Preprocessing and Analysis Method of Unplanned Event Data for Flight Attendants Based on CNN-GRU

Dongyang Li
Culture and Tourism College, Jilin Province Economic Management Cadre College, Changchun, 130012, China

***Abstract*—The data of unplanned flight attendant events has characteristics such as diversity and complexity, which pose great challenges to data preprocessing and analysis. This study proposes a preprocessing and analysis method for unplanned flight attendant event data based on Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU). Firstly, an efficient word vector tool is used to preprocess the raw data, improving its quality and consistency. Then, convolutional neural networks are taken to extract local features of the data, combined with gated loop units to capture dynamic changes in time series, thus achieving effective analysis and prediction of unplanned events in air crew. The results showed that in the 6-class task, the research model exhibited the highest accuracy at 99.24%, the lowest accuracy at 94.24%, and an average accuracy of 98.53%. The highest, lowest, and average accuracies in the 10-class task were 96.63%, 90.17%, and 93.21%, respectively. The performance of the research model was superior to support vector machine, K-nearest neighbor algorithm, and some advanced algorithms. This study provides a more accurate analysis tool for unplanned event data of flight attendants, to assist in the efficiency of aviation emergency event handling and improve aviation safety.**

***Keywords*—Convolutional neural network; gate recurrent units; air crew; unplanned events; data preprocessing; data analysis**

I. Introduction

The rapid development of the socio-economy has brought important strategic opportunities for the modern aviation transportation industry and has also put forward higher requirements for the improvement of aviation safety and security capabilities. The frequent occurrence of Unplanned Flight Attendant Events (UFAEs) and the complexity of data analysis pose a significant threat to aviation safety and passenger experience [1]. UFAEs analyses can help identify and predict events that may affect flight safety, such as mechanical failures, medical emergencies or security threats. By identifying these events in a timely manner, preventive measures can be taken to reduce the probability of accidents. In addition, by analysing unplanned events, airlines can better understand potential risks and develop effective risk management strategies to mitigate the impact of these risks, as well as reduce cost control and operational efficiency. How to effectively preprocess and analyze these event data to provide reliable prediction and decision support is an urgent problem in the current aviation management field. Traditional event analysis methods often rely on expert experience and simple statistical analysis, making it difficult to handle large-scale and diverse event data [2]. The advancement of Deep Learning (DL) technology has shown great potential for methods built on deep neural networks in processing complex data and mining deep features [3]. CNN and GRU are two widely used models in the field of DL. The former is good at extracting local features of data, while the latter has advantages in processing time series data [4-5]. However, existing DL methods still fall short in their combined ability to handle time series features and local features. Some of the methods focus too much on the long-term dependence of time series and ignore the importance of local features; or they can only capture local features and cannot effectively handle the long-term dependence of time series data. For this reason, the study proposes a method for preprocessing and analysing UFAEs data based on CNN and GRU. At First, Efficient Word Vector Tools (EWVTs) will be used to preprocess event descriptions, eliminate noise, and enhance data consistency. Then, CNN will be used to extract local features of event data, and GRU will be utilized to model the temporal dynamics of event occurrence, ultimately achieving classification and prediction of events. It aims to process UFAEs data through this method to provide support for airline unplanned event response and decision-making.

This study has six sections. Section II summarizes the current state of the industry. Section III has two sections. The first section introduces the UFAEs data preprocessing method based on EWVTs, and the second section introduces the UFAEs analysis method based on CNN-GRU. Section IV conducts performance testing on the proposed CNN-GRU. Discussion is given in Section V. Section VI is a summary of this study and prospects for future work.

II. Related Works

The essence of UFAEs data preprocessing is a text data preprocessing method. This method is a key step in processing text data analysis, involving a series of operations on the raw text data to improve the data applicability, thereby enhancing the efficiency of subsequent analysis. Hickman et al. focused on the capture of language content and style, statistical analysis ability, and effectiveness of insights obtained from text mining in the decision-making process of text preprocessing, and conducted research on computational linguistics and organizational text mining. Considering different types of text mining, research questions, and dataset features, this study provided experience- based text preprocessing decision recommendations [6]. Nova K used text preprocessing techniques such as noise removal, punctuation, and stop words to transform the original text into a Term Frequency - Inverse

Document Frequency (TF-IDF) feature matrix. This study employed three machine learning models for classification tasks, including polynomial naive Bayes, multi-layer perceptrons, and a Light Gradient Boosting Machine (LightGBM). LightGBM achieved an accuracy of 0.724 and had a higher accuracy of 0.77 when using text content for classification [7]. Thakkar et al. proposed a specific sequence of text data preprocessing steps to improve the performance of sentiment analysis, and proposed "output label matching features based on advanced techniques" to initialize the weights of Artificial Neural Networks (ANN). Simulation experiments have found that research methods have more advantages [8]. Situmeng S found that different forms of text preprocessing are helpful for successfully identifying named entities. By comparing and evaluating the three categories of people, places, and organizations, it was found that some preprocessing methods have significant effects on different entity categories. The combination of multiple preprocessing methods could significantly improve accuracy. Therefore, it was recommended to choose appropriate preprocessing methods based on the different entity categories in practical applications, rather than simply enabling or disabling preprocessing for all [9].

Meanwhile, the analysis methods for text data have been continuously optimized in recent years. Zhao C et al. used a multi-strategy text data augmentation method to handle the issues of data limitations and lack of high-quality corpora in text analysis. It compared the performance of the enhanced dataset and the original dataset: the F1 score of Long Short-Term Memory (LSTM) grounded on attention mechanism on the dataset increased by 5.0% and 4.4%, demonstrating excellent performance [10]. Sharma P and Pathak D proposed a method for analyzing social media data using a learning process, utilizing unsupervised learning and sentiment analysis to identify disaster events and their intensity. This method used annotated data to train improved fuzzy C-means clustering, using sentiment scores to identify negative emotions and determine the severity of disasters. Finally, Support Vector Machine (SVM) and ANN classifier were utilized to classify the text based on emotions. This method was effective and its accuracy continues to improve over time [11]. Sengupta S and Dave V introduced a method of legislative text analysis aimed at automatically identifying appropriate legal chapters applicable to cases. This method utilized supervised Machine Learning (ML) and natural language processing, treating the task as a multi-label classification problem. It applied traditional ML models such as logistic regression, Naive Bayes, decision trees, and SVM, and conducted hyperparameter fine-tuning analysis. Finally, SVM had the highest F1-score of 0.75 [12].

In summary, existing research on text data analysis mainly focuses on traditional statistical analysis and simple ML methods. Some studies use methods such as linear regression and decision trees for event prediction, but these methods often exhibit limitations when facing large-scale, high-dimensional, and complex event data. In addition, some studies have introduced DL techniques such as LSTM and simple CNN, but there are still shortcomings in their comprehensive ability to handle temporal characteristics and local features. In contrast, this paper constructs a data preprocessing and analysis method for UFAEs built on CNN-GRU. This method fully utilizes the Local Feature Extraction Capability (LFEC) of CNN and the TMC of GRU, which can more effectively capture the complex features and dynamic changes of event data. By introducing EWVTs for data preprocessing, the data quality and consistency have been further improved, providing more reliable inputs for subsequent DL models. The research method is not only innovative in theory, but also provides more scientific and effective decision-support tools for airlines and related management departments.

III. Methods and Materials

To provide efficient analysis tools for UFAEs, this study utilizes EWVT to preprocess event data, including data collection, cleaning, and vectorization. The CNN-GRU is adopted for in-depth analysis of preprocessed data, combining CNN's LFEC with GRU's TMC to achieve accurate classification and prediction of event data.

A. UFAEs Data Preprocessing Based on EWVT

UFAEs refer to various sudden and unexpected events encountered by flight attendants during flight operations. These events were not pre-arranged in the flight plan and may have a significant impact on flight safety, passenger service, and overall flight operations [13-14]. Common UFAEs include mechanical failures, passenger disputes, sudden weather events, and other unexpected events. The occurrence of these unplanned events has a high degree of uncertainty and suddenness, which puts extremely high demands on the adaptability and event handling ability of flight attendants. Preprocessing and analyzing these events can help improve airlines' emergency plans, enhance passenger safety, and improve service quality. The UFAEs processing flow is shown in Fig. 1.

In Fig. 1, the processing flow of UFAEs consists of six steps. Firstly, the data collection of unexpected events in the crew is carried out, followed by data preprocessing. Then, the data is vectorized, trained, and a word vector matrix is constructed. Then to conduct preliminary classification and match similar unplanned event cases. Finally, providing corresponding emergency response methods. Data collection is the step 1 in data preprocessing. When collecting data, to ensure the representativeness and comprehensiveness of the data, various unplanned events are covered as much as possible, including mechanical failures, passenger disputes, sudden illnesses, etc. Data cleaning is a key step in ensuring data quality, which requires removing duplicate records and obviously erroneous entries. The third step is to process the missing data. For cases with fewer missing values, mean or median filling methods are used; For entries with a large number of missing values that cannot be completed, they will be directly removed. After data cleaning, the integrity and reliability of the data have been preliminarily ensured. To convert textual event descriptions into numerical forms that can be processed by computers, this study uses EWVT for text preprocessing. Word2Vec, as a word vector model, performs well in semantic representation of words and can capture subtle semantic relationships between words [15]. Word2Vec is an optimization of neural network models, which includes the Continuous Bag of Words (CBOW) and Skip-gram models, as exhibited in Fig. 2.

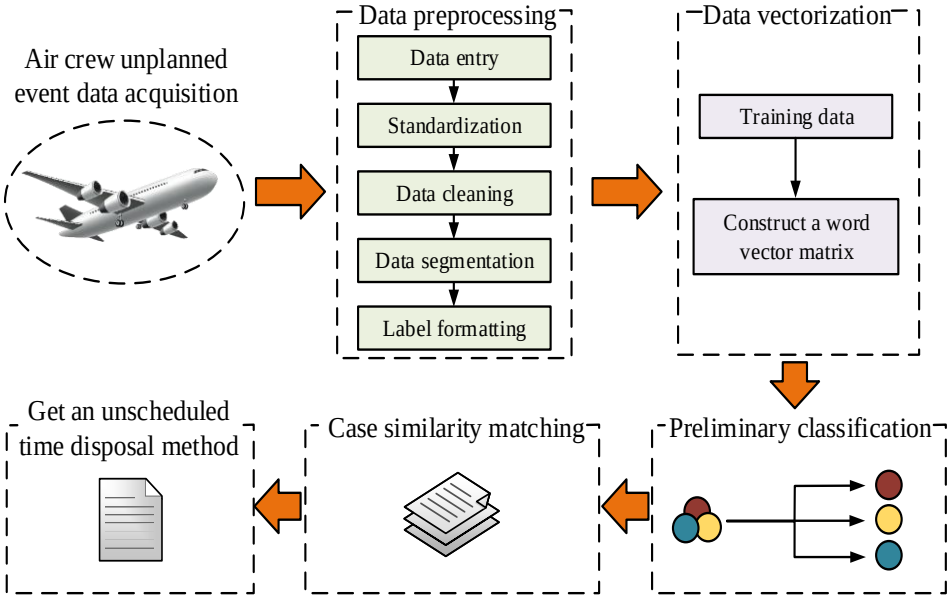


Fig. 1. UFAE processing flow.

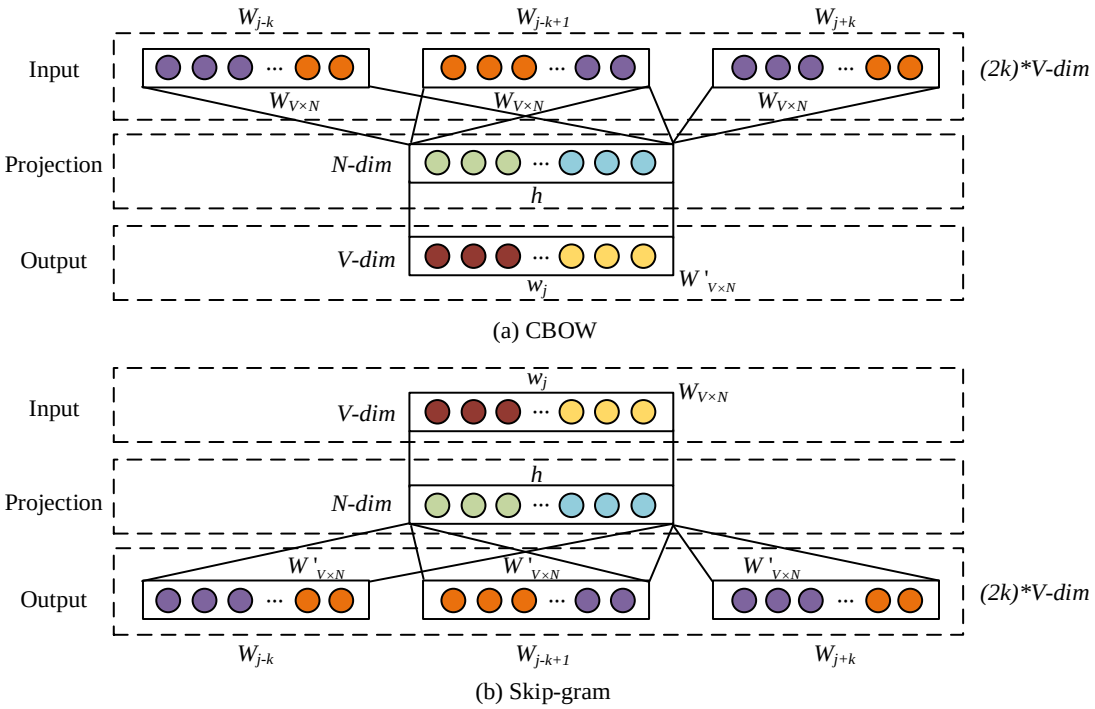


Fig. 2. CBOW and Skip-gram models.

In Fig. 2, Word2Vec consists of CBOW and Skio-gram. Fig. 2 (a) shows the structure of the CBOW, which predicts the word vector of the current word based on the word vector of the context. For the text position of target word w_j at position j, the sliding window size is designed to be k. k words above and below are used as context $con(w_j)$, with a scale of $2k$. Randomly to initialize the contextual words, then input the vector sum into the Softmax layer for normalization, and finally output the probability P of the occurrence of word w_j. The objective function of the CBOW model is Eq. (1).

$$L = \sum_{w_j \in V} \log P(w_{j-k+1}, ..., w_{j+k}) \tag{1}$$

In Eq. (1), V is the corpus where the target word w_j is located. L is the objective function. Skip-gram, in contrast to CBOW, obtains the contextual word vector from the current word vector [16]. Skip-gram uses stochastic gradient descent to optimize the objective function, and after training, the word

vector matrix W' can be obtained. This matrix is a $N \times V$ low dimensional dense word vector matrix with a vocabulary size of V. In Skip-gram, the target word w_j is input to the input layer, with the sliding window size set to k, and the word is mapped as a column vector of matrix W. The Softmax function is taken to output the $2k$ words with the highest possibility. The probability of obtaining the output word is Eq. (2).

$$P(w_{j-k}, w_{j-k+1}, ..., w_{j+k} | w_j) = \frac{e^{u_j{}^T h}}{\sum_{i=1}^{V} e^{u_i{}^T h}} \quad (2)$$

In Eq. (2), h and u_j are the row vectors of matrix W and W', and also the hidden layer vector and output vector of w_j. e is a natural constant. $u_j{}^T$ is the weight of weighted summation. The goal of Skip-gram is to maximize the logarithmic likelihood function, as shown in Eq. (3).

$$L = \frac{1}{V} \sum_{j=k}^{V=k} \log P(w_{j-k}, w_{j-k+1}, ..., w_{j+k} | w_j) \quad (3)$$

In Eq. (3), L is the maximum logarithmic likelihood function. After training the Word2Vec model, CNN is used for feature extraction. CNN can effectively extract local features from text and convert these features into fixed length vector representations as inputs for UFAEs analysis models.

B. Analysis of UFAEs Based on CNN-GRU

After completing data preprocessing, the next key step is to conduct in-depth analysis and modeling of the preprocessed data. To fully utilize the temporal and local characteristics of UFAEs data, this study proposes a method combining CNN and GRU. CNN can effectively extract local features of data, while GRU excels at handling long-term dependencies in time series data [17]. This study designs a multi-layer CNN that extracts local features from data by alternating between convolutional and pooling layers. The structure of CNN is shown in Fig. 3.

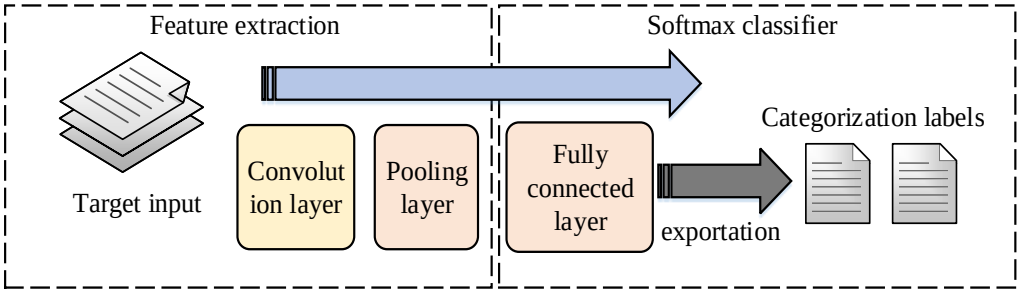


Fig. 3. CNN structure diagram.

In Fig. 3, CNN generally includes convolutional, pooling, and Fully Connected Layers (FCL), where the expression of the convolution operation continuous estimation function s is Eq. (4).

$$s(t) = \int x(a) w(t-a) da \quad (4)$$

In Eq. (4), x is the first parameter of the convolution, commonly referred to as the input. w is an effective probability dense function and also the 2nd parameter, called the Kernel Function (KF). This operation is called convolution, and the simplified expression is Eq. (5).

$$s(t) = (x * w)(t) \quad (5)$$

In CNN learning, high-dimensional data is usually input first, and the convolution kernel is the high-dimensional values generated by the algorithm. The calculation formula is Eq. (6).

$$s(i, j) = (K * I)(i, j) \sum_m \sum_n I(m, n) K(i-m, j-n) \quad (6)$$

In Eq. (6), m and n are the effective range of values for convolution. I is the input, and K is the KF of the input. For the application convenience in ML, a variant is usually utilized (Eq. (7)). Its operation is extremely semblable to convolution, but the variation is small within the valid range of m and n. This means that as m grows, the input index increases, but the kernel index decreases, achieving inter-variability of convolutions.

$$s(i, j) = (K * I)(i, j) \sum_m \sum_n I(i+m, j+n) K(m, j) \quad (7)$$

The is the core layer of CNN is the convolutional ones, which is crucial for conducting convolutional operations and enhancing CNN's feature extraction capabilities. Convolutional Layers (CL) generally refer to 2D convolution operations. If the input size is set to $D_f \times D_f$ and the convolution kernel size is $D_k \times D_k$, then the output feature size after convolution is $D_f^{'} \times D_f^{'}$. The formulas for the three are shown in Eq. (8).

$$D_f^{'} = (D_f - D_k + 2 \times pad) / stride + 1 \quad (8)$$

The CL takes Local Connections (LC) and Weight Sharing (WS) to reduce the amount of network values and decrease network complexity. LC refers to the feature extraction of CLs built on their size when moving. WS refers to the convolutional kernel not changing its parameter when extracting data features, but using the equal weight to extract features [18-19]. GRU is one of the popular variants of Recurrent Neural Network (RNN) and an improvement on LSTM, as shown in Fig. 4.

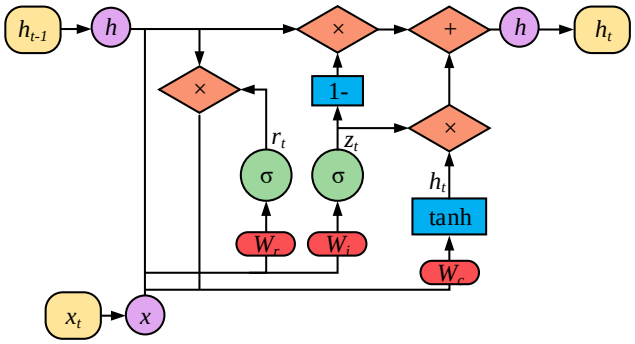


Fig. 4. GRU internal structure.

In Fig. 4, the formula for the update gate in the GRU is Eq. (9).

$$z(t) = sigmoid\left(W\_z * \left[h(t-1),\ x(t)\right] + b\_z\right) \tag{9}$$

In Eq. (9), $z(t)$ is the part that needs to be updated in the Hidden State (HS), and its value range is between 0 and 1. W_z means the Weight Matrix (WM). $h(t-1)$ denotes the HS from the previous moment. $x(t)$ is the current input state. b_z is the bias term. The calculation for resetting the door is Eq. (10).

$$r(t) = sigmoid\left(W\_r * \left[h(t-1), x(t)\right] + b\_r\right) \tag{10}$$

In Eq. (10), $r(t)$ controls how to combine the previous HS with the current input to produce Candidate HSs (CHS). W_r is the WM of the reset gate. b_r is the bias term in the reset gate. The CHS is expressed by Eq. (11).

$$h\sim(t) = tanh\left(W\_h * \left[r(t) * h(t-1), x(t)\right]\right) \tag{11}$$

In Eq. (11), $h\sim(t)$ is a CHS based on the current input and the previous HS. W_h is the WM in this state. The expression for the HS at the current moment is Eq. (12).

$$h(t) = \left(1 - z(t)\right) * h(t-1) + z(t) * h\sim(t) \tag{12}$$

In Eq. (12), $h(t)$ is the final HS at the current time. $\left(1-z(t)\right)$ and $z(t)$ are the parts that need to be discarded and retained. Through this approach, GRU networks are able to determine which information needs to be retained or forgotten in a new step [20]. This study establishes a CNN-GRU model to solve the classification and matching of text data to complete the analysis of UFAEs. The structure of CNN-GRU is Fig. 5.

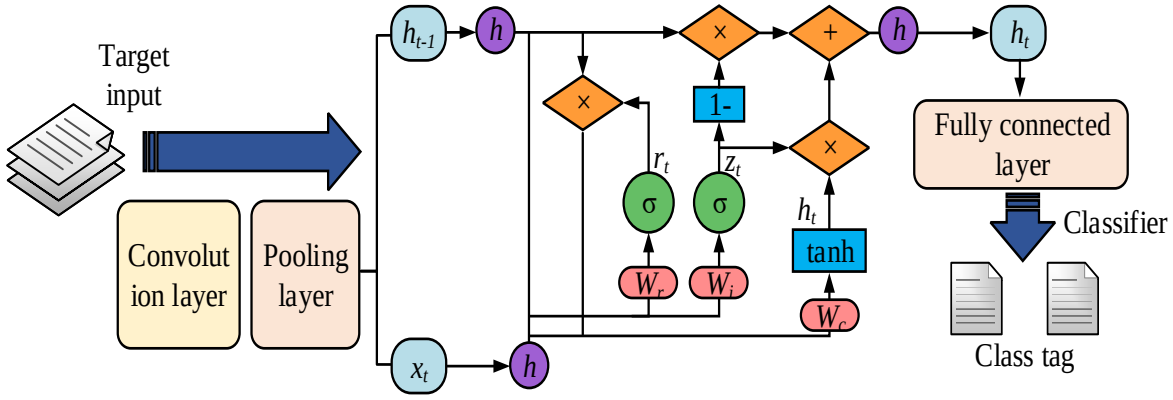


Fig. 5. CNN-GRU hybrid model structure.

In Fig. 5, CNN-GRU mainly consists of three parts, namely CNN layer, GRU layer, and FCL. This study uses the Softmax fully connected function for classification and selected Relu as the activation function. The group with the highest output probability is taken as the eventual classification result for UFAEs. Through this structure, the CNN-GRU model can effectively extract local and global features from text data, thereby increasing the classification precision and robustness of unplanned events. The training process of CNN-GRU is shown in Fig. 6.

In Fig. 6, the first is to input training data and establish a CNN-GRU model. Then, the model parameters is initialized, calculating the loss function, and updating the parameters of FCL. This process is repeated until the maximum iterations are reached, and finally the trained CNN-GRU model is obtained.

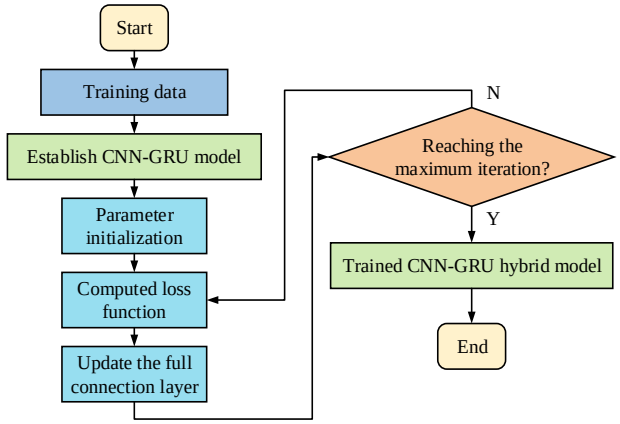


Fig. 6. Training process of CNN-GRU.

IV. RESULTS

This study conducts experiments on the 6-class and 10-class tasks of the UFAEs log sample dataset. The data is sourced from 11245 UFAEs logs related to system failures of an airline company from March 2011 to March 2021. In the fault log, some invalid data are manually removed, resulting in 10 categories and 6598 logs for classification experiments. 80% of the total data is set as training data, and 20% is set as testing data. Table I lists the experimental platform and environmental parameters.

Table II displays the fixed parameters in the constructed CNN-GRU model.

To demonstrate the superiority of the proposed CNN-GRU, traditional ML models, including SVM and K-Nearest Neighbor (KNN), as well as advanced neural network structure models like CNN-LSTM and CNN - Bidirectional GRU (CNN-BiGRU), are selected for comparison. The final FCL of all compared models is consistent, and the training batch and iteration times are selected based on the best values obtained after a large number of experiments. The test data are input into four trained comparison models and CNN-GRU. The classification task is divided into two types: 6-class and 10-class, and the obtained classification accuracy is shown in Fig. 7.

Fig. 7 (a) and (b) show a comparison of accuracy between the 6-class and 10-class tasks. In Fig. 7 (a), the CNN-GRU shows the highest accuracy at 99.24%, with the lowest at 94.24%, and an average of 98.53%. Compared to others, the average accuracies of CNN-LSTM, CNN BiGRU, SVM, and KNN are 81.24%, 74.84%, 45.55%, and 40.98%, respectively. In 7 (b), CNN-GRU also shows the highest accuracy, with the highest, lowest, and average accuracies of 96.63%, 90.17%, and 93.21%. The higher accuracy of CNN-GRU is due to its ability to solve the time series prediction problem in UFAEs analysis, accurately predict future trends and directions, and improve prediction accuracy. It continues to select F1 value as the evaluation metric. The F1 values include F1-score, Macro F1, Micro F1, and Weight F1. The F1 value represents the comprehensive classification performance. Table III exhibits the scores of different models.

TABLE I. EXPERIMENTAL PLATFORM AND ENVIRONMENTAL CONFIGURATION

Experimental environment	Disposition
Programming language	Python
Deep learning framework	Tensorflow
Operating system	Windows 10
CPU	Inter(R) Core(TM) i5-10210U
Internal memory	16G

TABLE II. FIXED PARAMETER SETTING

Argument	Set value
Activation function	ReLu
Loss function	Cross entropy
Optimization function	Adam
Word vector dimension	300
Number of convolution nuclei	128
GRU hidden layer size	256
Convolution kernel size	Three, four, five
Dropout	0.5
Batch size	64

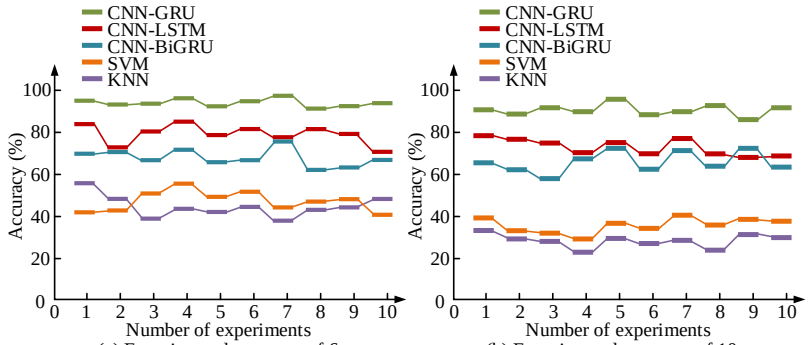


Fig. 7. Comparison of classification accuracy of different models.

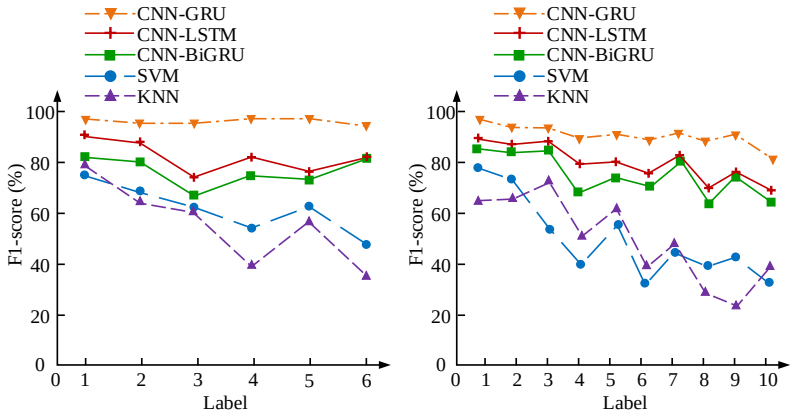
TABLE III. COMPARISON OF F1 VALUES OF DIFFERENT MODELS

Index		CNN-GRU	CNN-LSTM	CNN-BiGRU	SVM	KNN
F1-score	6-Class	98.34	89.23	87.66	72.24	70.25
	10-Class	95.25	87.12	86.82	61.48	60.99
Macro F1	6-Class	98.33	91.29	88.24	70.09	68.36
	10-Class	96.14	87.09	84.26	63.93	59.52
Micro F1	6-Class	98.92	92.42	87.72	71.82	68.93
	10-Class	96.12	88.29	81.25	63.55	60.06
Weight F1	6-Class	98.87	88.65	88.09	69.25	68.58
	10-Class	92.09	86.24	83.34	61.02	58.61

In Table III, the average F1 score of CNN-GRU is 92.09 points. The average scores of CNN-LSTM, CNN BiGRU, SVM, and KNN are 86.24, 83.34, 61.02, and 58.61. Therefore, the classification performance of different models is ranked from best to worst as CNN-GRU, CNN-LSTM, CNN-BiGRU, SVM, and KNN. CNN-GRU exhibits more stable classification performance, with higher metrics than other models, making it more suitable for UFAEs analysis. For further analysis, this study conducts repeated experiments using F1-score as the indicator to compare the F1 score of each label, as displayed in Fig. 8.

Fig. 8 (a) and 8 (b) show the F1 score of different labels in the 6-class and 10-class tasks. In 8 (a), CNN-GRU performs better and more stably on all six labels in the 6-class tasks, with an average F1-score of 98.17. Next are CNN-LSTM and CNN-BiGRU, followed by SVM and KNN. In Fig. 8 (b), CNN-GRU also performs the best in the 10-class tasks, with an average F1-score of 92.44. CNN-GRU compensates for the shortcomings of a single network and has more advantages in UFAEs analysis. Continuing to analyze the Receiver Operating Characteristic Curve (ROC) of different models, as shown in Fig. 9.

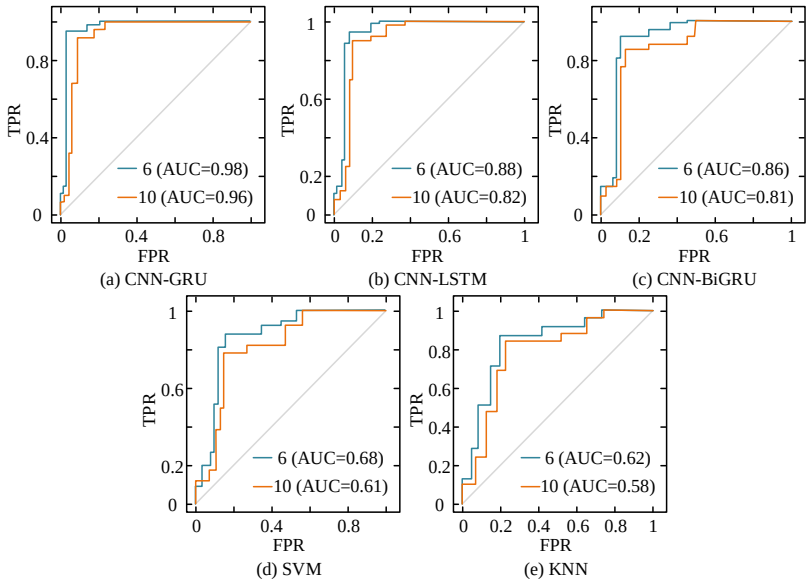
Fig. 9 (a) to 9 (e) show the ROC curves of CNN-GRU, CNN-LSTM, CNN-BiGRU, SVM, and KNN. The TPR means the True Positive Rate, while the FPR means the False Positive Rate. The Area Under Curve (AUC) under the ROC can be used to quantify the performance, and the closer it is to 1, the better the performance of the model. In Fig. 9, in the 6-class task, the AUC of CNN-GRU is 0.98, and in the 10-class task, the AUC is 0.96, which is the optimal value among all participating experimental models and has the best performance. Next is CNN-LSTM, with an AUC of 0.88 in the 6-class task and 0.82 in the 10-class task. Overall, CNN-GRU and CNN-LSTM have significantly outperformed other models. These two superior models are compared, analyzing the specific information of the models in classifying each type of label, and drawing a confusion matrix. Fig. 10 shows a comparison of six categories.



(a) Experimental accuracy of 6 classified tasks

(b) Experimental accuracy of 10 classified tasks

Fig. 8. Comparison of F1-score of different models.



(a) CNN-GRU (b) CNN-LSTM (c) CNN-BiGRU

(d) SVM (e) KNN

Fig. 9. ROC curves of different models.

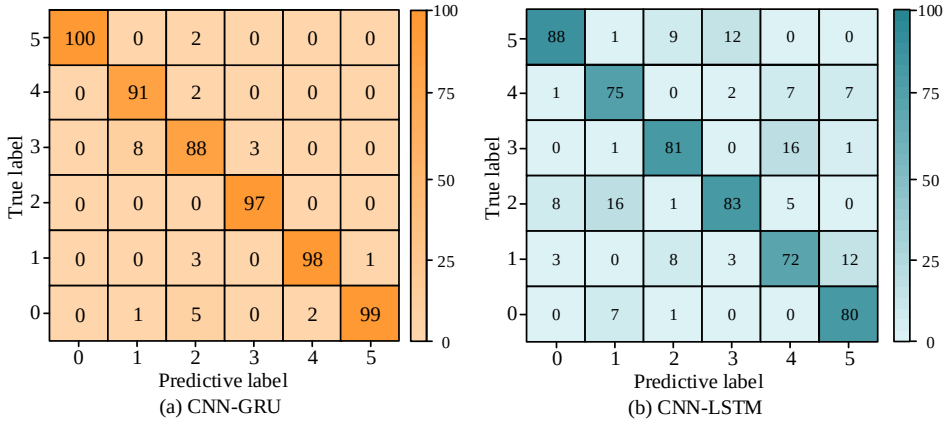


Fig. 10. Confusion matrix for 6 classification tasks.

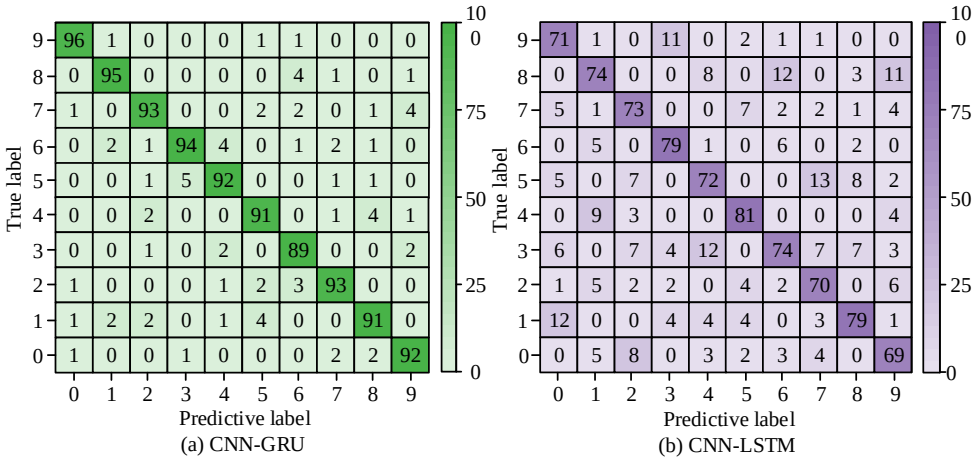


Fig. 11. Confusion matrix for 10 classification tasks.

Fig. 10 (a) and (b) show the confusion matrices of CNN-GRU and CNN-LSTM. CNN-GRU is more accurate in classifying all six labels, while CNN-LSTM clearly has more dark areas, meaning there are more misclassified labels. Moreover, the accurate classification number of each label CNN-GRU is greater than that of CNN-LSTM, indicating that CNN-GRU is more effective and robust than CNN-LSTM in UFAEs classification. Fig. 11 shows the confusion matrix for comparing 10 classification tasks.

In Fig. 11, the CNN-GRU model performs better than the CNN-LSTM in classification tasks. The number of correct classifications in CNN-GRU is higher than that in CNN-LSTM, while the number of incorrect classifications is lower. Therefore, the fusion of CNN's LFEC and GRU's TMC enables CNN-GRU to comprehensively understand and process UFAEs data, improving the accuracy of classification.

V. Discussion

The CNN-GRU model proposed in the study demonstrated strong performance in the preprocessing and analysis of UFAEs data.The CNN-GRU model was chosen for its advantages in local feature extraction and in dealing with time series dependencies. The model architecture, including the number of layers and neurons, was studied and determined based on preliminary experiments and literature recommendations of optimal configurations for similarly complex high-dimensional data [21]. While the selection of hyperparameters, such as word vector dimension 300, convolutional kernel sizes three, four, and five, and GRU hidden layer size 256, was determined through grid search methods and cross-validation to find the optimal balance of model complexity and generalisation capabilities. To prevent overfitting, a Dropout of 0.5 was used, which is a common practice in deep learning models dealing with high dimensional data. The performance of the model proposed in the study outperforms traditional models such as SVM and KNN, as confirmed by the studies of Hickman et al [6] and Zhao et al [10]. Compared to deep learning models such as CNN-LSTM and CNN-BiGRU, the CNN-GRU model performs better in terms of accuracy and F1 score.

It is worth noting that the UFAEs log data came from a single airline, which may limit the generalisability of the findings. The data collection period and the specific types of events logged may not cover the full range of unscheduled events that can occur in different aviation environments. Future work will focus on expanding the dataset to include data from multiple airlines, covering a wider range of event types and longer periods to enhance the scalability and applicability of the model in real-world environments.

VI. Conclusion

In modern air transportation, UFAEs pose a threat to aviation safety and passenger experience. To address this issue, this study proposed a UFAEs data preprocessing and analysis method based on CNN and GRU. By combining the LFEC of CNN with the TMC of GRU, complex event data could be effectively processed and accurate event classification and prediction could be achieved. This study first utilized EWVT to preprocess event description text, improving the quality and consistency of the data. Subsequently, a deep analysis was conducted on the preprocessed data using the CNN-GRU model. Experiments have shown that CNN-GRU performed well in event classification tasks, significantly outperforming traditional methods and other DL models. In the specific 6-class classification task, CNN-GRU performed better in classifying 6 labels and had stronger stability, with a mean F1-score of 98.17. The next best performers were CNN-LSTM and CNN-BiGRU, followed by SVM and KNN. Among the 10-class tasks, CNN-GRU also performed the best, with an average F1 score of 92.44, which is better than the comparison model. CNN-GRU exhibited high accuracy and robustness in processing large-scale, high-dimensional UFAEs data. This study provides airlines with scientific unplanned event response tools and technical support for improving aviation safety. In the future, the model structure can be further optimized, more advanced data preprocessing techniques can be introduced, and this method can be validated and promoted in more practical scenarios to continuously improve the level of aviation safety management.

References

[1] Maharana K, Mondal S, Nemade B. A review: Data pre-processing and data augmentation techniques. Global Transitions Proceedings, 2022, 3(1): 91-99.

[2] Ferguson-Cradler G. Narrative and computational text analysis in business and economic history. Scandinavian Economic History Review, 2023, 71(2): 103-127.

[3] Bestvater S E, Monroe B L. Sentiment is not stance: Target-aware opinion classification for political text analysis. Political Analysis, 2023, 31(2): 235-256.

[4] Sepehri A, Mirshafiee M S, Markowitz D M. PassivePy: A tool to automatically identify passive voice in big text data. Journal of Consumer Psychology, 2023, 33(4): 714-727.

[5] Werner de Vargas V, Schneider Aranda J A, dos Santos Costa R, da Sliva Pereira. Imbalanced data preprocessing techniques for machine learning: a systematic mapping study. Knowledge and Information Systems, 2023, 65(1): 31-57.

[6] Hickman L, Thapa S, Tay L, Cao M, Srinivasan P. Text preprocessing for text mining in organizational research: Review and recommendations. Organizational Research Methods, 2022, 25(1): 114-146.

[7] Nova K. Machine learning approaches for automated mental disorder classification based on social media textual data. Contemporary Issues in Behavioral and Social Sciences, 2023, 7(1): 70-83.

[8] Thakkar A, Mungra D, Agrawal A, Chaudhari. Improving the performance of sentiment analysis using enhanced preprocessing technique and Artificial Neural Network. IEEE transactions on affective computing, 2022, 13(4): 1771-1782.

[9] Situmeang S. Impact of text preprocessing on named entity recognition based on conditional random field in Indonesian text. Jurnal Mantik, 2022, 6(1): 423-430.

[10] Zhao C, Sun X, Feng R. Multi-strategy text data augmentation for enhanced aspect-based sentiment analysis in resource-limited scenarios. The Journal of Supercomputing, 2024, 80(8): 11129-11148.

[11] Sharma P, Pathak D. An Adoptive Learning Process for Social Media Text data Analysis for Disaster Management. Mathematical Statistician and Engineering Applications, 2022, 71(4): 10153-10165.

[12] Sengupta S, Dave V. Predicting applicable law sections from judicial case reports using legislative text analysis with machine learning. Journal of Computational Social Science, 2022, 5(1): 503-516.

[13] Xue H, Zhang T, Wang Q, Liu S, Chen K. Developing a unified framework for data sharing in the smart construction using text analysis. KSCE Journal of Civil Engineering, 2022, 26(11): 4359-4379.

[14] Chai C P. Comparison of text preprocessing methods. Natural Language Engineering, 2023, 29(3): 509-553.

[15] Liu J, Yu Y, Mehraliyev F, Hu S, Chen J. What affects the online ratings of restaurant consumers: a research perspective on text-mining big data analysis. International Journal of Contemporary Hospitality Management, 2022, 34(10): 3607-3633.

[16] Jagannathan M, Roy D, Delhi V S K. Application of NLP-based topic modeling to analyse unstructured text data in annual reports of construction contracting companies. CSI Transactions on ICT, 2022, 10(2): 97-106.

[17] Khurana D, Koli A, Khatter K, Singh S. Natural language processing: state of the art, current trends and challenges. Multimedia tools and applications, 2023, 82(3): 3713-3744.

[18] Dergaa I, Chamari K, Zmijewski P, Saad H B. From human writing to artificial intelligence generated text: examining the prospects and potential threats of ChatGPT in academic writing. Biology of sport, 2023, 40(2): 615-622.

[19] Hasan M D R, Ferdous J. Dominance of AI and Machine Learning Techniques in Hybrid Movie Recommendation System Applying Text-to-number Conversion and Cosine Similarity Approaches. Journal of Computer Science and Technology Studies, 2024, 6(1): 94-102.

[20] Chinthamu N, Karukuri M. Data Science and Applications. Journal of Data Science and Intelligent Systems, 2023, 1(1): 83-91.

[21] Asudani D S, Nagwani N K, Singh P. Impact of word embedding models on text analytics in deep learning environment: a review. Artificial intelligence review, 2023, 56(9): 10345-10425.