Point Process and its Application
Project Report

Yuxuan Duan 516030910573

1 Introduction

Task: For this project, our goal is to predict future system failure or repair ticket (can be seen as a repair request) based on the history information of some automatic teller machines (ATMs) of a bank headquartered in North America.

Methods: The direct work of this project is to learn and utilize two recent-year models to solve the problem above: Event Recurrent Neural Network (ERNN) by [2] and Recurrent Marked Temporal Point Process (RMTPP) by [1].

Test: You may verify the codes quickly by running *main_ernn.py* or *main_rmtpp.py*. Default parameters will be used, though you may open these two files to modify them.

2 ERNN

[2] was the first point process model with fully implicit mapping intensity. Event Recurrent Neural Network (ERNN) is one of its variety, where the time series RNN is removed (showed as Figure 1).

2.1 Environment Configuration

Programming Language: Python 3.6

Machine Learning Framework: PyTorch

Assistant Libraries: NumPy (mathematical calculation), SciPy (intergration), Sklearn and Matplotlib (plotting)

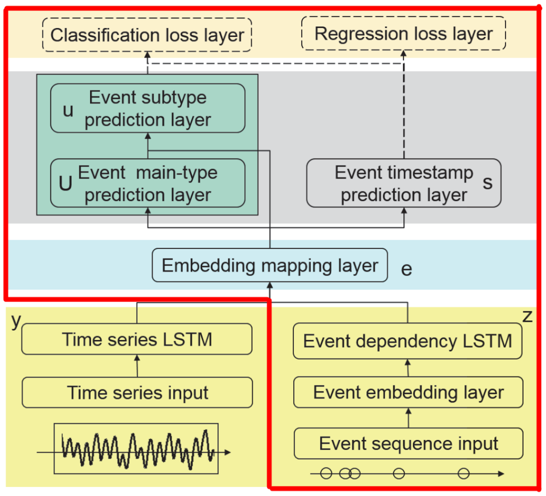

Figure 1: The network framed with red lines is ERNN

Evaluation: I used the test dataset to do the evaluation work. I only used it to check the performance, without updating the network with it.

2.2 Solution

0 - Base: The data we have span a few months, and the granularity of the timestamps is second. So to a certain degree, the data are sparse. It is difficult for the network to learn from the sequences happened at quite different time, therefore converting every sequence to 0-base time is necessary. For example, if a sequence contains 5 events: (the unit is second)

$$5 \quad 9 \quad 17 \quad 22 \quad 23$$

we shall convert it to

$$0 \quad 4 \quad 12 \quad 17 \quad 18.$$

2.3 Result

As Figure 2 shows, the model is capable of quickly succeed in classifying most of the events even after a single epoch.

Note that MAE is relatively small, I think it is because I set the network to predict the time interval between the latest event and the next event, instead of the timestamp itself. The performance may be better than directly predicting the timestamp, and is also faster for training.

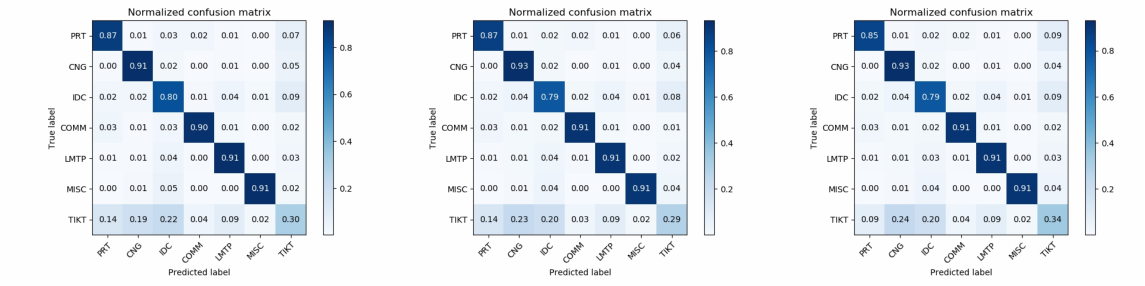

Figure 2: Confusion Matrix after 0, 25 and 49 epochs.

	Precision	Recall	F1-score	MAE on time
ERNN	0.751	0.802	0.780	0.581 days

Figure 3: Statistics of training with ERNN

3 RMTPP

[1] was a partially implicit mapping intensity point process model. Compared with ERNN, it still kept some forms of parametric modeling methods, though the network structures are quite similar.

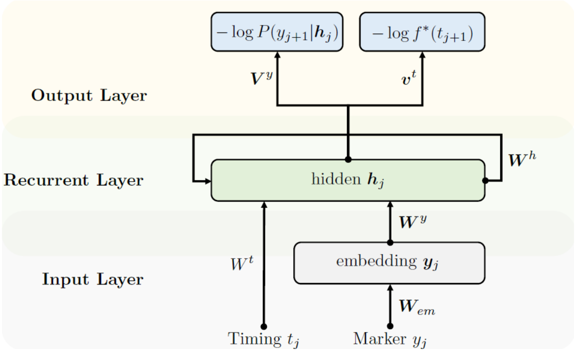

Figure 4: The structure of RMTPP

3.1 Environment Configuration

The same with ERNN.

3.2 Solution

Loss Explosion: The next event may happen after days or even weeks after the first event of a sequence. Thus, the loss value may explode when doing parametric calculations, even if we use 0-base as we did when using ERNN. So predicting time interval instead of the timestamp itself is crucial again.

3.3 Result

As RMTPP needs to do numerical integration, which is a time-consuming part. I did not evaluate the model every epoch as I had done with ERNN. Instead, I did evaluations after the 0th epoch and the 49th, as showed in Figure 5.

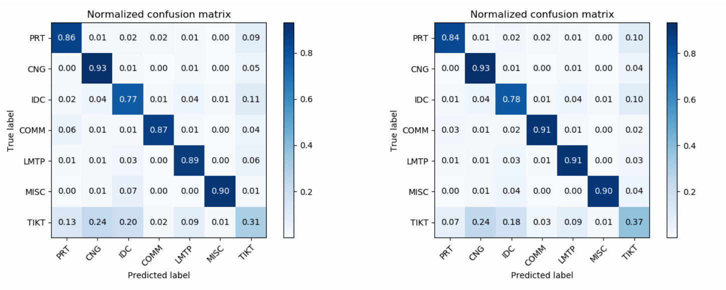

Figure 5: Confusion Matrix after 0 and 49 epochs.

	Precision	Recall	F1-score	MAE on time
RMTPP	0.753	0.807	0.779	3.29 days

Figure 6: Statistics of training with ERNN

4 Comparison

According to the two tables and graphs of confusion matrices, there is no much difference on the classification performance between the two methods. But ERNN outperforms RMTPP on the accuracy of time prediction. Also, ERNN does not need to do numerical integration, so it is more flexible and scalable to train with.

References

[1] Nan Du et al. “Recurrent Marked Temporal Point Processes: Embedding Event History to Vector”. In: *KDD* (2016).

[2] Shuai Xiao et al. “Modeling the Intensity Function of Point Process via Recurrent Neural Networks”. In: *Association for the Advancement of Artificial Intelligence* (2017).