhang, *Study on intelligent traffic search method based on driver facial feature analysis*, International Journal of Vehicle Information and Communication Systems, 6 (2021), pp. 151–160.

[2] L. Duan, J. Zhou, Y. Wu, and W. Xu, *A novel and highly efficient botnet detection algorithm based on network traffic analysis of smart systems*, International Journal of Distributed Sensor Networks, 18 (2022), pp. 182459–182476.

[3] M. Ghahramani, A. O'Hagan, M. Zhou, and J. Sweeney, *Intelligent geodemographic clustering based on neural network and particle swarm optimization*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52 (2021), pp. 3746–3756.

[4] X. He, Y. Yang, W. Zhou, W. Wang, P. Liu, and Y. Zhang, *Fingerprinting mainstream iot platforms using traffic analysis*, IEEE Internet of Things Journal, 9 (2021), pp. 2083–2093.

[5] J. Jagannathan and M. M. Parvees, *Vulnerability recognition and resurgence in network based on prediction model and cognitive based elucidation*, Journal of Physics: Conference Series, 2070 (2021), p. 012122.

[6] K. Li, Y. Wang, Y. Shao, and X. Wu, *Smart boat detection based on feature pyramid network and deformable convolution*, Journal of Physics: Conference Series, 2083 (2021), p. 042018.

[7] B. Liu, Y. Fan, B. Xue, T. Wang, and Q. Chao, *Feature extraction and classification of climate change risks: a bibliometric analysis*, Environmental Monitoring and Assessment, 194 (2022), pp. 1–41.

[8] Q. Majeed and A. Fathi, *A novel method to enhance color spatial feature extraction using evolutionary time-frequency decomposition for presentation-attack detection*, Journal of Ambient Intelligence and Humanized Computing, 14 (2023), pp. 3853–3865.

[9] H. Shen, P. Fan, Z. Wei, C. Zhao, S. Zhou, and Q. Wu, *Research on transmission equipment defect detection based on edge intelligent analysis*, Journal of Physics: Conference Series, 1828 (2021), p. 012087.

[10] Z. Shi, A. A. Mamun, C. Kan, W. Tian, and C. Liu, *An lstm-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing*, Journal of Intelligent Manufacturing, 34 (2022), pp. 1815–1831.

[11] Z. Song, X. Huang, C. Ji, and Y. Zhang, *Intelligent identification method of hydrophobic grade of composite insulator based on efficient ga-yolo former network*, IEEJ Transactions on Electrical and Electronic Engineering, 18 (2023), pp. 1160–1175.

[12] G. Tang, L. Yang, L. Zhang, W. Cao, L. Meng, H. He, H. Kuang, F. Yang, and H. Wang, *An attention-based automatic vulnerability detection approach with ggnn*, International Journal of Machine Learning and Cybernetics, 14 (2023), pp. 3113–3127.

[13] Z. Tong, C. Xing, Z. Zeng, and B. Lai, *Intelligent tidal lane system based on vehicle attribute recognition algorithm and related laws and regulations*, Journal of Physics: Conference Series, 1972 (2021), p. 012100.

[14] B. Wang, J. Li, J. Luo, Y. Wang, and J. Geng, *Intelligent deblending of seismic data based on u-net and transfer learning*, IEEE Transactions on Geoscience and Remote Sensing, 59 (2021), pp. 8885–8894.

[15] J. Xie, Y. Zhao, D. Zhu, J. Yan, J. Li, M. Qiao, G. He, and S. Deng, *A machine learning-combined flexible sensor for tactile detection and voice recognition*, ACS Applied Materials & Interfaces, 15 (2023), pp. 12551–12559.

[16] H. Xu, M. Guo, N. Nedjah, J. Zhang, and P. Li, *Vehicle and pedestrian detection algorithm based on lightweight yolov3-promote and semi-precision acceleration*, IEEE Transactions on Intelligent Transportation Systems, 23 (2022), pp. 19760–19771.

[17] H. Yilahun and A. Hamdulla, *Entity extraction based on the combination of information entropy and tf-idf*, International Journal of Reasoning-based Intelligent Systems, 15 (2023), pp. 71–78.

[18] L. Yu, Y. Lu, Y. Shen, H. Huang, and K. Zhu, *Bedetector: A two-channel encoding method to detect vulnerabilities based on binary similarity*, IEEE Access, 9 (2021), pp. 51631–51645.

[19] B. Zhang and Z. Xi, *A systematic review of binary program vulnerabilities feature extraction and discovery strategy generation*

*methods*, Journal of Physics: Conference Series, 1827 (2021), p. 012090.
[20] X. Zou, R. Lv, X. Li, H. Chen, Z. Wang, and H. Yang, *Intelligent electrical fault detection and recognition based on gray wolf optimization and support vector machine*, Journal of Physics: Conference Series, 2181 (2022), p. 012058.

*Edited by:* B. Nagaraj M.E.
*Special issue on:* Deep Learning-Based Advanced Research Trends in Scalable Computing
*Received:* Aug 31, 2023
*Accepted:* Nov 4, 2023