A Neural Network Approach to Predict Indian Railways Ticket Confirmation

Satyajit Gantayat
3rd-year Undergraduate student
Electrical & Electronics Engineering Department
National Institute of Technology, Karnataka-575025
Email: 16ee253.satyajit@nitk.edu.in, Mob: +91-7008179687

Abstract— **The Indian Railways supports around tens of millions of people each day and yet millions don't avail confirmed tickets. In this project, I have trained an Artificial Neural Network using the dataset obtained from a reliable source (Github) which has been scrapped from an application. I have used a simple feedforward backpropagation neural network to achieve the goal. From our experiments, we conclude that it is possible to learn complicated features that govern the confirmation of rail tickets. The learned model can then be used to predict if (and when) a wait-listed ticket will get confirmed**.

*Keywords**: Artificial Neural network; Backpropagation Algorithm; IRCTC ticket data; MATLAB***

I. INTRODUCTION

The Indian Railways is the 8th largest employer in the world. It caters to around tens of millions of travellers each day. Yet a large fraction of the travellers struggle a lot to get confirmed ticket. While some travel illegally with a wait-listed ticket, others cancel ticket at very last moment. The allocated tickets get sold in a very short duration. The situation is a lot more worsened during festive seasons. So, reserving a wait-listed ticket is risky without knowing its probability of getting confirmed.

In this experiment, it has been tried to provide a platform which can predict the confirmation of the wait-listed tickets efficiently. Empowered with this knowledge, users could be in a better position to make further decisions- either wait for the ticket to get confirmed or to go for alternative travel arrangements.

A feedforward neural network has been trained to accomplish the goal. The GUI features in MATLAB neural networks toolbox [1] help a lot in visualizing and analyzing the problem. Different activation functions like logsig, tansig and ReLU have been used and their performances have been compared. The best one has been chosen to train the model.

The contribution to the proposed paper is summarized to the following items: (1) Indian Railway wait-listed ticket confirmation prediction, (2) investigation of the features influence to the prediction technique, (3) performance analysis of the state-of-the-art ANN model designed for the aforesaid purpose.

The rest of the paper is organized as follows: Section II presents some deep insight into the dataset to be used to train the model. Section III discusses some basics of ANN and how to design an ANN for the problem. Section IV describes the results obtained and draws some conclusion on the performance of the trained network. Finally, section V gives some insight into all possible future works along with some conclusions.

II. DATASET VISUALIZATION

The dataset required for this study was obtained from a reliable Github repository, which was a dataset scrapped from some applications like Indian Rail (Fig. 1). This dataset contains 5024 data points for a single route i.e. from NDLS to PNBE.

	A	B	C	D	E	F	G	H	I
1	travel Class	travel date	travel month	booking status	status 1month	status 1week	status 2day	status 1day	confirmation
2	4	21	3	232	98	39	5	0	1
3	3	22	12	158	97	-1	-1	-1	0
4	4	11	6	157	95	-1	-1	40	0
5	3	19	5	148	94	48	-1	-1	0
6	4	4	7	196	92	-1	-1	-1	0
7	4	4	7	196	92	-1	-1	-1	0
8	2	30	5	11	9	6	6	6	0
9	2	15	2	23	9	7	7	-1	0
10	3	23	3	37	9	-1	-1	-1	0
11	4	31	3	134	89	-1	-1	-1	0
12	4	5	5	150	88	62	46	11	0
13	4	1	7	153	85	56	51	35	0
14	4	17	4	177	83	78	61	27	0
15	4	11	5	168	83	-1	12	0	1
16	3	29	1	133	83	-1	-1	-1	0
17	4	3	6	144	83	-1	-1	-1	0
18	3	18	5	126	82	-1	-1	-1	0
19	3	16	5	126	80	30	-1	-1	0
20	2	12	4	13	8	-1	7	-1	0
21	3	12	6	12	8	-1	-1	-1	0
22	4	16	3	194	79	76	31	28	0

Fig. 1 Dataset Used in the problem

This dataset contains 8 input features which have been listed below

- Feature 1: Travel Class, e.g. SL, 1A, 2A, 3A
- Feature 2: Travel date
- Feature 3: Travel month
- Feature 4: Booking Status
- Feature 5: Status before 1 month of journey

- Feature 6: Status before 1 week of the journey
- Feature 7: Status before 2 days of the journey
- Feature 8: Status before 1 day of the journey

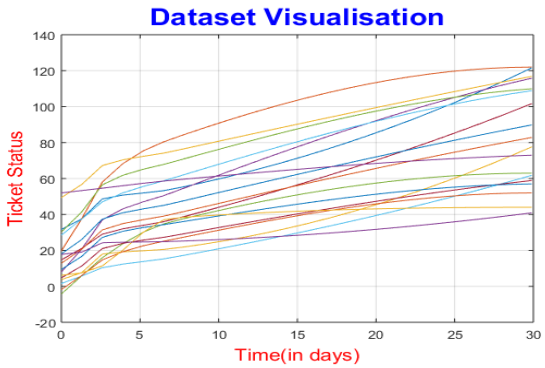

Fig. 2 Plot of booking status of 17 different datapoints w.r.t days

The last four features show the trend of cancellation as the journey date is approached gradually [3]. A plot of ticket status over the defined periods reveals a linear trend of cancellation from day 30 to day 5 and a sigmoidal trend from day 5 through day 0 (e.g. Fig. 1).

The dataset used here has 5240 instances in total which contains 8 input features in total. The target value is either 0 or 1. The target value 1 represents that the ticket confirmed ticket and 0 represents unconfirmed tickets. The missing data points have been replaced with -1.

The total dataset is divided into Training data, Validation data and test data randomly in the ratio of 90:5:5.

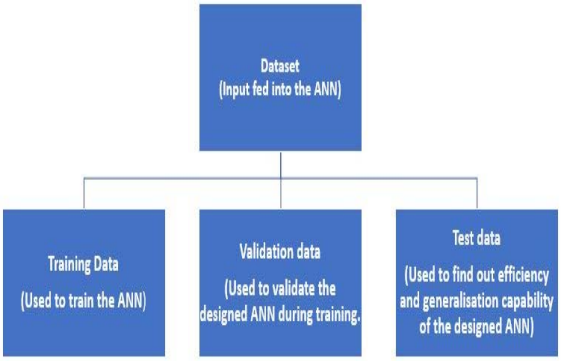

Fig. 3 dataset division in MATLAB

III. An ANN approach to the problem

A. Artificial Neural Network

An Artificial Neural Network (ANN) is an information processing paradigm [2] that is inspired by the biological nervous system such as the brain, processing the information. It consists of interconnected powerful computing units called neurons. The basic structure of an artificial neural network and its structural unit has been shown in Fig 4.

As shown in the figure, a typical ANN consists of 3 types of layers, i.e. the input layer, hidden layer, output layer. These layers consist of some neuron units which are computational units having an activation function unit which is connected to inputs through some weighted links and biases.

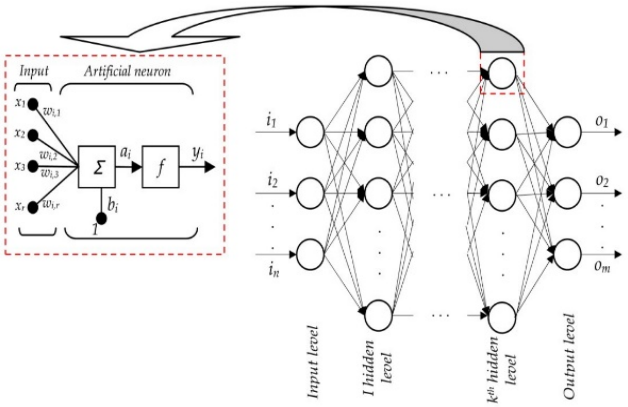

Fig. 4 Structure of an ANN and its structural unit i.e. Neuron

There are a lot of ANN structures which are used for different purposes [4]. The problem statement decides the type of the behavior and the dataset behavior decides the training algorithm needed to train the designed ANN. ANNs are very powerful tools which can be used to solve very complicated problems. Some specific features like fault-tolerance, the capability of handling very large datasets, great accuracy make them really powerful and help them get the edge over other available machine learning algorithms.

The problem statement defined here is a simple binary classification problem. The aim is to forecast the ticket confirmation probability. so, a simple feedforward ANN is the best suited for solving the problem. The approach has been discussed below.

B. Problem-solving approach

The complete approach to the problem solving can be studied using three different aspects as described below.

1) Pre-processing of data:

After collection of the raw dataset, the raw data has to be pre-processed to increase the efficiency of the ANN to be used for training. All the input features are modified so that they all lie in a range of [-1:1] having mean value zero. Now the data is cleaned and fed to the ANN designed. The pre-processing is done using the following formula.

$$x=\frac{x-\mu}{\sigma}$$

where μ=mean of the data

σ=standard deviation of the data

2) Design of ANN:

After cleaning the data, we have to design an ANN for our problem. As mentioned above, due to the binary classification problem, ‘feedforward network’ is the best suited. It doesn’t have any feedback network. The input has layer will have 8 nodes as it has 8 input features and the output layer will have only one node. Two hidden layers have been taken using the rule of thumb with 20 nodes in each of them. For binary classification, ‘sigmoid’ activation has been chosen. However, the effect of different activation functions has been studied in the later part. The network structure has been shown in Fig. 5.

The network has been created using MATLAB 'nntool' which provides a GUI feature to analyze the network design.

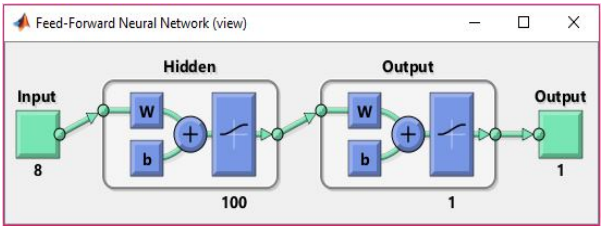

Fig 5. ANN structure for the problem

After designing the network, the next task is to decide the required parameters to train it. The problem being a forecast problem, we can use 'Backpropagation algorithm' [5] to accomplish the goal which is represented by 'trainlm' command in MATLAB. The algorithm compares the output of the network and compares with the target value. Using the error calculated, it again updates the weights of the network and continues until a minimum satisfactory error is obtained.

The training of the designed network took approximately 20 seconds to complete 34 iterations (Fig. 6). The learning rate was found to be 0.0001 with a final gradient value of 0.0107.

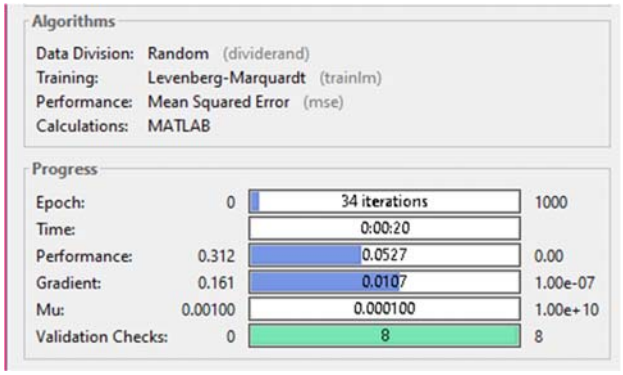

Fig 6. Results obtained after training of the ANN

3) *Performance Analysis:*

After training the designed ANN, its performance must be analyzed for validation of the network.

From the performance graph of the trained network in Fig. 6, it can be noticed that the test data error is very close to that of training data error. It suggests that the network is well regularised and hasn't overfitted the training data. It's capable of generalizing well for the test data, which the network has not encountered before.

From the graph, it's obvious that the efficiency of the network of predicting new test data is around 80%. The efficiency is still low due to the limited availability of data. To increase its efficiency a larger dataset is needed having approx. 50000 instances.

However, the performance of the designed network is quite satisfactory and it can be used to predict a wait-listed train ticket. It can be further improvised using some certain techniques which have been discussed in the next section.

To get more satisfactory results, the network has been modified with different number of hidden neurons and different number of input features. The analysis result has been shown below.

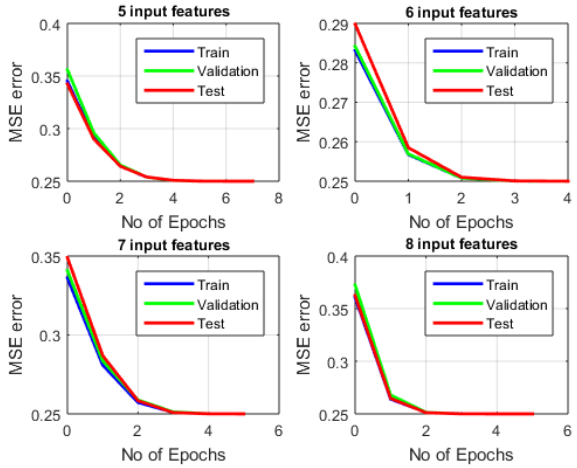

Fig 7. ANN with 20 hidden neuron

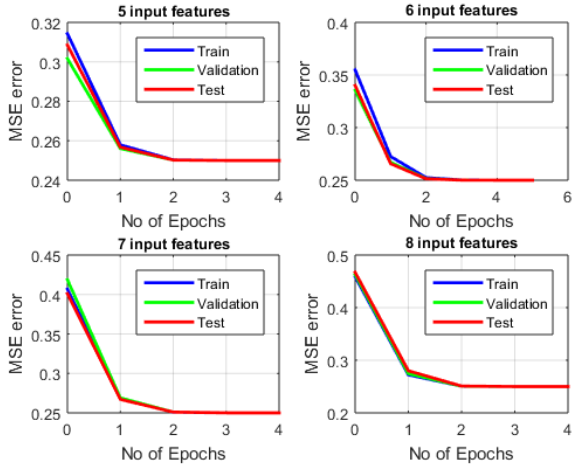

Fig 8. ANN with 40 hidden neurons

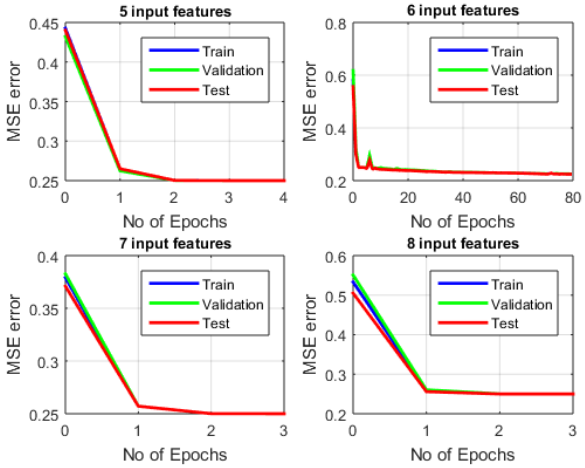

Fig 9. ANN with 60 hidden neurons

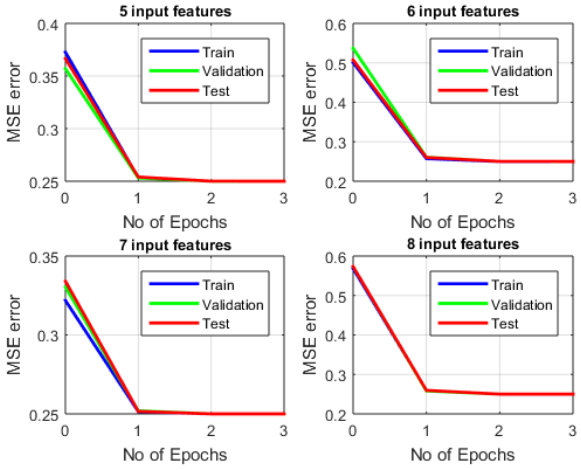

Fig 10. ANN with 80 hidden neurons

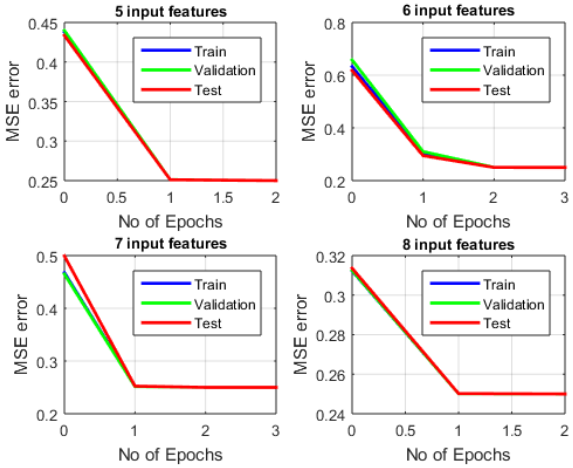

Fig 11. ANN with 100 hidden neurons

In case the performance is not satisfactory, the network has to be trained once again and the performance has to be re-analyzed.

IV. CONCLUSIONS

1. From the experiment done here, it's clear that ANN is really a powerful tool to solve any kind of classification problem.
2. The second input category i.e. Day of travel and Month of travel is a dominant feature which is obvious from the fact that the efficiency reduced drastically to 75% after dropping them.
3. The performance can be enhanced if more features like different routes, different trains are included.
4. If the continuous status of the ticket over a period of one month can be obtained, then the network will be able to generalize better.
5. More complex networks with more hidden layers can be employed to enhance the performance.

REFERENCES

[1] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Networks Toolbox: User's Guide*, The MathWorks, Inc.

[2] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edition, Pearson Education.

[3] A. Bhattad, J. Aneja, and S. Goswami, *Machine Learning based Prediction of Confirmation of Waitlisted Indian Rail Tickets*, Department of CEE, UIUC, Dec. 2016

[4] Ivan Nunes Da Silva, et al, *Artificial Neural Network: A Practical Course*, Springer

[5] A. A. Suratgar, M. V. Tavakoli, A. Hoseinbadi, *Modified Levenberg-Marquardt Method for Neural Network Training, World Academy of Science, 2007*