Text Sentiment Detection and Classification Based on Integrated Learning Algorithm

Zheng Lin[1], Zeyu Wang[2], Yue Zhu[3], Zichao Li[4] and Hao Qin[5]
[1]*Independent, China*
[2]*University of California, Los Angeles, USA*
[3]*Independent, China*
[4]*Canoakbit Alliance Inc., Canada*
[5]*Independent, China*

[1]*Corresponding Author: zhenglin1238@gmail.com*

Received: 22-04-2024 Revised: 15-05-2024 Accepted: 29-05-2024

ABSTRACT
The aim of this paper is to explore the importance of textual sentiment detection in the field of Natural Language Processing and to classify and detect sentiment through various machine learning algorithms. Firstly, we train using Park Bayes, Random Forest, XGB and Support Vector Machine models, and then integrate them into a voting classifier for comparative analysis. The results show that the Random Forest model performs the best in the training set; and in both the validation set and the test set, the accuracy of the voting classifier is the highest, reaching 93.32% and 94.47%, respectively, which shows its superiority in the classification of text sentiment detection. Taken together, voting classifier has the best prediction results and provides an effective solution for text sentiment detection. This study not only provides an in-depth comparative analysis of the performance of different machine learning algorithms in text sentiment detection, but also provides a useful reference for subsequent related research and applications.

Keywords: *text emotion, integrated learning, machine learning*

I. INTRODUCTION

Text sentiment detection is an important research direction in the field of natural language processing, aiming at identifying and classifying the sentiment or emotional tendency contained therein by analysing the text content [1]. With the popularity of social media and the Internet, people generate a large amount of text data on the Web, which contains rich emotional information. Therefore, automated detection and classification of text emotions is of great significance, which can help enterprises to understand users' attitudes towards products or services, help governments to monitor public opinion trends, and also help individuals to understand others' emotional tendencies [2].

Machine learning algorithms play a crucial role in text sentiment detection. Traditional rule- and dictionary-based methods often rely on manually constructing rules or dictionaries, making it difficult to adapt to different contexts and expressions [3]. Machine learning algorithms, on the other hand, are able to learn features and patterns from data by training models for more accurate and efficient text sentiment classification.

A commonly used machine learning algorithm is the Support Vector Machine (SVM) [4].SVMs are able to efficiently handle high-dimensional data and perform well on binary and multi-classification problems. In textual emotion detection, after representing the text in vector form, SVM models can be trained using SVMs to achieve classification of different emotion categories. In addition, deep learning algorithms such as Recurrent Neural Networks (RNN) [5] and Convolutional Neural Networks (CNN) [6] are also widely used in text sentiment detection tasks.RNN is suitable for processing sequential data and capturing contextual information when analysing text, while CNN is good at extracting local features and discovering important patterns in text. In addition to traditional supervised learning algorithms, unsupervised learning methods such as Topic Model [7] are also used in text sentiment detection. Topic modelling can discover the topic structure and sentiment tendency hidden behind a large amount of text data, providing clues for further analysis.

Machine learning algorithms play a key role in text sentiment detection, which not only improves classification accuracy and efficiency, but also provides researchers with the possibility of exploring deeper and more complex sentiment expressions. In this paper, we first detect and classify text emotions based on multiple machine learning algorithms, and then integrate multiple machine learning algorithms for integrated learning to compare and analyse their advantages and disadvantages in text emotion detection and classification.

II. SOURCE OF DATA SETS

This paper uses an open source dataset to conduct experiments, the dataset contains a variety of this paper as well as the corresponding emotion labels of the text, the emotion labels include joy, sadness, fear,anger, love and surprise, etc. In this paper, the dataset is divided into the training set, validation set and test set according to the ratio of 4:3:3. The proportion of each label in the training set, validation set and test set is counted and the results are shown in Figure 1 below.

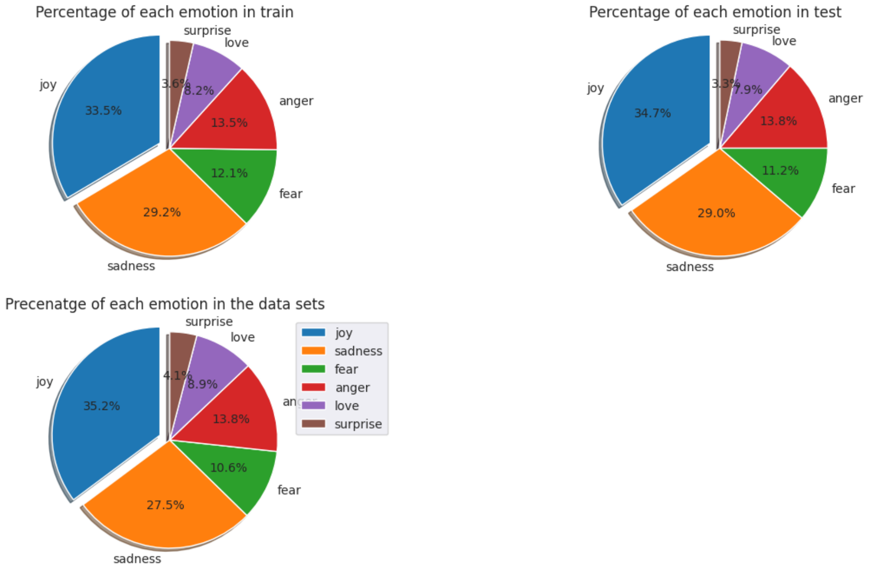

Figure 1: Statistical analysis of data
（**Photo Credit:** Original）

III. METHOD

3.1 Plain Bayes

Plain Bayes is a classification algorithm based on Bayes' theorem and the assumption of conditional independence of features. Its principle is simple but effective and is commonly used in text classification, spam filtering, sentiment analysis and other fields. In the plain Bayesian model, we predict which category the new data belongs to by calculating the probability of each category under a given data set.

Firstly, the plain Bayesian model is based on Bayes' theorem, i.e., the probability that the sample to be classified belongs to each category is calculated based on the training data, and then the category with the highest probability is chosen as the prediction result. There is a "plain" assumption in plain Bayes, i.e., it is assumed that all features are independent of each other. This means that the effect of each feature on classification is independent of each other, given the categories. Although this assumption does not always hold in reality, it works well in practice and makes the computation simple and efficient.

In practice, the plain Bayesian model is often combined with smoothing techniques to deal with sparse data and avoid the zero probability problem. Common plain Bayes algorithms include Gaussian plain Bayes, Polynomial plain Bayes and Bernoulli plain Bayes.

3.2 Random Forest

Random Forest is an integrated learning method for prediction or classification by integrating multiple decision tree models. Its principle is based on the idea of Bagging and random feature selection, which has the advantages of high accuracy, robustness and the ability to handle large amounts of data. Random Forest consists of multiple decision trees, each of which is a base learner. When constructing each decision tree, a subset of the training set is randomly sampled for training, a sampling method known as self-sampling. This ensures that the training set for each decision tree is slightly different, increasing the diversity of the model.

Instead of dividing the nodes using all the features each time they are split, a random selection of features from all the features is considered. This random selection of features reduces the correlation between models and further improves the generalisation of the overall model [8].

When predicting, the random forest will vote or average the results of each decision tree to get the final prediction. Since each tree may be overfitting or misclassified, the variance can be reduced by integrating the results of multiple models to improve the stability and accuracy of the overall model. Random Forest uses Bagging and random feature selection to construct multiple decision trees and integrates the sub-models by integrating learning, thus effectively solving the problems of a single decision tree that is prone to overfitting and weak generalisation [9].

3.3 XGB

XGB model is an integrated learning algorithm, which is based on the gradient boosting framework and written in C++, but supports multiple programming languages at the same time.XGB model is mainly based on decision tree integration and gradient boosting techniques. During the training process, XGB gradually improves the model performance by constructing multiple decision trees. Each decision tree is trained based on the residual error of the previous tree in order to make the overall model gradually approach the true value. This residual learning approach allows XGB to better fit the data and reduce the risk of overfitting.

When constructing each decision tree, XGB uses regularisation techniques such as learning rate, tree depth, subsampling and other parameters to control the model complexity. In addition, XGB introduces a gradient boosting technique to minimise the loss function by calculating the negative gradient of the loss function to update the model parameters in each iteration. This gradient descent allows XGB to converge faster and achieve better performance [10].

By integrating multiple decision trees, using gradient boosting techniques and regularisation methods to continuously optimise the model, XGB models perform well on various types of datasets and are widely used in machine learning tasks such as classification and regression.

3.4 SVM

Support vector machine is a commonly used supervised learning algorithm for solving classification and regression problems. Its principle is based on finding an optimal hyperplane that effectively separates data points of different categories in the feature space.The core idea of SVM is to maximise the classification boundaries[11], i.e., to find the hyperplane that maximises the separation between two categories.[12] In SVM, support vectors refer to those data points that are closest to the hyperplane, and they play a key role in defining the hyperplane.

SVM maps the data to a high-dimensional feature space; then, a hyperplane is found in the feature space that can separate the different categories; then, the hyperplane that maximises the classification boundaries is found by an optimisation algorithm; and finally, in the testing phase, the new data points are mapped to the feature space and classified according to their positions.[13]

SVM has a variety of kernel functions to choose from, such as linear kernel, polynomial kernel, Gaussian kernel[14], etc. Different kernel functions are suitable for different types of datasets.SVM has good generalisation ability and strong robustness, and it performs well in dealing with small samples, non-linear and high dimensional data.

3.5 Voting Classifier

Voting Classifier is an integrated learning method that makes the final classification decision by combining the predictions of multiple base classifiers for voting.The model structure of Voting Classifier is shown in Fig. 2, which integrates Simple Bayes, Random Forest[14], XGB and Support Vector Machine models.

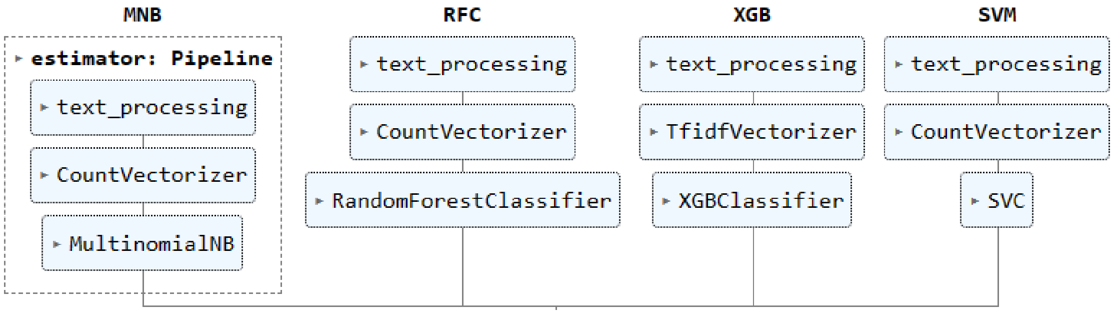

Figure 2: Integrated Learning
（**Photo Credit:** Original）

Voting Classifier[15] can contain different types of base classifiers, such as Support Vector Machine (SVM), Logistic Regression, Decision Tree, and so on. By combining multiple algorithms, Voting Classifier can make up for the shortcomings of individual algorithms and improve the generalisation ability and accuracy of the overall model.As an integrated learning method, Voting Classifier has strong robustness and generalisation ability in practical applications, which can effectively enhance model performance and improve prediction accuracy.

IV. EXPERIMENTS AND RESULTS

Simple Bayes, Random Forest, XGB and Support Vector Machine models are introduced for training respectively, and the prediction confusion matrices for the training, validation and test sets of each model are output, and the results are shown in Fig. 3, Fig. 4, Fig. 5 and Fig. 6.

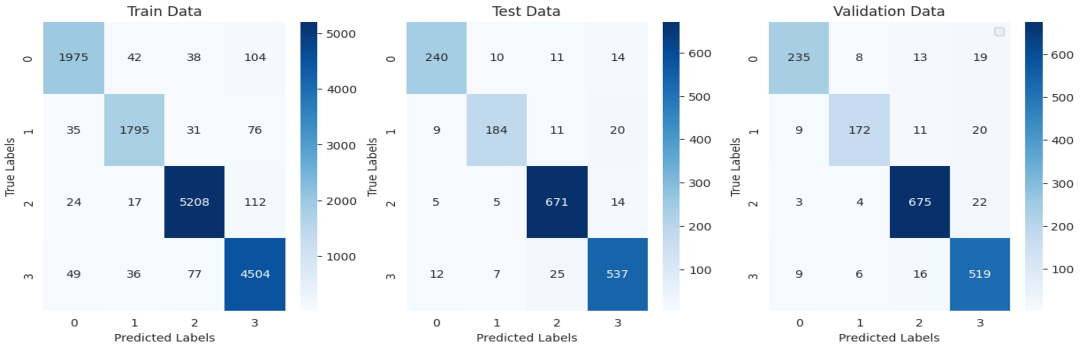

Figure 3: Confusion matrix
（**Photo Credit:** Original）

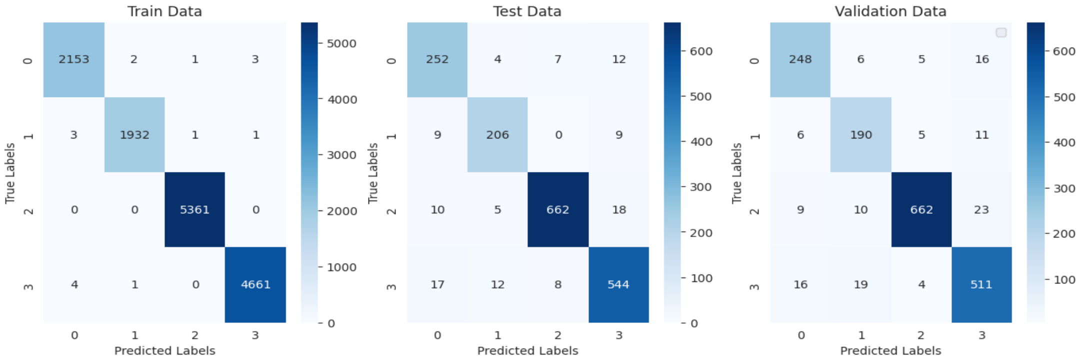

Figure 4: Confusion matrix
（**Photo Credit:** Original）

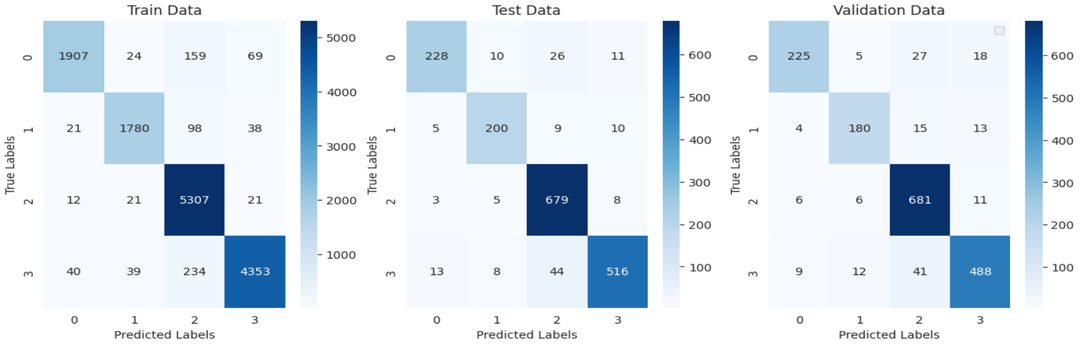

Figure 5: Confusion matrix
（**Photo Credit:** Original）

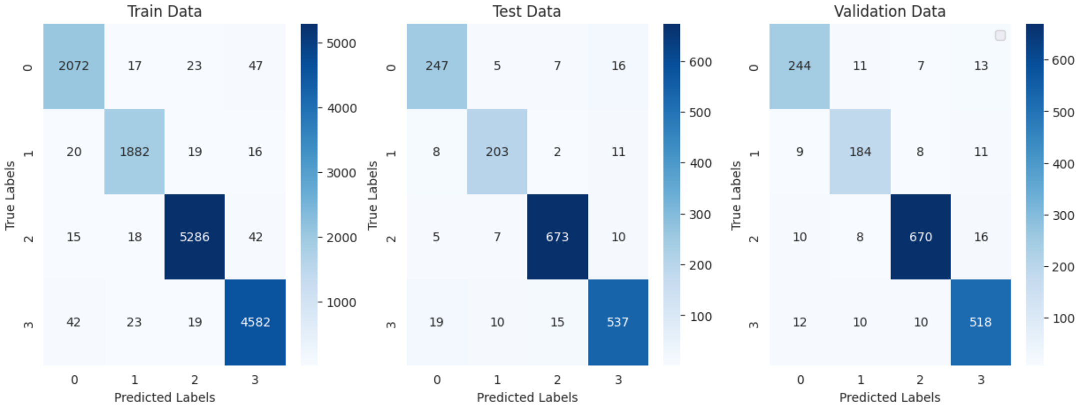

Figure 6: Confusion matrix
（**Photo Credit:** Original）

The classification effect of voting classifier training set, validation set and test set is output and the results are shown in Fig. 7.

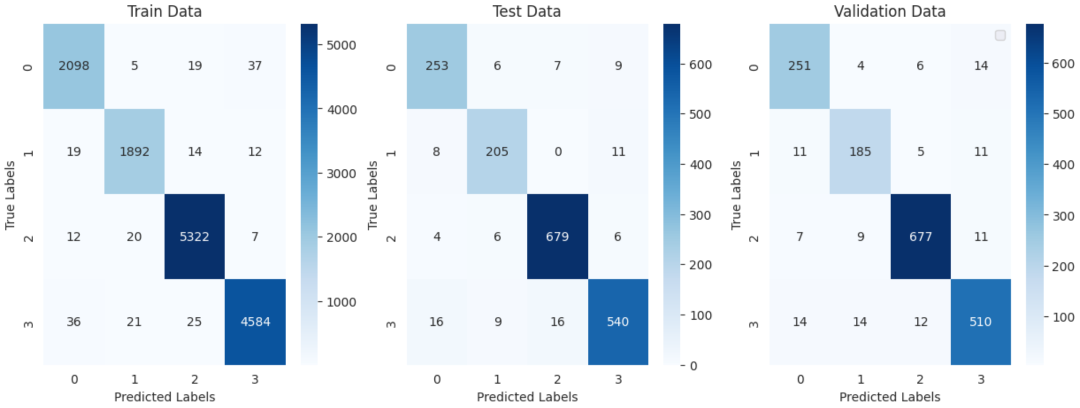

Figure 7: Confusion matrix
（**Photo Credit:** Original）

The four base models and voting classifier are put together and compared to compare the accuracy of each model in terms of training set, validation set and test set and the results are shown in Table 1. The histogram of the accuracy of each model with respect to training set, validation set and test set is output as shown in Fig. 8.

Table 1: Modelling assessment

	Naive bayes	Random Forest	XGB	SVM	Voting classifier
Train	0.9546	0.9988	0.945	0.9786	0.9839
Test	0.9194	0.9374	0.9143	0.9352	0.9447
Validate	0.9195	0.9253	0.904	0.9282	0.9322

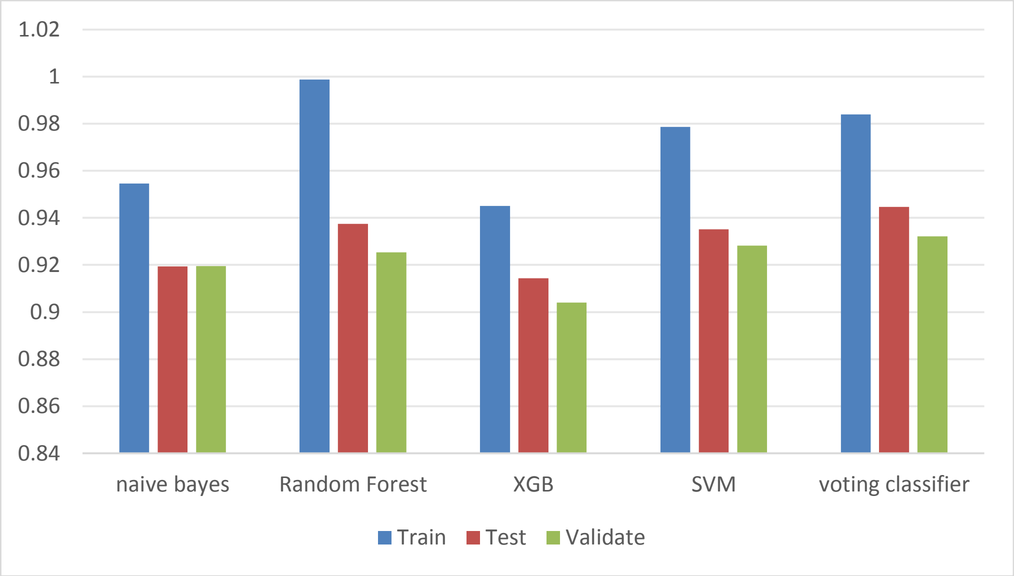

Figure 8: Modelling assessment
（**Photo Credit:** Original)

As can be seen from Table 1, the best accuracy in the training set is the Random Forest model, while the best accuracy in both the validation set and the test set is the voting classifier, the accuracy of the voting classifier in the training set is 98.39%, the accuracy of the voting classifier in the test set is 94.47%, and the validation set The accuracy of voting classifier is 93.32% and voting classifier has the best prediction.

V. CONCLUSION

The aim of this paper is to explore the importance of text sentiment detection in the field of natural language processing, and to perform sentiment detection and classification by means of various machine learning algorithms, and finally to compare and analyse the results using integrated learning methods with a view to finding the most effective model. In our study, we introduced classical machine learning algorithms such as Park Bayes, Random Forest, XGBoost and Support Vector Machine for training.

Comparison of the experimental results reveals that in the training set, the Random Forest model performs well and achieves the best accuracy; however, in the validation and test sets, the Voting Classifier model demonstrates even better prediction performance. Specifically, Voting Classifier achieved 98.39% accuracy in the training set, 94.47% in the testing set, and 93.32% in the validation set. This result indicates that Voting Classifier as an integrated learning method can effectively improve the model performance and achieve the best prediction results in the text sentiment detection classification task.

Further analysis reveals that the reason why Voting Classifier can achieve high accuracy rates on both validation and test sets may be because it combines the prediction results of multiple base classifiers, effectively balancing the strengths and weaknesses between individual algorithms. In contrast, a single algorithm, although outstanding in some aspects, often fails to cover the best solution in all cases.

REFERENCES

1. J. Jin, F. Ni, S. Dai, K. Li, & B. Hong. (2024). Enhancing federated semi-supervised learning with out-of-distribution filtering amidst class mismatches. *Journal of Computer Technology and Applied Mathematics, 1*(1), 100-108.
2. S. Li, Y. Mo, & Z. Li. (2022). Automated pneumonia detection in chest x-ray images using deep learning model. *Innovations in Applied Engineering and Technology*, 1-6.
3. Z. Li, H. Yu, J. Xu, J. Liu, & Y. Mo. (2023). Stock market analysis and prediction using lstm: A case study on technology stocks. *Innovations in Applied Engineering and Technology*, 1-6.
4. K. Li, P. Xirui, J. Song, B. Hong, & J. Wang. (2024). *The application of augmented reality (ar) in remote work and education*. arXiv preprint arXiv:2404.10579.

5. K. Li, A. Zhu, P. Zhao, J. Song, & J. Liu. (2024). Utilizing deep learning to optimize software development processes. *Journal of Computer Technology and Applied Mathematics, 1*(1), 70-76.
6. T. Liu, S. Li, Y. Dong, Y. Mo, & S. He. (2024). Spam detection and classification based on distilbert deep learning algorithm. *Applied Science and Engineering Journal for Advanced Research, 3*(3), 6-10.
7. Y. Mo, H. Qin, Y. Dong, Z. Zhu, & Z. Li. (2024). Large language model (llm) ai text generation detection based on transformer deep learning algorithm. *International Journal of Engineering and Management Research, 14*(2), 154-159.
8. Y. Mo, S. Li, Y. Dong, Z. Zhu, & Z. Li. (2024). Password complexity prediction based on roberta algorithm. *Applied Science and Engineering Journal for Advanced Research, 3*(3), 1-5.
9. J. Zhang, A. Xiang, Y. Cheng, Q. Yang, & L. Wang. (2024). *Research on detection of floating objects in river and lake based on ai intelligent image recognition*. arXiv preprint arXiv:2404.06883.
10. J. Song, H. Liu, K. Li, J. Tian, & Y. Mo. (2024). A comprehensive evaluation and comparison of enhanced learning methods. *Academic Journal of Science and Technology, 10*(3), 167-171.
11. A. Zhu, K. Li, T. Wu, P. Zhao, & B. Hong. (2024). Cross-task multi-branch vision transformer for facial expression and mask wearing classification. *Journal of Computer Technology and Applied Mathematics, 1*(1), 46-53.
12. Li, Huan, Feng Xu, & Zheng Lin. (2023). ET-DM: Text to image via diffusion model with efficient Transformer. *Displays,* 80.
13. Lin, Zheng, & Feng Xu. (2023). Simulation of robot automatic control model based on artificial intelligence algorithm. *2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)*.
14. Qiu, Shushan, et al. (2022). Day-ahead optimal scheduling of power–gas–heating integrated energy system considering energy routing. *Energy Reports, 8*(2022), 1113-1122.
15. Chen, Jinfan, et al. (2023). Stochastic planning of integrated energy system based on correlation scenario generation method via Copula function considering multiple uncertainties in renewable energy sources and demands. *IET Renewable Power Generation 17*(12), 2978-2996.
16. Chen, Jinfan, et al. (2024). Reinforcement learning based two-timescale energy management for energy hub. *IET Renewable Power Generation 18*(3), 476-488.
17. Zhan, Rongrong, et al. (2023). Operation strategy of energy router considering compressed air energy storage. *4th International Conference on Advanced Electrical and Energy Systems (AEES)*.
18. Chen, Jinfan, et al. (2023). Robust optimization based multi-level coordinated scheduling strategy for energy hub in spot market. *7th International Conference on Green Energy and Applications (ICGEA)*.
19. Li, Zhuoying, et al. (2024). *AD-aligning: Emulating human-like generalization for cognitive domain adaptation in deep learning*. arXiv preprint arXiv:2405.09582.
20. Meng, Yinan, et al. (2023). Spring-IMU fusion-based proprioception for feedback control of soft manipulators. *IEEE/ASME Transactions on Mechatronics*.