Distributed Fire Agent Model based on Ensemble Clustering

Huanhuan Chen

Logistics Research Center Shanghai Maritime University Shanghai, China

2782080028@qq.com

Abstract. Fire Dynamics Simulation (FDS) is a numerical simulation method based on physical principles, which can reflect the fire development process and its influencing factors relatively realistically. However, FDS also has problems such as large computational volume and long simulation time. And due to the complexity of fire data, single-agent models have limitations and risks in expressing fire characteristics, model complexity, computational time and overfitting. To solve the above problems, an adaptive distributed fire agent model is proposed. Using the Spark distributed computing framework, integrated cluster analysis is used to dynamically select the best agent model for each cluster of data, thus taking full advantage of different agent models and improving the accuracy, robustness and generalization ability of fire prediction. The experimental results show that the distributed fire agent model has advantages in prediction accuracy, adaptability, data security and computation time, which meets the needs of modern fire numerical simulation research.

Keywords: Fire numerical simulation, Distributed computing, Agent model, Integrated learning, Security management

1 Introduction

Fire prediction and field simulation face many challenges. Fire numerical simulation has become a new research method and prediction tool. It can accurately simulate the complex physical processes in fire scenarios and provide rich fire data, which is helpful to predict the development trend of fire, evaluate the fire risk and optimize the rescue and evacuation work. Based on the principle of fire dynamics, the fire dynamics model (FDS) [1]can accurately simulate the changes of key parameters such as temperature, CO and CO2 concentration, visibility and smoke flow when a fire occurs, and accurately reproduce the development of fire under different buildings and conditions. However, the calculation of FDS model takes a long time, which can be as long as several hours and months, which seriously affects the timeliness of fire trend prediction and is not conducive to timely fire rescue.

Surrogate model technology is a method of using a simplified mathematical model to approximately reflect the performance and characteristics of the original model in a complex simulation model. A single surrogate model may not be able to fully learn and

S. M. Ghadiri et al. (eds.), *Proceedings of the 2025 4th International Conference on Engineering Management and Information Science (EMIS 2025)*, Advances in Computer Science Research 121,
https://doi.org/10.2991/978-94-6463-736-6_33

express all the fire source characteristics. Distributed machine learning can take advantage of the advantages of multiple surrogate models to improve prediction accuracy and shorten prediction time [2]. Based on the fire data of FDS simulation, this paper uses the distributed computing framework and voting ensemble clustering algorithm to cluster the data first, so that the data points in each cluster have similar characteristics. Then, according to the similarity and difference between the data points in each cluster, the appropriate machine learning method is adaptively selected to predict the output of all data points in the cluster. The advantage of this algorithm is to use the distributed computing framework to process a large number of data points in parallel, shorten the running time and reduce the memory consumption. It can adapt to different types and scales of fire scene data, and can further improve the reliability of the fire agent model to meet the complexity requirements of modern fire scenes.

2 Method

2.1 Voting Ensemble Clustering

Clustering ensemble is a semi-supervised learning algorithm based on the characteristics of ensemble learning, which combines ensemble learning and clustering algorithm. Clustering ensemble, as an extended version of traditional clustering, was first proposed by Strehl [3]. Its core idea is to output partition results with better performance than a single clustering method by integrating multiple base clustering results. Usually, in the clustering ensemble model, for a given data set $DX = \{x_1, x_2, \cdots, x_j, \cdots, x_N\}$ with N sample points, a base clustering algorithm performs L divisions of the dataset DX to obtain L base clustering results $\prod = \{\pi 1, \pi 2, \cdots, \pi L\}$, and then uses the consensus function to fuse the generated base clustering results to obtain the final consensus clustering results.

This paper adopts a voting-based method, and the majority voting method is a simple and intuitive clustering ensemble algorithm. In the process of clustering ensemble, the majority voting method divides the data objects into the clusters selected by the majority clustering results. Suppose that the data set X, which is divided into C clusters, has M base clustering results for the data set, and D indicates whether a data is classified into the J class in the first clustering result. The majority voting method determines that a data is assigned to class k as shown in formula (1).

$$\sum_{i=1}^{m} d_{ik} = \max_{j=1}^{c} \sum_{j=1}^{c} d_{ij} \tag{1}$$

2.2 Distributed Machine Learning

Distributed machine learning achieves parallelization and collaborative processing of computational tasks by decomposing and distributing machine learning tasks to multiple computational nodes. Each computational node is responsible for processing a portion of the dataset and performing model training locally [4]. Through efficient communication and synchronization mechanisms, each node can share the training results and jointly optimize the global model. This distributed training approach not only greatly improves the speed and efficiency of data processing, but also enables machine learning models to scale to larger datasets and more complex model structures. In addition, distributed machine learning also provides better fault tolerance and robustness, so that even if some of the computing nodes fail, the whole system can still continue to run. Spark is an open source distributed computing framework that is based on in-memory computing to efficiently process large-scale data and supports a variety of data processing modes, such as batch processing, stream processing, machine learning, and graph computation. Spark provides a rich set of APIs and tools, such as RDD, Data Frame, Spark SQL, Spark Streaming, Graph X and MLlib, etc., enabling users to easily perform data processing, analysis, and mining, etc. Spark speeds up data processing and analysis by spreading tasks across multiple computers and executing them in parallel.

2.3 Distributed Fire Agent Model

Distributed fire agent model combines the advantages of distributed computing framework and voting integrated clustering algorithm for effective clustering and prediction of fire data. The core idea of the distributed fire agent model is: under the distributed framework, the voting integrated clustering algorithm is performed first, which uses multiple k-means clustering algorithms with different initializations to divide the fire data into a number of clusters, in which the data points with similar fire characteristics are grouped into one cluster; then, the data can be partitioned for the number of clusters divided into clusters, and cross validation of the fire data of different clusters is performed in parallel to train the fire data from the Different agent models, such as support vector machine, linear regression, random forest, etc., are selected for each cluster to predict the fire state parameters in each cluster; finally, the prediction results in each cluster are summarized to obtain the overall prediction results of the fire state parameters. The advantages of the distributed fire agent model are mainly reflected in two aspects: on the one hand, it utilizes the clustering ability of the voting integrated clustering algorithm and the fitting ability of the agent model, so that the most suitable agent model can be trained and selected according to the characteristics of different subregions and different clusters, which improves the accuracy and reliability of fire prediction; on the other hand, it takes advantage of the parallelism and scalability of the distributed computation framework, which can quickly process large-scale fire data, saving computational time and resources.

3 Experimental Results and Analysis

3.1 FDS Fire Simulation

In this study, a physical model of an indoor room [5] was established by using FDS fire dynamics software, see Figure 1.The length, width and height of the model were 5m, 3m and 3m respectively, and there was only one door as the entrance/exit, which was 0.8m in width and 2m in height. The fire sources were set up to be uniformly distributed in the room, and the size of each fire source was 0.5m*0.5m, and the distances from the walls were 0.25m, with a total of 9 fire sources with an interval of 1.5m. the interval between neighboring fire sources is 1.5m, totaling 9 places. In this experiment, six different room temperature values were set, the lowest room temperature was 5°C, and one room temperature value was set every 5 degrees Celsius. In this experiment, we focused on the smaller interval of the heat release rate of the fire source, setting the maximum value to 4000 watts and the minimum value to 1500 watts, with a total of 15 different intervals of the heat release rate of the fire source. The specific room temperature and fire source heat release rate value ranges are shown in Table 1. The experimental dataset for this experiment consists of 810 sets of fire sample data obtained from FDS simulation, and each set of data consists of values of different state parameters of the fire occurrence (temperature, CO concentration, CO_2 concentration, visibility) varying per second with the simulation time of 1000s

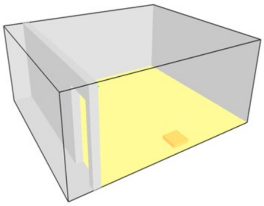

Fig. 1. Indoor Physical Modeling

Table 1. The value of temperature and fire heat release rate

Parameter	Minimum	Maximum	Specific rounding
Temperature (°C)	5	30	5,10,15,20,25,30
Heat release rate (w)	1500	4000	1500,1600,1700,1800,1900,2000,2200, 2400,2600,2800,3250,3500,3750,4000

3.2 Analysis of the effectiveness of distributed fire agent model fitting

Based on the parameter settings described in the previous section, we constructed a fire agent model using the training set and evaluated it with the test set. In order to visualize the prediction effect of the agent model, we plotted the fitted scatter plots of the four fire state parameters (temperature, CO concentration, CO_2 concentration, and visibility). Figure 2 demonstrates the predicted values of these four state parameters, and Figures 2(a), 2(b), 2(c), and 2(d) show the predicted effects of temperature, CO concentration, CO2 concentration, and visibility, respectively, where the horizontal axis represents the FDS simulation data, and the vertical axis represents the predicted values of the fire agent model. As can be seen in Figure 2, most of the data points in the graph are concentrated above or near the fitting diagonal, and the distribution of the data points shows that the predicted values of the fire proxy model basically coincide with the FDS simulation values, with high fitting accuracy.

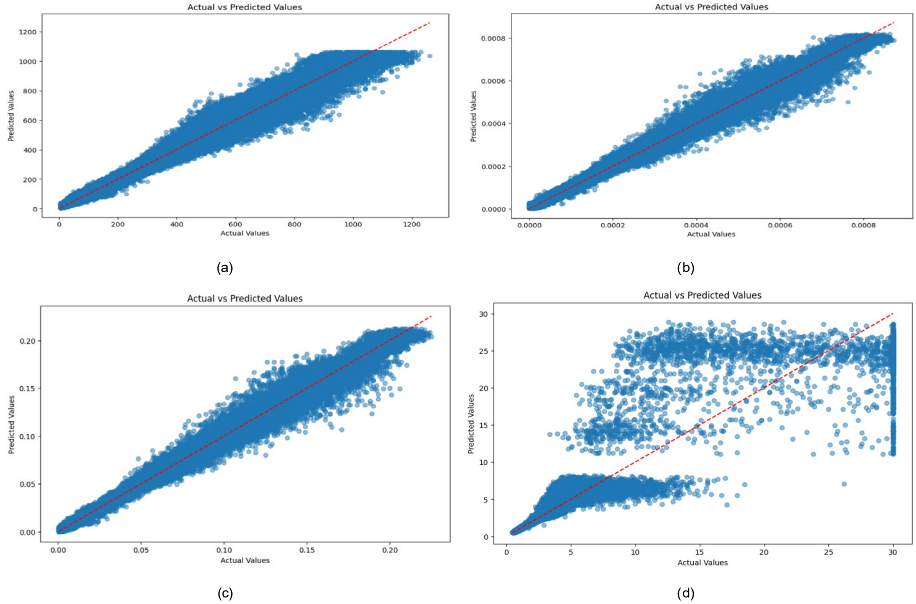

Fig. 2. Distributed fire agent model prediction results vs. FDS simulation results (a) Temperature (b) CO concentration (c) CO2 concentration (d) Visibility

3.3 Comparative Analysis of Distributed Fire Agent Model Predictions

In order to validate the prediction performance of the distributed fire agent model model, two evaluation metrics, Root Mean Square Error (RMSE) and Absolute Percentage Error (APE), are used to compare with the currently available fire agent proxy models. The prediction performance of the model on each metric is shown in Table 2, Table 3, Table 4, and Table 5. By comparing and analyzing the RNSE and APE evaluation indexes in the list, the distributed agent model has reduced RMSE and APE compared to the single agent model, and we can conclude that the distributed fire agent model in this experiment has an advantage in fire state parameter prediction over the single agent model in terms of both prediction accuracy and adaptability.

Table 2. CO prediction results of different algorithms

Algorithms	RMSE	APE (%)
DT	1.74E-5	5.15%
RF	3.34E-5	5.21%
GBT	7.03E-5	6.37%
Distributed fire agent model based on ensemble clustering	2.07E-5	4.75%

Table 3. Visibility prediction results of different algorithms

Algorithms	RMSE	APE (%)
DT	1.31	4.98%
RF	0.3573	5.11%
GBT	0.99	5.52%
Distributed fire agent model based on ensemble clustering	0.2745	3.33%

Table 4. Temperature prediction results of different algorithms

Algorithms	RMSE	APE (%)
DT	25.8061	5.60%
RF	38.8284	5.91%
GBT	52.6374	5.75%
Distributed fire agent model based on ensemble clustering	24.9906	5.05%

Table 5. CO_2 prediction results of different algorithms

Algorithms	RMSE	APE (%)
DT	0.0044	7.11%
RF	0.0086	5.25%
GBT	0.0179	10.21%
Distributed fire agent model based on ensemble clustering	0.0063	4.69%

4 Conclusion

In order to solve the fire numerical simulation model FDS can accurately simulate the change rule of state parameters such as temperature, smoke concentration, visibility and so on during the fire process, but there are problems such as larger consumption of computational resources and longer simulation time. And the single-agent model has deficiencies in model complexity, calculation time and overfitting, etc. An adaptive distributed fire agent model is proposed. The experimental results show that the distributed fire agent model has advantages in prediction accuracy, adaptability and computation time, which is very suitable for the needs of modern fire numerical simulation research, and at the same time, the agent model in the paper can be used in other numerical simulation fields. However, the clustering method used in this paper is a simple division of fire data, which does not sufficiently take into account the different degree of influence of fire data characteristics, and the machine learning method is used, which can be utilized in the future to make distributed predictions using neural network models

References

1. Taylor B N .National Institute of Standards and Technology [J]. analytical chemistry, 2006.
2. Xu J , Jin Y , Du W ,et al.A Federated Data-Driven Evolutionary Algorithm[J]. 2021.
3. Strehl A , Ghosh J .Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions[J].Journal of Machine Learning Research, 2002, 3(3):583-617.
4. Verbraeken J , Wolting M , Katzy J ,et al.A Survey on Distributed Machine Learning [J].ACM Computing Surveys (CSUR), 2020.
5. Jiang C , Fan Q .State Parameter Prediction of Fire Based on Extreme Learning Machine [C]//2019 IEEE Symposium Series on Computational Intelligence (SSCI).IEEE, 2019.