ISSN 2706-655X Vol.5, Issue 12: 88-96, DOI: 10.25236/IJFET.2023.051214

Study on Deep Learning of Aircraft Safety Enhancement and Autonomous Flight Assistance

Ruixiang Luo*,#, Xiangqi Chen#

School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, 510006, China
Corresponding author: luorx5@mail2.sysu.edu.cn
#These authors contributed equally.

Abstract: *Flight safety has always been an important concern in the aviation field. This study proposes a comprehensive scheme for aircraft assisted piloting based on deep learning. This solution uses the LSTM network for real-time aircraft status monitoring and error correction prompts, as well as aircraft autopilot assistance functions. At the same time, the database is updated through the real-time flight data processing module. The comprehensive program is designed to improve flight safety and pilot decision-making. Through automated early warning, error correction and assisted driving, real-time flight advice and control parameters are provided to improve flight safety and pilot operating capabilities.*

Keywords: *Neural Network, SEGA, LSTM, Aircraft Auxiliary Driving*

1. Introduction

Aircraft safety is an important issue in the aviation field, and real-time monitoring of the current safety status of the aircraft and providing operational error correction assistance are key components in promoting the development of aircraft modernization[1,2]. However, traditional methods have some limitations in this regard, such as relying on predefined rules and thresholds, being unable to adapt to complex environments, and processing big data, among other challenges. In order to overcome these limitations and promote the development of aircraft intelligence, artificial intelligence has become an innovative option. With the help of artificial intelligence methods, technologies such as machine learning and deep learning can be used to learn and extract patterns from large amounts of flight data to achieve more accurate aircraft status monitoring and operational error correction assistance[3-5].

2. Data preprocessing

2.1 Clustering feature generation using K-means

The data are obtained from the files 201404080617 and 201404100843 attached to the open source website (http://www.mathorcup.org/), and the flight parameters of the aircraft are extracted. In order to simulate the system, the article need simulation history data and real-time simulation test data. Attachment 1: file 201404080617 is used as the historical data of the simulation, at the same time, in order to obtain the safety evaluation criteria of the aircraft. K-means clustering is used to classify the data into 0 and 1, and a new safety label is added. 0 indicates that the aircraft state is safe and 1 indicates that the aircraft state is dangerous. For the simulation test data, the article choose Appendix 1: 201404100843.

2.2 Data handling and balance

It can be seen from the field description of the key parameter segment data in the dataset that some data are recorded multiple times in 1 second. In order to prevent data redundancy and the subsequent unexpected explosion of model data dimensions, the data that meet the above conditions are averaged, and the original measured data is deleted and replaced by the average.

At the same time, because the number of safety category samples and non-security category samples in the sample is not smooth, the article use the down-sampling method to balance the samples. The Cluster-Centroids method is used to reduce the samples of the security category and balance the samples for subsequent training of the security status classifier.

ISSN 2706-655X Vol.5, Issue 12: 88-96, DOI: 10.25236/IJFET.2023.051214

The general model of artificial neural network consists of four basic elements, which are:

(1) The BP neural network is linked by different node coefficients. When connecting weights and weights are positive, it indicates that the current link is an exciting state. Conversely, if the link coefficient is negative, the link state is a state of suppression.

(2) The input signal and the linear signal are the combination of the signals for each input signal.

(3) The function of the nonlinear activation function: making the neuron output signal within a certain range.

$$u_k = \sum_{j=1}^{p} w_{kj} x_j \tag{1}$$

$$v_k = net_k = u_k - \theta_k \tag{2}$$

$$y_k = f(v_k) \tag{3}$$

BP neural network is back propagating, mainly composed of three parts: input layer, middle layer and output layer. The number of nodes in the input and output layers is relatively easy to determine, but the determination of the number of nodes in the hidden layer is a very important and complex problem.

2.3 The determination of the number of network layers

BP neural network is back propagating, mainly composed of three parts: input layer, middle layer and output layer. The number of nodes in the input and output layers is relatively easy to determine, but the determination of the number of nodes in the hidden layer is a very important and complex problem.

3. Model building and solving

3.1 BP neural network aircraft safety state predictor optimized based on PSO algorithm

3.1.1 Introduction

The prediction of aircraft safety state is crucial to ensure optimal performance and ensure passenger safety. Backpropagation (BP) neural networks have shown promise in modeling complex systems. However, the efficiency of BP networks often depends on appropriate parameter initialization. In this section, the article introduce an optimization strategy based on the Particle Swarm Optimization (PSO) algorithm to enhance the performance of the BP neural network for predicting aircraft safety state.

3.1.2 BP Neural Network Overview

The BP neural network consists of an input layer with 7 neurons, two hidden layers with 4 neurons each, and an output layer with 1 neuron. The input layer corresponds to the 7 features of the input data, while the output layer is used for binary classification. And the frame diagram is shown in Figure 1.

The activation function used in the hidden layers is ReLU, and the output layer uses a sigmoid activation function. The model is compiled with the Adam optimizer and binary cross-entropy loss function, and accuracy is used as the evaluation metric.

The model is trained for 300 epochs with a batch size of 32, and early stopping is applied to prevent overfitting.

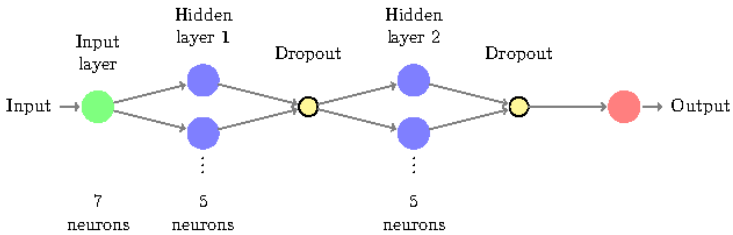

Figure 1: BP Neural Network frame diagram

3.1.3 Particle Swarm Optimization (PSO) Overview

PSO is a heuristic optimization method inspired by the social behavior of birds. It optimizes a problem by iteratively refining a population of individual solutions.

The article represents the properties of the particles as Table 1.

Table 1: Presentation of all the properties

Properties	Presentation
Current position	x_i
Current velocity	v_i
Best position found by the particle so far	$pbest_i$
Best position found by any particle in the swarm	$gbest$

At each iteration, the velocity and position of particle i are updated using the following equations:

$$v_i(t+1) = \omega \times v_i(t) + c_1 \times r_1 \times \left(pbest_i - x_i(t)\right) + c_2 \times r_2 \times \left(gbest - x_i(t)\right) \quad (4)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

In the above formula, ω is the inertia weight, which determines the tendency of a particle to continue in its current direction; c_1 and c_2 are learning factors, typically, c_1 is seen as the cognitive component, and c_2 as the social component; r_1 and r_2 are random numbers in the interval [0,1].

In the process of optimizing the BP neural network with PSO, each particle represents a set of weights and biases for the neural network. Thus, x_i is essentially a vector of all the weights and biases of the neural network. The aim of PSO here is to minimize some fitness function.

3.1.4 Threshold tuning based on SEGA algorithm

SEGA (Strengthened Elite-preserving Genetic Algorithm) is an improved form of the traditional genetic algorithm (GA). In standard genetic algorithms, there's a common strategy known as the elite strategy. Its purpose is to ensure that the best individual(s) in each generation is preserved, preventing the loss of the best solutions due to crossover and mutation operations.

First, selection process selects individuals from the current population based on their fitness. It can be represented using tournament or roulette wheel selection. If using the roulette wheel, the probability $P(i)$ of selecting an individual i would be:

$$P(i) = \frac{fitness(i)}{\sum_j fitness(j)} \quad (6)$$

Where $fitness(i)$ is the fitness of individual i, and the denominator is the sum of the fitnesses of all individuals in the population.

Then, Pairs of individuals are chosen (based on the selection process) to undergo crossover to produce offspring. Given two parents A and B, a simple one-point crossover might yield offspring A' and B'.

With a probability P_{mut}, an individual's genes might undergo mutation. If x is a gene in an individual's chromosome, then:

$$\begin{cases} flip(x) & if\ rand < P_{mut} \\ 0 & otherwise \end{cases} \quad (7)$$

Where $flip(x)$ is a function that changes the gene x.

Therefore, this paper tries to use heuristic algorithms to obtain the final classification value of the model to obtain the optimal segmentation strategy. Considering that the final output of the model is a pixel probability graph of two categories, it is important to take appropriate entry values for pixel-level classification to evaluate the final classification performance of the model.

On the basis of the model trained by using the training set, the $y\_pred\_foreground$ obtained by entering the validation set y_pred two category probability plots.

The Genetic Algorithm for Enhanced Elite Retention (SEGA) is used to determine the optimal classification threshold, and the algorithm design is as Figure 2.

```
Algorithm 1: Genetic Algorithm for Threshold Optimization
  Input: Predictions and true labels
  Output: Best threshold and evaluation results on test set
  Define evaluation metrics;
  Define objective function;
  Define fitness function;
  Define decode function;
  Initialize genetic algorithm parameters;
  Initialize population;
  for generation in num_generations do
      Update crossover rate and mutation rate;
      Evaluate fitness of current population;
      Select elites;
      Select individuals;
      Crossover;
      Mutate;
      Add elites back to population;
  end
  Find best individual;
  Test and evaluate model using best threshold;
  Plot confusion matrix and ROC curve;
```

Figure 2: Genetic Algorithm for Enhanced Elite Retention

First, the article randomly initialize populations of individuals, each representing a threshold of ζ, and encoding in binary. Then, Statistical analysis of the current population was performed to record the optimal individual and fitness best optimal fitness. According to the fitness of each individual, the best E individual is selected as the elite individual, where E is the predetermined number of elite individuals, except for the elite individual, the parent is selected from the current population by roulette to generate offspring in a multi-point cross way, and the offspring are randomly mutated according to the set mutation rate. Then, Combine elite individuals with offspring from crossovers and mutations to generate new populations. If a predetermined algebra is reached or a solution that satisfies the conditions is found, the algorithm is terminated. Otherwise, go back to the second step.

In the end, the article found the optimal threshold of 0.58276.

3.1.5 Complete model establishment

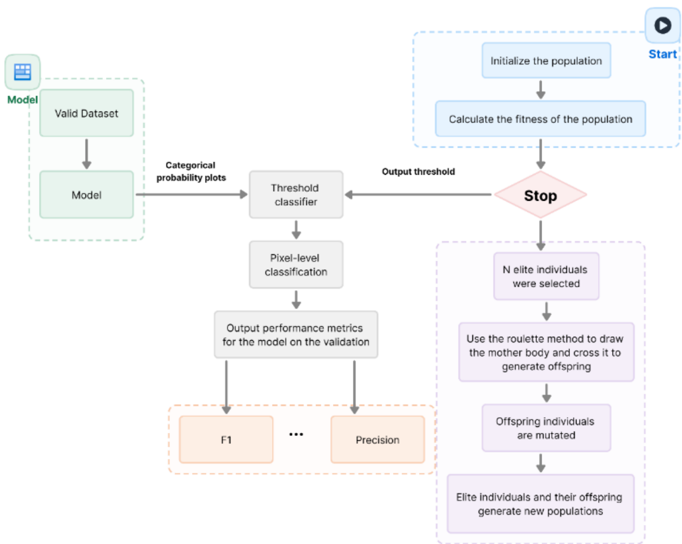

Figure 3: Flowchart of the Aircraft Safety Status Predictor

In this study, the article adopted an approach that combines model validation with SEGA algorithms, aiming to enhance the predictive accuracy of the model and optimize its parameters which process is like Figure 3. The following is a detailed description of this method.

Initially, our model takes a validation dataset as input. After processing by the model, it outputs a series of classification probability plots, displaying the model's prediction probabilities for each category.

ISSN 2706-655X Vol.5, Issue 12: 88-96, DOI: 10.25236/IJFET.2023.051214

To determine which probability values can be considered as positive predictions, the article introduced a threshold classifier. The primary task of this classifier is to decide, based on a predefined threshold, under which category a pixel point should be classified.

Considering that the model classification performance indicators include F1 score and so on, the variables are defined as ζ. And for each performance index, the formula is as follows, all of which are functions about variable ζ:

$$P = \frac{TP}{TP+FP} \tag{8}$$

$$REC = \frac{TP}{TP+FN} \tag{9}$$

$$F1 = G(\zeta) = \frac{2\times P\times REC}{P+REC} \tag{10}$$

The weight allocation method is used for each performance index, because the IOU and Dice coefficient indicators are more important in this problem, the weight adjustment of the two indicators is larger, and the final optimization function is defined as follows:

$$Fitness = F(\zeta) = 0.4P + 0.4REC + 0.2F1 \tag{11}$$

After completing pixel-level classification, the article calculate the model's performance metrics on the dataset, such as Intersection over Union (IOU), by comparing it with the true values of the validation dataset.

Simultaneously, to further optimize the model's parameters and structure, the article incorporated a SEGA algorithm. This algorithm starts with the initialization of a population, where each member represents a set of potential model parameters. Next, the algorithm evaluates the fitness of the population members. This is achieved by applying each member (or set of model parameters) to the problem and assessing its performance.

Once the evaluation is completed, the article select N elite individuals with the highest fitness. Then, using the roulette wheel selection method, pairs are chosen from the population and offspring are generated through crossover. To increase the diversity of the population, these offspring undergo a mutation process. Finally, by combining the elite individuals and the newly generated offspring, a new population is formed, laying the foundation for subsequent fitness evaluations and optimizations.

In summary, our approach merges the precision of model validation with the continuous optimization offered by genetic algorithms, aiming to provide the best model parameters and structure for a specific problem.

3.2 An aircraft assisted driving model based on LSTM

3.2.1 Model Overview

To enhance the accuracy and safety of airplane navigation, the article have adopted an assistant driving model based on LSTM shown by Figure 4. LSTM is a special type of Recurrent Neural Network (RNN) particularly suitable for processing and predicting time-series data, such as flight data of an airplane.

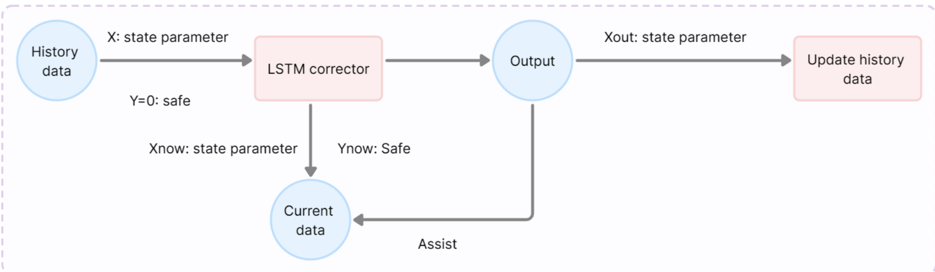

Figure 4: Aircraft assisted driving model frame chart

3.2.2 Data Preprocessing

Initially, the article divide the data into training and test sets. To neutralize the influence of different units of measurement, the article utilize MinMaxScaler for feature normalization, scaling the data to the [-1, 1] range.

ISSN 2706-655X Vol.5, Issue 12: 88-96, DOI: 10.25236/IJFET.2023.051214

3.2.3 LSTM Corrector

An LSTM corrector is utilized to process and correct the state parameters of the airplane. This corrector takes input from both "History Data" and "Current Data" and outputs a corrected state parameter, Xout.

Subsequently, the article construct an LSTM model featuring 64 neurons in the hidden layer and four layers deep. The article also introduce a dropout layer with a 0.4 dropout rate to mitigate the risk of overfitting and employ the Adam optimizer for parameter optimization, coupled with a learning rate scheduler to enhance convergence performance(Table 2).

Table 2: The settings of LSTM parameter

Parameter	Value Setting
Input_size	21
Hidden_layer_size	64
Output_size	21
Num_layers	4
Dropout rate	0.4

3.2.4 Assistance Decision

Once the LSTM corrector generates the output, Xout, it is compared with the current data to decide the next move of the airplane. If Xout indicates safety, the system produces a "safe" signal and proceeds. Otherwise, an alert is raised, and necessary corrective measures are taken.

3.2.5 Data Update

To ensure the LSTM corrector continuously receives the latest and most relevant flight data, the system adds Xout to the "History Data," allowing the model to self-learn and adjust based on the most recent flight data.

3.2.6 Solution and Optimization

During the training phase, the article create input-output sequences from each time window's data and the subsequent value at the end of that window. Throughout 150 epochs, the model iteratively learns by minimizing the Mean Squared Error loss function to adjust the network weights. The article also record the training and validation losses to monitor the model's learning progress.

Ultimately, the article use matplotlib to plot the loss curves throughout the training and validation processes, facilitating a visual assessment of the model's learning efficiency and potential overfitting. The following figure displays the loss variation during model training, providing insights into the model's performance on both the training and validation sets.

3.3 Automatic Flight Aid model

In summary, the article establish the automatic flight assistance model shown in Figure 5 by using the BP neural network aircraft safety state predictor optimized based on the PSO algorithm and the aircraft assisted piloting model based on LSTM.

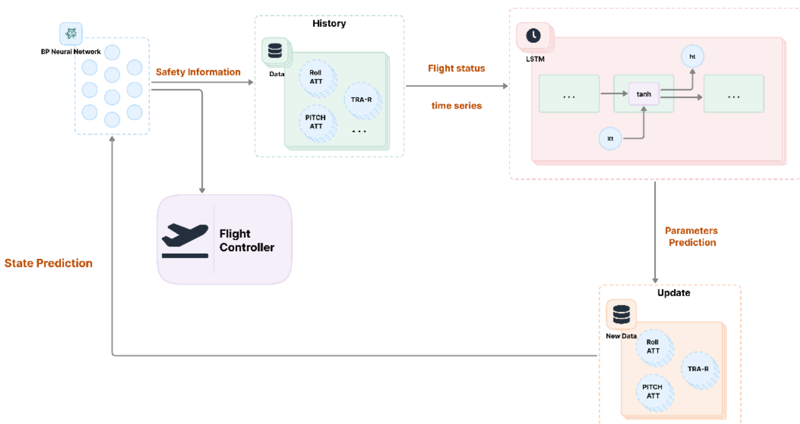

Figure 5: Automatic flight aid frame chart

ISSN 2706-655X Vol.5, Issue 12: 88-96, DOI: 10.25236/IJFET.2023.051214

The article use the BP neural network to judge the safety status of the aircraft from the current flight parameters, and store the data in historical data. Then, the historical data is put into the prediction of the flight parameters of the aircraft at the next point in time of the LSTM model, and the predicted data is re-invested into the BP network to judge the safety state.

4. Results

4.1 The performance of BP neural network

4.1.1 Confusion Matrix Metrics

After using the SEGA algorithm to calculate the optimal threshold of 0.58, the article output the thermodynamic diagram of the test set of the BP model at the optimal threshold of 0.58, and the ROC curve is shown below.

Figure 6 shows the confusion matrix of the result that is made by BP network. Based on the confusion matrix image, the article can deduce the following information:

First, the number of cases where the actual class was 1 and the predicted class was 1 is 265. This means that there were 265 instances where the safety state was correctly predicted as safe.

Then, the number of cases where the actual class was 0 and the predicted class was 0 is 280. This means that there were 280 instances where the non-safe state was correctly predicted as non-safe.

Third, the number of cases where the actual class was 0 but predicted as 1 is 3. This means that there were 3 instances where non-safe states were incorrectly predicted as safe.

At last, the number of cases where the actual class was 1 but predicted as 0 is 20. This means that there were 20 instances where safety states were incorrectly predicted as non-safe.

According to this confusion matrix, the article can say that the model performs relatively well on the test set, as the number of true positives and true negatives is substantially higher than the number of false positives and false negatives.

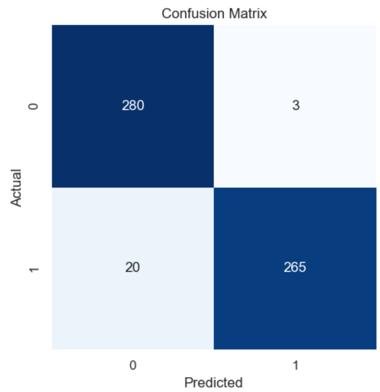

Figure 6: Confusion Matrix of BP network

4.1.2 ROC Metrics

Figure 7 shows the ROC curve produced by BP network training. ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It curve plots the True Positive Rate against the False Positive Rate at various threshold settings. The area under the ROC curve is a measure of the model's ability to discriminate between the two classes. An AUC of 1.00 is perfect, indicating that the model has a perfect measure of separability and is able to distinguish between the positive and negative classes without error.

The ROC curve that the image in the first image hugs the top-left corner, suggesting an excellent level of discrimination.

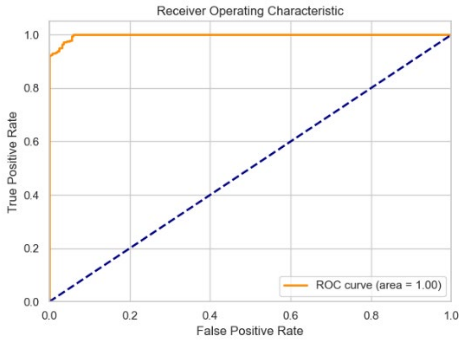

Figure 7: ROC Curve from BP network

4.1.3 Fitness Metrics

Figure 8 appears to show the optimization process over a number of generations, which is common in evolutionary algorithms like Genetic Algorithms or Particle Swarm Optimization.

The "Best Fitness" line indicates the fitness score of the best individual in each generation, while the "Average Fitness" line represents the average fitness score of the population at each generation.

The graph indicates that the fitness quickly improved in the initial generations, then both the best and average fitness appear to plateau. This suggests that the optimization process is converging to a solution.

By the end of the displayed generations, the best and average fitness values are close, which may indicate a lack of diversity in the population.

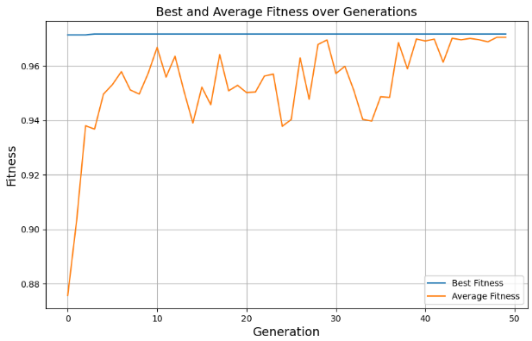

Figure 8: Best and Average fitness over generations chart

From these images, the article can conclude that the model being optimized has achieved an excellent discriminative performance according to the ROC curve, and the optimization process via the PSO algorithm has quickly converged to a high fitness solution, which remains stable over subsequent generations.

4.2 The performance of LSTM

From Figure 9, the article can see that both training and validation losses significantly decrease as the number of epochs increases, indicating that the model is indeed learning the features of the data throughout the training process.

Around 20-30 epochs, the loss curves begin to plateau, suggesting that the model may be starting to converge. The further reduction in loss value becomes slower, indicating limited gains in model learning at this stage. Throughout the training process, the training loss and validation loss remain close and decrease without the validation loss increasing while the training loss continues to decrease, indicating no significant overfitting is occurring.

At last, there is no significant gap between training and validation losses visible in the chart, meaning the model has likely converged to an appropriate state without overfitting the training data.

ISSN 2706-655X Vol.5, Issue 12: 88-96, DOI: 10.25236/IJFET.2023.051214

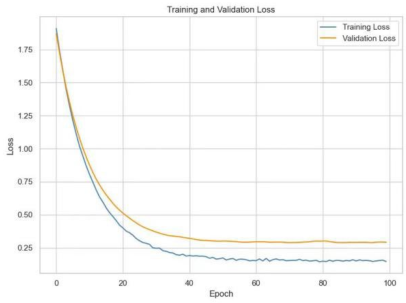

Figure 9: Training and validation loss of LSTM

Figure 10 sees that the predicted MAGNETIC HEADING, produced by LSTM regression follows the actual MAGNETIC HEADING quite accurately. This suggests the model is capable of capturing the temporal characteristics of the flight data. In most points, the predicted values closely follow the actual values, demonstrating a good fit.

At points of large fluctuations or trend changes, the model is able to respond to changes in the actual data, but there are occasional lags or overshoots, which may be due to the intrinsic characteristics of LSTM in time-series prediction.

Although the model performs well, there is room for improvement in prediction accuracy at certain points. Further parameter tuning, more complex network structures, or additional training data might be needed to enhance the model's generalization and accuracy.

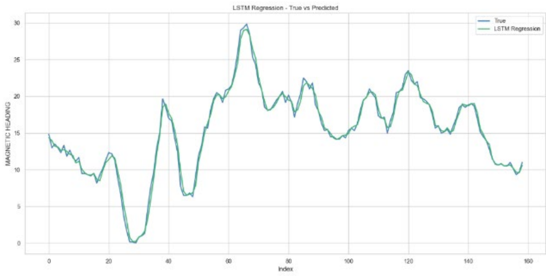

Figure 10: The prediction of MAGNETIC HEADING from LSTM

5. Conclusions

The trend of mass data in power system provides a basis for load characteristic analysis and prediction model establishment, but the classical load forecasting method can not afford such a huge time and computing resource consumption. The problem of over fitting in large sample set will affect the prediction accuracy. In this paper, a power load forecasting model is built by using the BP neural network model, making full use of the powerful data processing function of Clementine and preventing the over fitting function. The experimental results show that the BP neural network model has good predictability and robustness, and has a certain practical application value.

References

[1] Cheng Qiyun, Sun Caixin, Zhang Xiaoxing, et al. Short-Term load forecasting model and method for power system based on complementation of neural network and fuzzy logic [J]. Transactions of China Electrotechnical Society, 2004, 19(10): 53-58.
[2] Qixi F U , Xiaolong L , Jiaqiang Z ,et al. Design and Implementation of Autonomous Flight Unmanned Aircraft System Geo-Fence Algorithm[J]. Journal of Xi'an Jiaotong University, 2019.
[3] Amjady N. Short-term hourly load forecasting using time series modeling with peak load estimation capability [J]. IEEE Transactions on Power Systems, 2001, 16(4): 798-805.
[4] Ma Kunlong. Short term distributed load forecasting method based on big data [D]. Changsha: Hunan University, 2014.
[5] SHI Biao, LI Yu Xia, YU Xhua, YAN Wang. Short-term load forecasting based on modified particle swarm optimizer and fuzzy neural network model [J]. Systems Engineering-Theory and Practice, 2010, 30(1): 158-160.