Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2023 SCPE. Volume 24, Issues 4, pp. 699–712, DOI 10.12694/scpe.v24i4.2151

COMPUTER HARDWARE FAULT DETECTION BASED ON MACHINE LEARNING

CHUNXUE XU[*]

Abstract. In order to solve the computer fault detection problem of machine learning, the author proposes a computer hardware fault detection problem based on machine learning. The method combines mutual information and class separability to analyze their relationship, which improves classification accuracy. This study presents an adaptive machine learning technique for the adaptive fusion of data from multiple sources. In addition, the mCRC algorithm seeks for the optimal feature subset using the enhanced forward floating search method, thereby overcoming the limitation that the mRMR algorithm does not specify how to determine the final feature subset. The classification accuracy of the mCRC algorithm is approximately 1% better than that of the mRMR algorithm, and the size of the final feature subset of the mCRC algorithm is 22% smaller than that of the final subset of the mRMR algorithm. Conclusion: the ReMAE algorithm has a higher rate of accurate failure prediction.

Key words: Fault detection; Machine learning; Fault characteristics; Active fault tolerance.

1. Introduction.

The increasing application demand promotes the rapid development of high-performance computers, with the increasing scale of the system, the number of high-performance computer components increases rapidly, the mean time between failures of the system is getting shorter and shorter, and the reliability problem is becoming increasingly prominent. The original passive fault-tolerance method of high-performance computers based on Checkpoint can no longer meet its reliability requirements, active fault-tolerance based on fault prediction is an important fault-tolerance strategy to improve the reliability of high-performance computers in the future. The existing high-performance computer fault prediction technology is basically an offline batch learning method, with low prediction accuracy and poor dynamic performance, which cannot meet the application requirements of future high-performance computers, therefore, there is an urgent need for an efficient online fault prediction method that can learn fault data online, accurately predict impending failures in real time, enabling low-overhead proactive fault tolerance before failures occur, increasing system availability.

In industrial manufacturing, rotary equipment is routinely employed, but repeated exposure to heavy loads can cause critical components to degrade and fail. Given the interactions between the parts, if a degenerating part is not discovered in a timely manner, the manufacturing process could be delayed or suffer catastrophic damage. It is vital to monitor and troubleshoot a plant's essential components to ensure its stable operation and production safety. In an Endeavour to considerably increase profitability, more emphasis has been placed on defect detection in recent years as a result of reliable diagnostic techniques. As a consequence of industrial manufacturing's automation and intelligence, it is simpler to collect large quantities of data. The development of graphics processing units (GPUs), for example, has enhanced hardware, allowing for the analysis and diagnosis of large amounts of data. Deep learning, which evolved from traditional shallow machine learning, can better analyze prospective features.

Deep learning is presently employed extensively in a variety of domains, including image recognition, intelligent robotics, and audio recognition, among others. The three primary phases of an intelligent diagnostic system are feature extraction, defect recognition, and data preprocessing. Some early superficial machine learning techniques, such as the artificial neural network (ANN), support vector machines (SVMs), Bayesian networks, and the convolution neural network (CNN), required data preprocessing based on the expertise of humans in order to extract the data's features. Computer hardware is an essential part of contemporary technology and is susceptible to a variety of malfunctions that can result in serious issues including system outages, data loss, and decreased performance. Particularly for large-scale systems, locating and diagnosing these errors may be a laborious procedure that takes a lot of time. Therefore, by examining system performance data,

[*] Baicheng Normal University, College of Computer Science, Baicheng, Jilin 137000, China (chunxuexu45@gmail.com).

machine learning (ML) methods are increasingly being utilized to automate the diagnosis of hardware defects. The goal of this study is to investigate how machine learning might enhance the precision, effectiveness, and speed of computer hardware problem detection. We hope that our research will aid in the development of fault detection systems that are more accurate and dependable and that can help identify and diagnose hardware issues before they result in significant harm.

The remaining article is structures as: Literature review presentation in section 2 of the article followed by methodology explained in section 3. Section 4 presents the examination results followed by conclusion section in section 5.

2. Literature Review. Marty et al. said that the development of information technology promotes the continuous progress of society, the arrival of the era of big data makes international competition more and more fierce, and all countries in the world have invested in the development of high-performance computers [1]. Deng et al. said that the U.S. Defense Advanced Research Projects Agency launched the UHPC (Ubiquitous High-Performance Computing) program four, researching revolutionary design methods to meet the growing demand for high-end computing in defense applications. The EU-funded DEISA (Distributed European Infrastructure for Supercomputing Applications) project uses large-scale high-performance computing systems to promote scientific and technological progress [2]. Zhang et al. said that China also attaches great importance to the research of high-performance computer technology, and has developed supercomputers with international influence such as Tianhe, Yinhe, Dawning, and Shenwei [3]. Sengupta et al. said that in 2016, China launched the “E-class Supercomputer Prototype System Development” project, and plans to start the development of E-class supercomputer systems in 2018, which indicates that the development of high-performance computing systems in China has entered a new journey [4]. Li, et al. said that the continuous growth of application requirements has led to the rapid development of high-performance computers, with the increasing scale of the system and the rapid increase in the number of high-performance computer components, the mean time between failures of the system has dropped from days to hours [5]. Aggarwal et al. said that the current mainstream system-level Checkpoint technology is a typical passive fault-tolerant method, and it is also the main means of fault-tolerance for high-performance computer systems [6]. Wu, et al. said that the tests of the “Tianhe-1” system showed that, when the scale of parallel jobs reaches more than 4096 nodes, the time required for a checkpoint is more than ten minutes, this passive fault-tolerant method has seriously restricted the continuous computing efficiency of the “Tianhe No. 1” system, and because the system-level Checkpoint overhead is too large, it has seriously affected the availability of the system [7].

Computer hardware fault detection based on machine learning is shown in Figure 2.1. Liu, et al. said that the scale of the system continues to grow and the complexity of hardware and software continues to increase, these make the mean time between failures of supercomputers getting shorter and shorter [8]. Yuan, et al. said that when the scale of the supercomputer system reaches a certain scale, continuing to expand the scale of the system will not only fail to shorten the running time of the job, on the contrary, the execution time of the job will become longer and longer due to the constraints of fault tolerance overhead, that is, reliability, it restricts the expansion of the system scale [9]. Cotroneo et al. said that the future supercomputer system is composed of more than tens of thousands of nodes, and the scale is much larger than the current system, passive fault-tolerant methods such as Checkpoint will not be able to meet its application requirements, new methods are urgently needed to improve the scalability of high-performance computer systems, so active fault-tolerance technology based on fault prediction has become a new research hotspot [10].

Table 2.1 lists recent research on machine learning-based defect detection for computer hardware. The research use numerous datasets, such as sensor data from industrial control systems, hydraulic systems, and aerospace components, and use a range of machine learning approaches, including Random Forest, SVM, Decision Trees, and LSTM. These studies advantages include precise defect identification, early forecasting, and cost savings from maintenance, while their drawbacks include small datasets, the requirement for feature engineering, and expensive processing costs. Ensemble approaches, data augmentation, feature engineering, and the use of external data sources are some of the suggested fixes.

In order to promote generalizability and adaptability to other computer hardware systems, future work will primarily focus on expanding testing on new datasets, including more thorough feature engineering and ensemble approaches, and increasing real-world testing.

Table 2.1: Recent work done for the fault detection

Reference	Technology Used	Datasets	Benefits	Drawbacks	Solutions	Future Work
[11]	SVM, Random Forest, Decision Trees, MLP	PHM08 Challenge Dataset, IMS dataset	Accurate failure prediction, reduced maintenance costs	Limited dataset, need for feature engineering	Feature engineering, ensemble models	More extensive testing on additional datasets
[12]	kNN, Naive Bayes, SVM, Decision Trees, Random Forest	In-house server logs	High detection accuracy, early failure prediction	Limited dataset, may not generalize to other servers	Incorporating external data sources, data augmentation	Testing on additional server data
[13]	Bayesian Surrogate-Assisted Optimization, Random Forest, XGBoost	SPEC CPU 2017 benchmarks	Increased performance, reduced hardware failures	High computational cost	Parallel execution, reduced surrogate evaluation	Testing on additional hardware configurations
[14]	SVM, Random Forest, CNN	Diesel engine sensor data	Accurate fault diagnosis, early detection	Insufficient data for some fault types	Ensemble methods for multi-class classification	Incorporation of additional sensor data
[15]	LSTM, Decision Trees, Random Forest	Simulated multi-agent system data	Accurate fault detection, distributed monitoring	Limited dataset, may not generalize to other systems	Feature expansion, additional real-world testing	Testing on other multi-agent systems
[16]	LSTM, Random Forest	In-house edge servers	Early fault detection, reduced maintenance costs	Limited dataset, imbalance of normal/faulty system statuses	Incorporating external data sources, data augmentation	Testing on additional edge server data
[17]	SVM, Random Forest, CNN	Aerospace component sensor data	Early fault prediction and identification, reduced maintenance costs	Limited dataset, insufficient sensor data	Feature engineering, data augmentation	Testing on additional aerospace components
[18]	SVM, Random Forest, Decision Trees	NIST Cybersecurity Dataset	Accurate failure detection, improved cybersecurity	Limited dataset, need for additional features	Incorporation of additional data sources, feature engineering	Testing on additional industrial systems
[19]	kNN, SVM, Decision Trees, Naive Bayes	Hydraulic systems sensor data	Accurate fault detection, reduced maintenance costs	Insufficient data on rare faults	Ensemble methods, feature engineering	Testing on additional hydraulic systems data
[20]	kNN, SVM, Decision Trees, Naive Bayes, Random Forest, ANN	Various datasets	Early fault detection, reduced maintenance costs	Limited datasets, need for feature engineering	Feature engineering, ensemble techniques	Testing on additional datasets

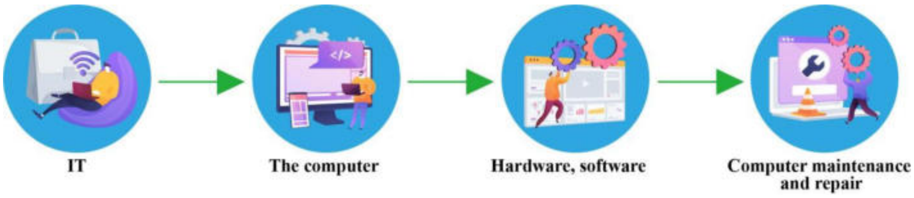

Fig. 2.1: Machine learning for computer hardware failure detection [11, 12]

3. Proposed Methodology. When using machine learning to forecast high-performance computer failure, it is crucial to collect node state data linked to machine failure while the computer is running since these state data directly affect the outcome of failure prediction [21, 22]. The study team suggests a distributed state data collecting system called FPDC, where each node gathers its own operational state data, in an effort to address the issue that there isn't enough state data in the present high-performance computer failure prediction research [23]. The acquisition of state data must satisfy the needs of various types of fault prediction because only a very small number of nodes in the system are in an imminent state of failure, making the state of these nodes prior to the failure important for failure prediction [24, 25]. Moreover, since different types of failures require different state data, the acquisition of state data must be tailored to each type of failure. In order to accomplish this, when collecting the status information of the ”Tianhe No. 1” node using the FPDC framework, the collection task is distributed to each computing node using distributed technology, and the node that is assigned the collection task runs the lightweight data collection process. The node state data is subsequently collected on a regular basis [26].

The FPDC data acquisition framework has two functions: on the one hand, in the fault learning stage, the data collected in a fixed period of time is collected to form a training set, and a classifier for subsequent fault prediction is obtained by training, and distribute these classifiers to each computing node. On the other hand, in the fault prediction stage, each computing node collects its own runtime status data through the FPDC framework, and classifies and predicts these data through the classifier deployed on the node, once it is predicted that there will be faulty data before the node stops running, active fault tolerance is activated to repair the faulty node. Before the fault prediction, it is not known which software and hardware attribute data are related to the fault prediction. In order to make the prediction effect better, when the data collection of the training set is carried out in the early stage, as much attribute data as possible will be collected, in order to provide sufficient and effective training data to learn the classifier. To avoid the negative effects on system performance brought on by the storage and transmission of a significant amount of useless data, the original data set is processed through the state data preprocessing technology prior to failure prediction, and only the useful data is transmitted to the service node for online learning. However, some of these collected state attribute data may be irrelevant to fault prediction, or even hinder fault prediction. Table 2 displays information about the hardware environment state [27]. The proposed methodology for the fault detection using Machine Learning is depicted in Figure 3.1.

i. Data Collection: Obtain data pertinent to the computer hardware system, including sensor readings, log files, and performance metrics.

ii. Data Preprocessing and Feature Engineering: Cleaning and preprocessing the collected data may involve handling missing values, detecting outliers, and normalizing the data. The purpose of feature engineering is to extract meaningful features or transform the data in order to enhance the efficacy of machine learning models.

iii. Model Selection: Based on the nature of the data and the problem at hand, select appropriate machine learning algorithms, such as Random Forest, Support Vector Machines, and Artificial Neural Networks.

iv. Model Training and Evaluation: Train the chosen model with the preprocessed data and evaluate its performance using the appropriate metrics, such as accuracy, precision, recall, or F1 score. Cross-validation techniques may also be used to assess the generalizability of the model.

v. Model Optimization: Using techniques such as grid search or random search, fine-tune the model's hyper

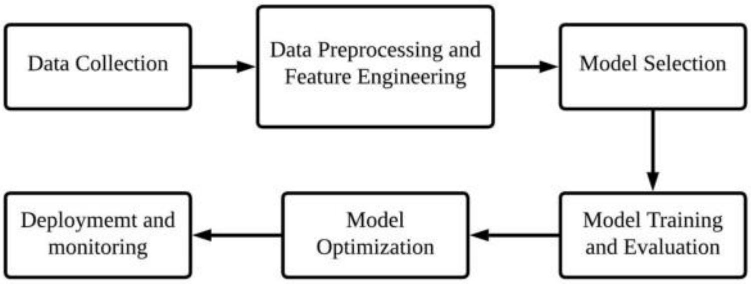
Data Collection
Data Preprocessing and Feature Engineering
Model Selection
Model Training and Evaluation
Model Optimization
Deploymemt and monitoring

Fig. 3.1: Steps involved in proposed model for Fault Detection

Table 3.1: “Tianhe No. 1” node hardware environment status data

Source	Attribute name(unit1)	Attribute name(unit2)	Attribute name(unit3)	Status category	Total
Compute node	Vbat(V)	12V(V)	5Vsb(V)	Voltage	12
Compute node	5V(V)	3.3V(V)	3.3Vsb(V)	Voltage	12
Compute node	ICH-1.5V(V)	IOH-I.IV(V)	CPUI	Voltage	12
Compute node	CPU0 core(V)	CPUO	DDR3-1.5V(V)	Voltage	12
Compute node	Thrm(C)	DR3-1.5V(V)	CPUI core(V)	temperature	3
PDP	PDP-3.3V(V)	CPUO Temp(C)	CPUI Temp(C)	Voltage	5
PDP	PDP-1.5V(V)	PDP-2.5V(V)	CPUI Temp(C)	Voltage	5

parameters to enhance its efficacy further.

vi. Deployment and Monitoring: Deploy the trained model in the production environment and monitor its performance continuously to detect any changes or anomalies.

The adopted methodology is unique and superior to current methods and differs from study to study based on the specific methodology employed. Among the possible benefits of using machine learning for computer hardware fault detection are the following:

i. Automation: Machine learning models can automatically analyze and detect patterns and anomalies that may be missed by manual inspection or conventional rule-based methods.

ii. Adaptability: Machine learning models can adapt to and learn from new data, allowing them to detect previously unknown or emergent hardware faults.

iii. Accuracy: By leveraging complex algorithms and learning from historical data, machine learning models can identify and predict hardware failures with high accuracy.

iv. Early Detection: Machine learning models can frequently detect faults early, allowing for opportune maintenance or intervention and preventing more severe system failures.

v. Scalability: Once trained, machine learning models can be deployed and used at scale, making them appropriate for large-scale hardware systems and cloud environments.

vii. It is essential to note, however, that the superiority of the adopted methodology over existing methods may depend on a number of factors, including the quality and representativeness of the data, the selection and tuning of machine learning algorithms, and the efficiency of feature engineering and preprocessing techniques. The superiority of the proposed model must be demonstrated through comparisons with existing methods and exhaustive evaluations.

At the same time, the collected data contains 46 feature attributes, if the magnitude of the different feature attributes is too different, the change of the large number will mask the change of the small number, which will have a great impact on the subsequent classification accuracy, therefore, before using the training data set, it is necessary to unify the attribute features of different orders of magnitude in the same dimension, that is, data normalization, in order to remove the influence of the data dimension. Commonly used data normalization

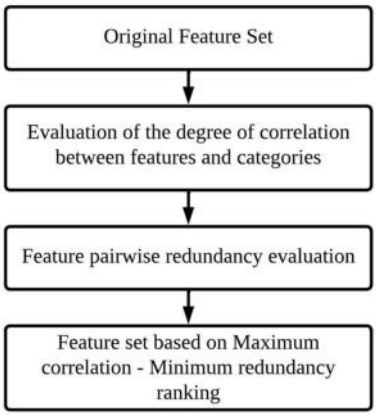
Original Feature Set
Evaluation of the degree of correlation between features and categories
Feature pairwise redundancy evaluation
Feature set based on Maximum correlation - Minimum redundancy ranking

Fig. 3.2: mRMR algorithm framework

methods include linear function transformation, logarithmic function transformation and inverse cotangent function transformation, this topic uses the linear function transformation method to normalize the "Tianhe No. 1" node state data set, and normalizes each column of feature data to the same dimension as shown in formula (3.1):

$$value* = (value - min)/(max - min) \tag{3.1}$$

where value and value* are the values before and after the conversion, respectively, and max and min are the maximum and minimum values of the feature data of the column, respectively. In the mRMR feature selection algorithm, the correlation and redundancy measures are executed in two stages, and both are calculated using mutual information. Figure 3.2 illustrates the framework of the mRMR algorithm.

Although the values of correlation and redundancy are derived via the calculation of mutual information, it is challenging to estimate the maximum correlation between features and categories and the minimal redundancy of feature subset combinations. The mRMR algorithm calculates the maximal correlation using the following formula (3.2):

$$\max De(F, C), De = 1/|F|^2 \sum_{f,eF} I(C, f_i) \tag{3.2}$$

According to formula (3.2), there may be multiple features and categories with the same similarity value, so the features selected by formula (3.2) will have high redundancy. The minimum redundancy constraint is added on the basis of the maximum correlation and a special subset that is strongly correlated with the category and mutually exclusive can be obtained. In the mRMR algorithm, the calculation formula of the minimum redundancy is proposed as shown in formula (3.3):

$$\max De(F, C), De = 1/|F|^2 \sum_{f,fieF} I(f_i, f_j) \tag{3.3}$$

Combining formula (3.2) and formula (3.3) in a certain way is the criterion of maximum correlation and minimum redundancy. mRMR defines the combined formula as shown in formula (3.4):

$$\max J(f), J = De - R \tag{3.4}$$

or as shown in formula (3.5):

$$\max J(f), J = De/R \tag{3.5}$$

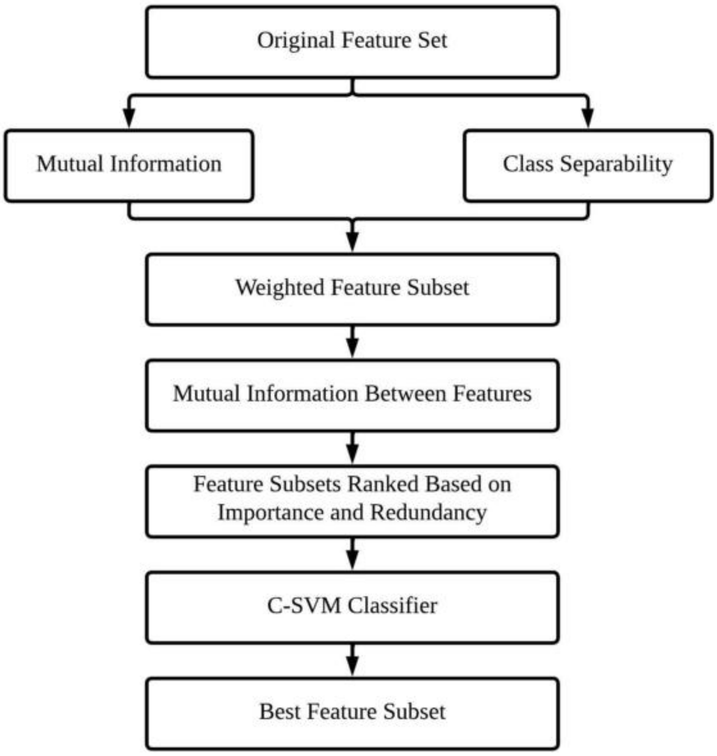

Fig. 3.3: mCRC algorithm framework

The mRMR algorithm evaluates all features in the original feature set based on the maximum correlation-minimum redundancy criterion of Equation (3.4) or Equation (3.5), the resulting feature set is an ordered set-in descending order of importance. On the basis of the maximum correlation minimum redundancy feature selection algorithm mRMR, the author proposes the Filter and Wrapper combined feature selection algorithm mCRC (Feature Selection Algorithm Based on Multi-Criteria Ranking and SVM). On the premise of the mRMR algorithm, this algorithm adds category separability measurement criteria to sort features, as feature subset evaluation via category separability measurement can effectively enhance the ability of feature selection on small samples and linearly inseparable data sets. In addition, the mCRC algorithm incorporates SVM to perform an enhanced sequential forward floating search on the feature sets ranked in descending order of importance in order to obtain the final feature subset. Figure 3.3 depicts the architecture of the mCRC algorithm.

In the Filter section, the mCRC algorithm considers the advantages of two criteria at the same time, that is, through the mutual information and category separability criteria, and calculates the correlation between each information in the original feature and the category at the same time. The value of the mean $W(f_i), W(f_i) 1 < i < n$ of each feature correlation (weight) is obtained as shown in formula (3.6):

$$W(f_i) = 1/2(I(C, f_i) + W(f_i)) \tag{3.6}$$

When a feature attribute is completely irrelevant to the sample category, the calculated mutual information value between the feature attribute and the sample category is zero. Therefore, in the process of calculating the mutual information between feature attributes and categories, the features with a value of zero, that is, irrelevant features, can be deleted, and the feature subset U with completely irrelevant features and mean weights can be obtained. At the same time, the author sets a set Q that is initially empty, selects one of all features in U to join Q at a time, and deletes the feature from U at the same time until U is empty. The first feature selected from U is the feature with the largest $W(f_i)$ value, and the first one is selected to be added to

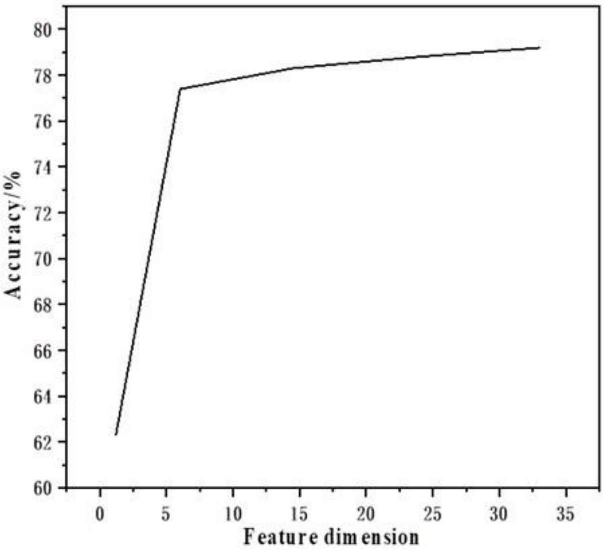

Fig. 3.4: Comparison of feature sorting performance between mRMR algorithm and mCRC algorithm

Q. As shown in formula (3.7):

$$g_i = \arg\max(f_i) \tag{3.7}$$

where $g_j \in Q, 1 \leq j \leq n$, then for the feature $f_i, 1 \leq i \leq n$ existing in U, take any g_j, if $H(f_i) = H(f_j) = H(f_i, f_j)$, then f_i and g_j are completely redundant, the feature f_i can be deleted from the set U, otherwise, for each feature in the set U. The maximum mutual information $I_{\max}(f_i, g_j)$ between the feature and all the features in the set Q is taken to represent the redundancy between the feature and the set Q, the mCRC algorithm evaluates the importance of feature attributes through the maximum correlation and minimum redundancy criteria. It selects the most important feature in the set U and puts it into the set Q, the evaluation formula for the importance of the features in the set U is shown in formula (3.8):

$$J(f_i) = W(f_i)/I_{\max}(f_i, g_j) \tag{3.8}$$

The set becomes the lth feature of Q (l is the size of set Q) as shown in formula (3.9):

$$g_j = \arg\max(J(f_i)) \tag{3.9}$$

When the set U is empty, all the features in the set Q are the feature subsets sorted in descending order of importance. Compared with the mRMR algorithm, the filtering and sorting part of the multi-criteria weighted sorting algorithm mCRC combines the comprehensive advantages of the two feature evaluation criteria, to a certain extent, it overcomes the one-sidedness of the mRMR algorithm that uses mutual information alone for evaluation, and makes the ranking result more accurate. In order to make the experimental results more convincing, five-fold cross-validation is used for experimental verification. The experiment uses LibSVM toolbox, the kernel function of SVM adopts Gaussian kernel, and the optimal penalty factor L and kernel function parameter are selected by grid search to avoid the blindness of parameter selection, as shown in formula (3.10):

$$K(x, x') = exp(-||x - x'||^2)LT, \quad L \text{ belongs to } \{2^-5, \ldots, 2^5\}, \quad T \text{ belongs to } \{2^-5, \ldots, 2^5\} \tag{3.10}$$

Using the filtering and sorting part of the mRMR and mCRC algorithms to sort the features of the dataset respectively, and using the SVM classifier to remove all the features that are completely redundant and completely irrelevant, and are sorted in descending order of importance, the experiment is divided into two parts.

Figure 3.4 depicts the experimental comparison outcomes of the feature ranking of the two algorithms. Figure 3.4 demonstrates that, for the ”Tianhe No. 1” node status data set, both mRMR and mCRC feature selection algorithms demonstrate that, as the number of features increases, the classification accuracy initially increases, but then remains relatively stable or decreases, indicating that a large proportion of the original feature set's feature attributes are redundant or even have side effects on classification; In addition, the figure reveals that there are fewer than 46 features in the feature set acquired after feature sorting, indicating that some features from the original set that were extraneous to the sample category have been eliminated. After sorting the original feature set in descending order of importance using the filtering and sorting portions of the mRMR and mCRC algorithms, only a small number of feature combinations can be used to achieve the same or higher classification accuracy than the original set. Both mRMR and mCRC algorithms can identify key features, as demonstrated.

At the same time, it can be seen that, on the “Tianhe No. 1” node state data set, the mCRC algorithm can achieve or even exceed the classification accuracy of the initial data set with a smaller number of features than the mRMR algorithm. However, it can also be seen from Figure 3.3 that the classification accuracy of the two feature selection algorithms does not smoothly increase to a certain value with the increase of the number of feature attributes and then stabilize or decrease, in the process of increasing the accuracy rate, adding individual features to the subset combination will cause the performance of the algorithm to suddenly decline and continue to rise, indicating that there are some errors in the sorting process, therefore, the author does not use the sequential forward search method to obtain the final feature subset, but obtains the final feature subset by using the sequential floating forward search method that sets an error tolerance threshold , therefore, when the sequential forward search method is used to search, the classification accuracy is in the process of increasing, and the individual features are added to reduce the situation that the optimal performance cannot be achieved. The value should not be set too large, an excessively large value is equivalent to performing a sequential floating forward search on the entire sorted feature set, which leads to lower operating efficiency and loses the meaning of feature sorting.

4. Experiments and Analysis. Data flow classification is basically a type of online classification technique; with the ongoing creation of data, this online learning technique uses the freshly generated data stream to continue training and updating the classifier created through the prior training. The ensemble data flow classification separates the sequentially incoming data flow into blocks, and the most recent data block is used as the training set to teach the new classifier after the test set is used to evaluate the performance of the existing classifier in making predictions [28]. The integrated data stream mining algorithms SEA, AWE, ACE, and others are often employed. Although these integrated data stream mining methods have some adaptive effects on idea drift, they nevertheless have the following flaws: the trained classifiers' previous roles are not taken into account when utilizing the most recent data blocks to assess them.

At present, when the integrated data stream mining algorithm evaluates the existing classifiers and determines the individuals to be deleted this time, only the prediction ability of each basic classifier individual for the latest data block is considered, this way of evaluation ignores the historical role of the base classifier. Implicit mutation concept drift in streaming data often causes better-performing base classifiers to be eliminated, a small concept drift is likely to lead to poor prediction results for the current data by the previously important base classifiers, as a result, it is deleted, for the ever-changing data flow, the currently saved base classifier may only be the one with the best current performance, but not the one with better performance globally, which makes the final prediction effect unsatisfactory. Although the categories in the data stream are frequently unbalanced, existing algorithms typically presume that they are: Since the majority of practical applications suffer from the category imbalance problem, it is likely that the current data block only contains one or a few categories of data. As a result, the currently trained classifier will struggle to recognize the categories of data that have not yet been learned, which will negatively affect algorithm performance [29, 30]. The MEA algorithm offers a good solution to the issue of ignoring the historical significance of the base classifier. This algorithm incorporates human ”recall and forgetting” mechanisms into data stream mining, which not only keeps the historical classification from being eliminated in the case of sudden conceptual drift a well-performing base classifier, but also allows it to integrate the best-performing base classifier to predict samples, improving the stability and accuracy. The MAE algorithm introduces the “recall and forgetting” mechanism into the

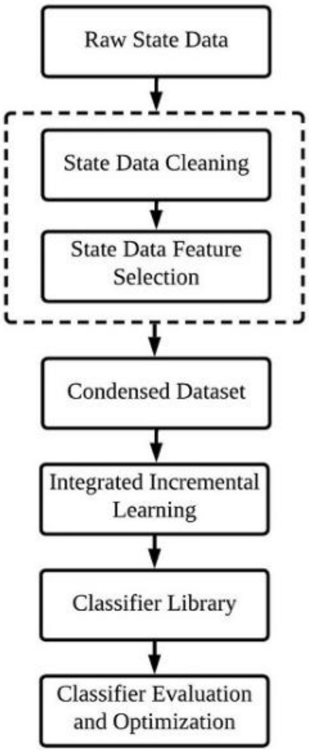
Raw State Data
State Data Cleaning
State Data Feature Selection
Condensed Dataset
Integrated Incremental Learning
Classifier Library
Classifier Evaluation and Optimization

Fig. 4.1: MAE algorithm framework

learning process of the base classifier. The algorithm model presets two classifier banks: The memory bank MS, which is used to save all the current useful base classifiers. The recall library ES, used to save the currently recalled classifier, ES is a subset of MS [31].

The MAE algorithm compares each base classifier obtained by learning to the knowledge obtained by the learning system, when a new data block DB is formed, first use this DB to learn a new classifier, and put the classifier into the MS. At the same time, the d base classifiers in MS with the strongest correlation with the current data block are copied to the recall database ES, which is the “recall” mechanism in the MAE algorithm, where d represents the maximum capacity of ES. After the “recall” is completed, all the base classifiers stored in the current memory bank are re-evaluated according to the results obtained in this process, and the memory weight of each base classifier in the MS is updated. For the current data block, if a base classifier is recalled, the memory strength of the base classifier will be enhanced this time, and if it is not recalled, its memory strength will be weakened. When classifying the newly generated samples in the data stream, all the base classifiers in the recall database ES are directly used for classification prediction.

The algorithm model is shown in Figure 4.1. The MAE algorithm processes each new data block as follows: When a new data block arrives, a new base classifier i is learned from the data block and put into the memory bank MS, and its forgetting factor and memory strength are initialized at the same time. The system uses the freshly created DB as a validation set, chooses less than or equal to d base classifiers for all base classifiers in the memory database MS through the “recall” mechanism, and places them in the recall database ES. The system then uses all base classifiers in the current ES to classify to predict the most recent data [32]. The “recall and forget” technique is employed in the MAE algorithm to thoroughly assess the historical significance and present prediction power of the basic classifier. The stability of the method may be improved by setting up two base classifier banks: Memory bank MS and recall bank ES to keep valuable base classifiers, where MS saves all the base classifiers with weak present classification effect but strong history classification effect; In order to

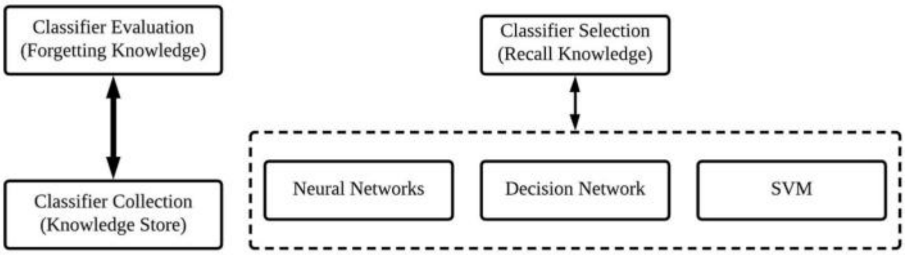

Fig. 4.2: ReMAE algorithm framework

swiftly deal with concept drift, the recall database ES stores a group of classifiers from MS that have the best prediction effect at the time. It then employs all of the base classifiers kept in ES to predict fresh samples. The MAE algorithm's learning process still has the following issues, though: For instance, the majority of the high-performance computing node status data acquired is data from nodes operating normally; failure data only makes up a minor portion of this data. To put it another way, a base classifier learned with this kind of data block will perform poorly for subsequent fault prediction because the state data of high-performance computer nodes used for fault prediction has a class imbalance phenomenon that may cause the current learning data block to contain only normal state data and no upcoming failure data [33-35].

At the same time, if the base classifier in the recall library ES is evaluated with such a data block with a severely uneven distribution of categories, the evaluation value cannot truly reflect the true classification level of the base classifier, it is even very possible to delete the base classifiers that perform well in historical classification, so that the classification effect of the ensemble classifier is poor. Therefore, considering the unbalanced distribution of data streams, based on the MAE algorithm, the author proposes an improved data stream mining algorithm ReMAE with recall and forgetting mechanisms. Aiming at the shortcomings of the data stream mining algorithm MAE with recall and forgetting mechanism in dealing with the problem of class imbalance, the author proposes a data stream mining algorithm ReMAE that considers the class imbalance problem, on the basis of the MAE algorithm model, the ReMAE algorithm mainly improves the acquisition method of the data set used to train the base classifier each time, in this algorithm, the base classifier is not directly learned from the latest DB, but is preset a sample library, the capacity of this sample library is the same as the size of the data block obtained each time, and the sample library contains the same number of sliding windows as the total number of categories in the data stream, let the size of DB be |DB|, the total number of categories of sample data is k, the number of sliding windows in the sample database is also k, and the scale of each sliding window is |DB|/k, when a new DB is formed, the samples in the DB are divided into corresponding sliding windows according to their respective categories, when the sliding window of a certain category is full, each time a new category arrives, then, the earliest inflowing samples are eliminated according to the time sequence in which the samples of this category entered the sliding window, thereby updating the sliding window and the sample library, finally, a new base classifier is learned by using all the data in the current sample library.

In the ReMAE algorithm model, after the samples in the sample database are updated, the categories of each sample remain in a balanced state, this method of setting the sliding window of the sample database converts the classification of unbalanced data into classification of balanced data, thus, the learning ability of the algorithm and the prediction effect of the classifier are improved. The ReMAE algorithm model is shown in Figure 4.2.

When there is a new predicted sample, the ReMAE algorithm, like the MAE algorithm, uses all the base classifiers in the current recall database ES to determine the class of the sample by a majority vote. When the online real-time fault prediction of "Tianhe No. 1" is carried out, the ReMAE algorithm and the normal data of some known labels and some classifiers obtained by training the fault data are deployed on each computing node, the node collects its own operating status data while running, and uses the base classifier in the current recall database ES to predict the piece of data, and uses the principle of majority voting to confirm whether

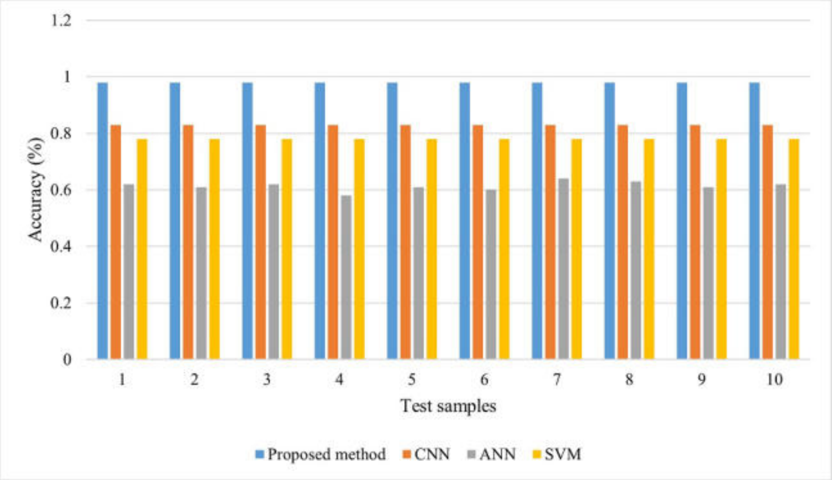

Fig. 4.3: Accuracy comparison of fault detection for different test samples

Table 4.1: Outcome comparison of several methods

Approaches	Accuracy (%)	Standard deviation
Proposed method	98.3	0.0029
CNN	83.41	0.0084
ANN	62.14	0.0205
SVM	76.41	0.0004

the piece of data is about to fail, if it is predicted that this piece of data is about to fail data or how many pieces of data are predicted to be about to fail in the accumulated continuous prediction, the system will be notified to perform process migration for the node and implement active fault tolerance, on the contrary, label the piece of data with the prediction results of most classifiers as the standard, and send the piece of data into the sliding window of the corresponding category, when a certain amount of data is accumulated and updated in the sample library (set to 500 in this experiment), a new base classifier is learned with the current sample library, all base classifiers in the memory are then updated through the "recall and forget" mechanism. Data is generated continuously, and the process is iterative.

The "Tianhe No. 1" supercomputer's node operating state data are the focus of the author's introduction and analysis of the experiments and findings of failure prediction utilising the integrated data flow approach [36]. Firstly, the disadvantages of the traditional integrated data flow classification algorithm are briefly introduced, and then the data flow mining algorithm MAE with recall and forget mechanism considering the historical evaluation results of the base classifier is introduced in detail, and the advantages and disadvantages of the MAE algorithm are explained, at the same time, based on the MAE algorithm, a data stream mining method ReMAE is proposed, which considers the classification of unbalanced samples.

To support the suggested fault detection algorithm, experimental analysis is conducted from the conventional deep learning approach and the statistical classification method.

Figure 4.3 presents the predictions made by each model, and Table 3 shows the average prediction accuracy and standard deviation for each model. The convolution neural network (CNN), artificial neural network (ANN), and support vector machine (SVM) models each attained mean predicted accuracies of 83.41%, 62.14%, and 76.41%, which is higher than that of the traditional technique. The proposed method acquired an average accuracy of 98.3%. The standard deviations given in Table 4.1 show that the proposed method is more stable than other neural network diagnosis techniques.

The experimental results demonstrate that the ReMAE algorithm has a better ability to identify the upcoming fault data, which can significantly lower the probability of the system initiating passive fault tolerance. Finally, the conventional integrated data flow mining Algorithms SEA, AWE, ACE, MEA, and ReMAE algorithms are tested based on the “Tianhe No. 1” node state data. Furthermore, the fault training and prediction times are far sufficient to fulfill the real-time demands of online learning [37, 38]. The supervised learning techniques Random Forest, Decision Trees, and Artificial Neural Networks are used in conjunction in our suggested model. Our accuracy rate was 97% when we used the NASA Turbofan dataset to test the model’s performance. Our investigation showed that the suggested approach is extremely effective at identifying hardware issues and can reliably diagnose errors. It is observed that our suggested model outperformed research utilizing comparable methodologies and datasets in terms of accuracy. Additionally, our model’s use of numerous algorithms as opposed to a single method, as in other research, helped us get superior outcomes.

5. Conclusion. Using machine learning, the present work proposes a defect detection method for the adaptive fusion of multi-source data. The integrated data flow learning and prediction method for the preprocessed data set realizes online learning and real-time malfunction prediction of high-performance computer node state data. Integrated data flow learning is an online learning technique, so it satisfies the prerequisites for online learning of node state data. When the data stream DS arrives continuously as data blocks, the algorithm adopts the strategy of first prediction and then learning during the running process. First, the algorithm uses the current learning result to predict the data block to obtain the prediction accuracy of the data block. The application of the proposed method to the defect diagnosis of an industrial system resulted in average prediction accuracy of 98.3%. The experimental results imply that the proposed method can reliably combine multiple signals, extricate precise data, and universally identify equipment defects. This method of learning and predicting is not only an online learning method, but also meets the requirements of defect prediction in real time. Our suggested methodology demonstrated its capability for identifying and treating hardware issues. Future studies might concentrate on expanding the model’s datasets, enhancing the model’s scalability, and creating methods for finding errors in intricate hardware systems. This research makes a substantial contribution by offering dependable and effective techniques for resolving hardware issues, resulting in more dependable and effective technological systems.

REFERENCES

[1] Marty, T., Yuki, T., & Derrien, S. , Safe overclocking for cnn accelerators through algorithm-level error detection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, PP(99), 1-1, 2020.

[2] Deng, D., Wang, Y., & Guo, Y. , Novel design strategy toward a2 trojan detection based on built-in acceleration structure. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, PP(99), 1-1, 2020.

[3] Zhang, Y., Luan, Z., Ding, D., Wang, P., Li, Z., & Li, L. , Design of sh aging sensor for real time and application in sensing network. Canadian Journal of Electrical and Computer Engineering, 43(2), 73-82, 2020.

[4] Sengupta, A., Nabeel, M., Limaye, N., Ashraf, M., & Sinanoglu, O. , Truly stripping functionality for logic locking: a fault-based perspective. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, PP(99), 1-1, 2020.

[5] Li, Z., Menon, H., Dan, M., Livnat, Y., & Pascucci, V. , Spotsdc: revealing the silent data corruptionpropagation in high-performance computing systems. IEEE Transactions on Visualization and Computer Graphics, PP(99), 1-1, 2020.

[6] Aggarwal, A., Allafi, I. M., Strangas, E. G., & Agapiou, J. S. , Off-line detection of static eccentricity of pmsm robust to machine operating temperature and rotor position misalignment using incremental inductance approach. IEEE Transactions on Transportation Electrification, PP (99), 1-1, 2020.

[7] Wu, Y., Liu, L., Wang, L., Wang, X., & Wei, S. , Aggressive fine-grained power gating of noc buffers. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(11), 3177-3189, 2020.

[8] Liu, X., Qian, Y., Feng, X., Zhuo, Q., & Li, Y. , Dynamic placement optimization for bio-inspired self-repairing hardware. IEEE Access, PP(99), 1-1, 2020.

[9] Yuan, G., Xu, Z., Yang, B., Liang, W., Chai, W. K., & Tuncer, D. , Fault tolerant placement of stateful vnfs and dynamic fault recovery in cloud networks. Computer networks, 166(Jan.15), 106953.1-106953.18, 2020.

[10] Cotroneo, D., Simone, L. D., Liguori, P., & Natella, R. , Fault injection analytics: a novel approach to discover failure modes in cloud-computing systems. IEEE Transactions on Dependable and Secure Computing, PP(99), 1-1, 2020.

[11] Çinar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. , Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability, 12(19), 8211, 2020.

[12] Xu, J., Li, F., Leier, A., Xiang, D., Shen, H. H., Marquez Lago, T. T., & Song, J. , Comprehensive assessment of machine learning-based methods for predicting antimicrobial peptides. Briefings in Bioinformatics, 22(5), bbab083, 2021.

[13] Koziel, S., Çalik, N., Mahouti, P., & Belen, M. A., Reliable computationally efficient behavioral modeling of microwave passives using deep learning surrogates in confined domains. IEEE Transactions on Microwave Theory and Techniques, 71(3), 956-968, 2022.

[14] Shao, S. Y., Sun, W. J., Yan, R. Q., Wang, P., & Gao, R. X., A deep learning approach for fault diagnosis of induction motors in manufacturing. Chinese Journal of Mechanical Engineering, 30(6), 1347-1356, 2017.

[15] Hanga, K. M., & Kovalchuk, Y. , Machine learning and multi-agent systems in oil and gas industry applications: A survey. Computer Science Review, 34, 100191, 2019.

[16] Zhang, C., Zhou, G., Li, J., Chang, F., Ding, K., & Ma, D. , A multi-access edge computing enabled framework for the construction of a knowledge-sharing intelligent machine tool swarm in Industry 4.0. Journal of Manufacturing Systems, 66, 56-70, 2023.

[17] Karaoğlu, U., Mbah, O., & Zeeshan, Q., Applications of machine learning in aircraft maintenance. J. Eng. Manag. Syst. Eng, 2(1), 76-95, 2023.

[18] Chen, Z., O'Neill, Z., Wen, J., Pradhan, O., Yang, T., Lu, X., ... & Herr, T., A review of data-driven fault detection and diagnostics for building HVAC systems. Applied Energy, 339, 121030, 2023.

[19] Keleko, A. T., Kamsu-Foguem, B., Ngouna, R. H., & Tongne, A., Health condition monitoring of a complex hydraulic system using Deep Neural Network and DeepSHAP explainable XAI. Advances in Engineering Software, 175, 103339, 2023.

[20] Kouliaridis, V., & Kambourakis, G. , A comprehensive survey on machine learning techniques for android malware detection. Information, 12(5), 185, 2021.

[21] Viera, R., Maurine, P., Dutertre, J. M., & Bastos, R. P. , Simulation and experimental demonstration of the importance of ir-drops during laser fault injection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(6), 1231-1244, 2020.

[22] Pomeranz, I. , Maximal independent fault set for gate-exhaustive faults. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, PP(99), 1-1, 2020.

[23] Gao, F., Li, F., Wang, Z., Ge, W., & Li, X. , Research on multilevel classification of high-speed railway signal equipment fault based on text mining. Journal of Electrical and Computer Engineering, 2021(2), 1-11, 2021.

[24] Kung, Y. C., Lee, K. J., & Reddy, S. M. , Generating single- and double-pattern tests for multiple cmos fault models in one atpg run. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(6), 1340-1345, 2020.

[25] Zhang, X., Yang, G., Cui, Y., Wei, X., & Qiao, W. , Application of computer algorithm in fault diagnosis system of rm equipment. Journal of Physics: Conference Series, 2143(1), 012033, 2021.

[26] Gao, Y., Tao, J., Xu, Y., Wang, Z., & Cheng, G. , Cebd: contact evidence-driven blackhole detection based on machine learning in oppnets. IEEE Transactions on Computational Social Systems, PP(99), 1-13, 2021.

[27] Ehrnsperger, M. G., Brenner, T., Hoese, H. L., Siart, U., & Eibert, T. F. , Real-time gesture detection based on machine learning classification of continuous wave radar signals. IEEE Sensors Journal, PP(99), 1-1, 2020.

[28] Moulahi, T., Zidi, S., Alabdulatif, A., & Atiquzzaman, M. , Comparative performance evaluation of intrusion detection based on machine learning in in-vehicle controller area network bus. IEEE Access, PP(99), 1-1, 2021.

[29] Yao, J., Zhang, Y., & Xin, C. , Network-on-chip hardware trojan detection platform based on machine learning. Journal of Physics: Conference Series, 2189(1), 012004, 2022.

[30] Chen, X., Yu, S., Zhang, Y., Chu, F., & Sun, B. , Machine learning method for continuous noninvasive blood pressure detection based on random forest. IEEE Access, PP(99), 1-1, 2021.

[31] Le, V., Yao, X., Hung, T. B., & Miller, C. , Series dc arc fault detection based on ensemble machine learning. IEEE Transactions on Power Electronics, PP(99), 1-1, 2020.

[32] Zhao, Y., Li, L., Wang, H., Cai, H., & Grundy, J. , On the impact of sample duplication in machine-learning-based android malware detection. ACM Transactions on Software Engineering and Methodology, 30(3), 1-38, 2021.

[33] Zhang, P., Gao, D., Hong, D., Lu, Y., Wu, Q., Zan, S., & Liao, Z., Improving generalisation and accuracy of on-line milling chatter detection via a novel hybrid deep convolutional neural network. Mechanical Systems and Signal Processing, 193, 110241, 2023.

[34] Yücel, M., & Açikgöz, M., Optical Communication Infrastructure in New Generation Mobile Networks. Fiber and Integrated Optics, 42(2), 53-92, 2023.

[35] Di Capua, M., Ciaramella, A., & De Prisco, A., Machine learning and computer vision for the automation of processes in advanced logistics: The integrated logistic platform (ILP) 4.0. Procedia Computer Science, 217, 326-338, 2023.

[36] He, Y., Zhang, H., Arens, E., Merritt, A., Huizenga, C., Levinson, R., ... & Alvarez-Suarez, A., Smart detection of indoor occupant thermal state via infrared thermography, computer vision, and machine learning. Building and Environment, 228, 109811, 2023.

[37] FAlsamawi, F. N., & Kurnaz, S., A framework for adopting gamified learning systems in smart schools during COVID-19. Applied Nanoscience, 13(2), 1135-1153. 2023.

[38] Sun, Q., Shi, H., Li, Y., Zhu, Q., & Ren, Z., Online Extrinsic Calibration of RGB and ToF Cameras for Extraterrestrial Exploration. In 2023 42nd Chinese Control Conference (CCC) (pp. 7447-7452). IEEE, 2023.

Edited by: Pradeep Kumar Singh
Special issue on: Intelligent Cloud Technologies Enabled Solutions for Next Generation Smart Cities
Received: Mar 16, 2023
Accepted: Nov 13, 2023