January 2020

A FAILURE PREDICTIVE ALGORITHM USING SEQUENCE OF EVENT CODES WITH A DEEP LEARNING MODEL (LSTM)

HP INC

A Failure Predictive Algorithm Using Sequence of Event Codes with a Deep Learning Model (LSTM)

Abstract

We are building a failure prediction algorithm with sequences of event code data from devices using a deep learning model called LSTM (Long Short-Term Memory). LSTM models are widely used to predict the sequence of words in word embedding technique. The principle idea of this algorithm is very similar to next word prediction in your cell phone when you are sending messages. The algorithm has the ability to predict the event codes leading up to the failure and resulting time of the failure. The algorithm is fed a list of the last five event codes on the specific device and can predict the next event codes leading up to the next failure by applying the LSTM model recursively. The data used to train this algorithm are telemetry data which contains event code data, time when the event happened and repair data, which is used to cut the sequence of event code data. We treated one sequence leading to a failure (failure date) as one paragraph of words. Employing this algorithm, we can accurately predict the next sequence of event codes and the resulting time until the next failure.

Problem Statements

We want to increase service efficiency and reduce support costs by predicting failures. In our current support operations, there are high field support costs due to service inefficiencies and lagging quality indicators. This algorithm will be used to predict when a printer will fail by learning trends in sequences of event codes. These event code trends help the algorithm learn when the next failure will happen. Additionally, it provides useful knowledge of what happened leading to failure. This algorithm will help us in improving the quality of the devices and reduce support cost by increasing the efficiency in analysing hundreds of thousands of customer cases.

Prior Solutions

In the past, we have not had a failure prediction algorithm using deep learning (particularly LSTM architecture). This problem was handled with a rule-based model. Engineers and technicians would triage quality concerns based on manual sorting through databases of event codes and intervention/repair data. Commercial printing division has about 90,000 repair cases logged monthly. This drives a time-consuming process of manual analysis of support case notes looking for top field issues to address. If there is not enough notes information on the cases producing customer dissatisfaction, no action can be taken.

Description

We started by pulling repair data related to one service part, such as fuser. We further filter each repair to break-fix replacement so that we can truly understand the lifespan of that part. We then pull historical telemetry data (event-code data) back to 120+ days before failure happened. We then treat each sequence of events as it is sentences of words (event-code == word). These sequences of events then ready to be used in a Long-Short Term Memory (LSTM) model.

The LSTM is a variant of Recurrent Neural Network with a way for carrying over information across many time series without getting diluted (vanished) due to a lot of processing and used typically to predict the sequence of time series events, for example next word in cell phone text. In Simplify explanation, LSTM can do this by using three gates (Forget, Input and Output gates). The forget gate determines the extent to which output of the previous timestamp state should be use, the input gate determines the extent to which the current timestamp should be used, and the output gate determines the output of the current timestamp [1, 2].

In the case of next word prediction, a LSTM model trains on the previous text the user has typed, then uses the last word to predict the next. There are many ways to train the model. First it breaks up the text into several words to gain context and then predicts the resulting word. It does this through the whole paragraph until the accuracy has reached a maximum. With this training complete, the user would input the next word and the model would predict the next word to follow. Using the same logic, we used LSTM to predict next event-code based on previous event-codes.

This approach/algorithm is used to analyze repair and event code data to determine the frequency and reasons for failures. The algorithm has two data inputs; 1) telemetric event code data, and 2) repair/intervention data. The event code data contains sequences of event codes and time of occurrence, while repair/intervention data contains information of when a customer called and reported a problem that has occurred in their printer. To get a sequence of training data, the algorithm uses a sequence of event codes and cuts the sequence using the repair data from the repair/intervention data, to signify the failure point in the sequence.

For event code data, the algorithm pulls data from the product line from the last 120 days. The data is then sorted by individual products and the sequences are then cut, each sequence leading to failure is considered one paragraph. These ‘paragraphs’ are then used to train the LSTM model; the model was found to have the highest accuracy when trained looking at every 5 codes as a word and using this to predict the next word. Here are examples of the list of event codes look like in the dataset:

```
['32.08.A3 33.01.03 13.A2.FF 40.08.00 13.A2.FF',
 '99.06.21 32.08.A3 33.01.03 13.A2.FF 40.08.00',
 '13.B2.D2 99.06.21 32.08.A3 33.01.03 13.A2.FF',
 '13.B9.D2 13.B2.D2 99.06.21 32.08.A3 33.01.03',
 '41.03.02 13.B9.D2 13.B2.D2 99.06.21 32.08.A3']
```

Fig. 1. Example of a list of event-codes from one device.

After being trained the user inputs the last 5 event codes seen from the printer set trying to be analyzed, the program will then predict the next event codes leading up to the failure sequence.

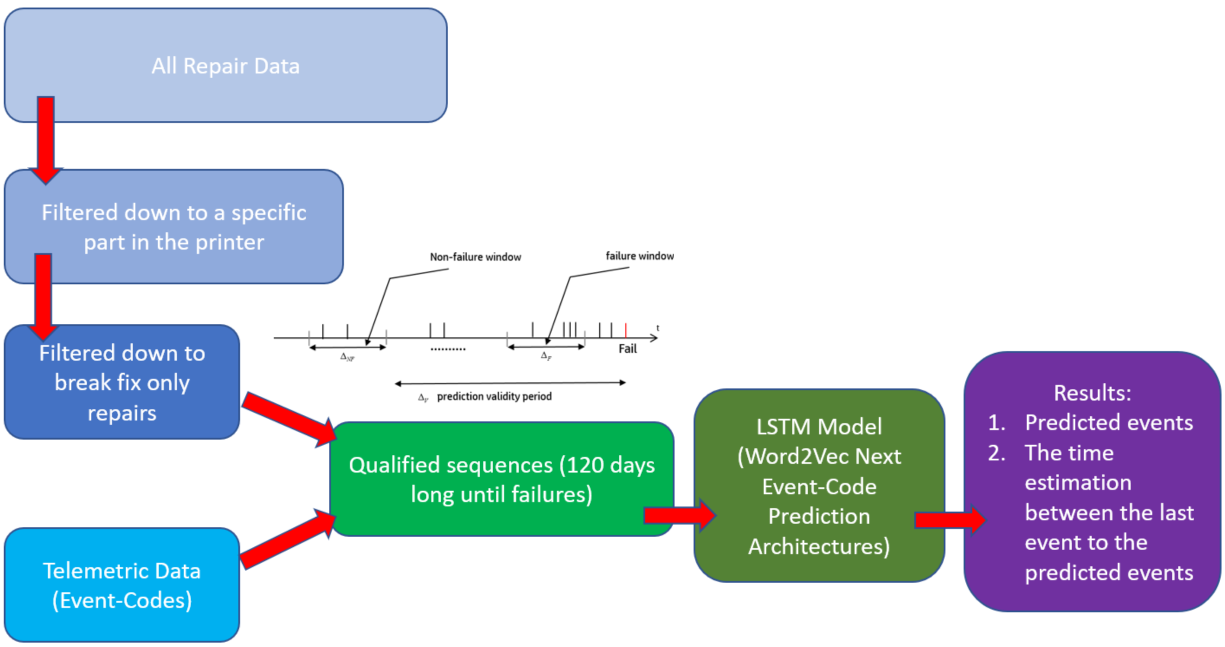

Fig. 2. The flow of the algorithm

The graphs below show the model loss and accuracy after training to stagnate around 90 percent accuracy and .5 model loss.

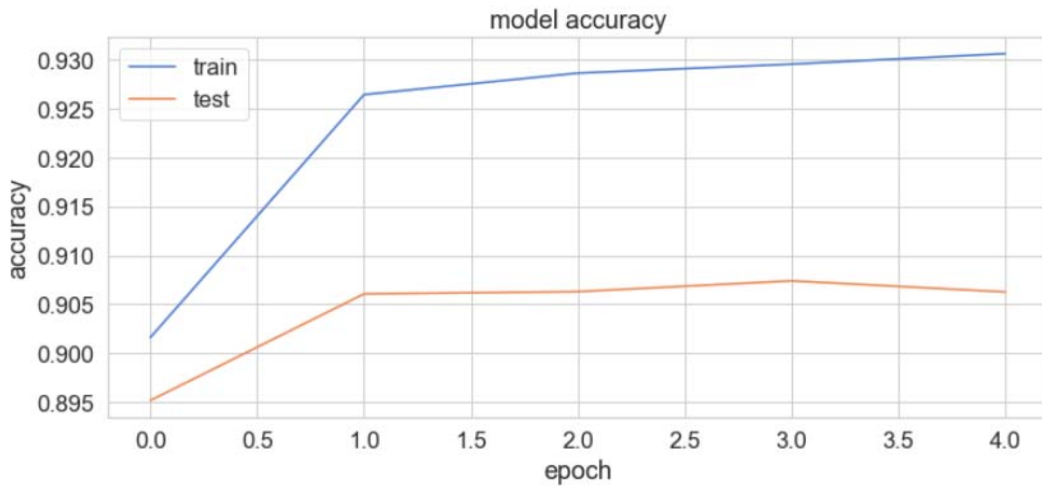

Fig. 3. Model accuracy. The blue line is accuracy of the training data and the orange line is the accuracy of the testing data.

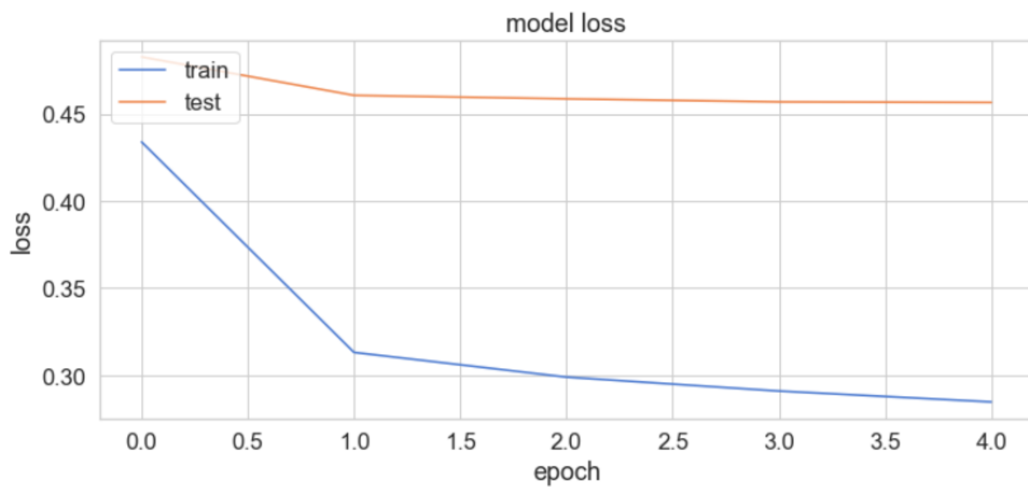

Fig. 4. Model loss. The blue line is accuracy of the training data and the orange line is the accuracy of the testing data.

As we can see both model accuracy and model loss between training and testing is quite different. This is an overfitting problem we have here. We have this problem because we are using limited dataset for now. We are increasing the size of the dataset and also including regularization term in the network.
Regarding the time from one event code to failure, we are using average time method. For example, if LSTM predicts that the next event-code is 13.B9.D2, then the algorithm will calculate historical average time between the event-code and the failure time and use the calculated results as an estimation for time required to fail.

References

[1] https://adventuresinmachinelearning.com/keras-lstm-tutorial/
[2] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". *Neural Computation*. **9** (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276.

Disclosed by Sophie Van Pelt, Francisco Arca, Katie Hirschey, Manjunath Bhuyar and Anton Wiranata, HP Inc.