PROBABILISTIC POINT PROCESS PREDICTION

Xinyuan Lyu, Jiajun Cheng, Yuan Shan

fancyhdr

ABSTRACT

Temporal Point Processes (TPPs) are powerful tools for modeling event sequences with irregular timestamps, crucial in domains such as finance, healthcare, and social networks. This project builds on the classical Recurrent Marked Temporal Point Process (RMTPP) model by introducing two probabilistic variants, RMTPP2 and RMTPP3, as well as a novel extension utilizing Generative Adversarial Networks (GAN). RMTPP2 and RMTPP3 enhance the modeling of event uncertainty by incorporating stochastic elements into the intensity function. The GAN-based model further improves predictive performance by learning and generating realistic event sequences, subsequently applying transfer learning for accurate event type and timing predictions. Experimental evaluations on real-world datasets demonstrate that these models outperform the baseline in temporal accuracy. This work bridges the gap between deterministic RNN-based TPP modeling and probabilistic generative paradigm, offering a promising direction for future advancements in event sequence prediction.

1 INTRODUCTION

Temporal Point Processes (TPPs) provide a flexible framework for modeling event sequences with event type information and irregular timestamps in continuous time. These sequences are widely applied in domains like finance, healthcare, and social networks. A key component of TPP prediction is the conditional intensity function, which defines the instantaneous event occurrence rate based on historical data. Effective modeling of this intensity function is crucial for capturing the underlying dynamics of the processes.

While deterministic models, such as the Recurrent Marked Temporal Point Process (RMTPP), have achieved notable success, there still remains opportunity in incorporating uncertainty and probabilistic modeling into TPP prediction. Our work try to take stochastic and probabilistic modeling into the classical RMTPP framework. Specifically, our contributions unfold in three steps:

1. Baseline Implementation: We replicated the RMTPP model using the EasyTPP framework and established its performance as our baseline.
2. Probabilistic Enhancements: We developed two novel variants—RMTPP2 and RMTPP3—that introduce randomness and probabilistic modeling into the intensity function.
3. GAN-Based Transfer Learning: We extended the TPP framework further by integrating Generative Adversarial Networks (GANs). And created the RNN which can generate new event sequences as well as make predictions.

By systematically building on the deterministic RMTPP, our approach bridges the gap between TPP prediction RNNs and generative methods.

2 RELATED WORK

2.1 RECURRENT MARKED TEMPORAL POINT PROCESSES (RMTPP)

The RMTPP model proposed by Du et al. (2016) leverages Recurrent Neural Networks (RNNs) to encode the historical sequence of events into a hidden state, which informs the modeling of event times and types. The intensity function $\lambda^*(t)$ is parameterized using a combination of the hidden state h_t and the time interval since the last event Δt: $\log \lambda^*(t) = \mathbf{w}^\top h_t + v\Delta t + b$ where $\mathbf{w}, v, b$

are learnable parameters. This approach circumvents the restrictive assumptions of traditional statistical parametric models by allowing the intensity function to adapt dynamically to complex event histories. RMTPP's integration of RNNs provides flexibility in capturing long-range dependencies, making it suitable for diverse applications. However, its deterministic nature may limit its ability to model uncertainty inherent in real-world event sequences.

2.2 Modeling Intensity Functions via Recurrent Neural Networks

Xiao et al. (2017) extend the application of RNNs to TPPs by separately modeling the background intensity and historical effects. The intensity function is decomposed as follows: $\lambda(t) = \lambda_{\text{background}}(t) + \lambda_{\text{history}}(t)$ where: $\lambda_{\text{background}}(t)$ models spontaneous events using a time-series RNN; and $\lambda_{\text{history}}(t)$ captures long-range dependencies via an event-aligned RNN. This dual-RNN architecture provides a unified framework for end-to-end training, leveraging deep learning's representational power to model intensity functions nonparametrically.

2.3 EasyTPP Framework

EasyTPP, introduced by Xue et al. (2024), addresses the challenges of standardization and reproducibility in TPP research. It provides a modular and extensible benchmarking platform with support for diverse TPP models, including RMTPP. Key features include: **Unified Data Preprocessing**: Standardized input formats and efficient padding/masking mechanisms. **Evaluation Metrics**: Comprehensive metrics such as log-likelihood and next-event prediction accuracy. **Sampling Methods**: Efficient implementations of algorithms like thinning for event generation.

EasyTPP facilitates rapid experimentation and comparison, enabling researchers to focus on model innovation rather than implementation intricacies. EasyTPP provides the platform for us to develop new RMTPP models and use the ready-made data-processing and training procedures to get our results.

3 Model Frameworks

In this section, we present the framework of our proposed models for temporal point processes. The framework consists of three parts: the original model, probabilistic modifications via intensity function sampling, and extensions incorporating Generative Adversarial Networks. Each subsection delves into the underlying architecture, rationale, and key mathematical formulations.

3.1 Original RMTPP Model

The original model employs a recurrent structure to predict the type and timing of the next event. The core components of the framework are as follows:

Model Structure: The hidden state h_j is computed using the current event type y_j, timestamp t_j, and the previous hidden state h_{j-1}:

$$h_j = \text{Relu}(W^y y_j + W^t t_j + W^h h_{j-1} + b_h),$$

where W^y, W^t, W^h, b_h are learnable parameters. The probability of the next event type y_{j+1} is computed as:

$$P(y_{j+1} = k \mid h_j) = \frac{\exp(V_k^\top h_j + b_k^y)}{\sum_k \exp(V_k^\top h_j + b_k^y)},$$

where V_k, b_k^y are parameters associated with event type k. And the intensity function governs the rate of occurrence of events at any given time t:

$$\lambda^*(t) = \exp(V^t h_j + \beta^t(t - t_j) + b^t) = \exp(V^t h_j + b^t) \cdot \exp(\beta^t(t - t_j)) = \exp(\gamma(h_j)) \exp(\beta^t(t - t_j))$$

where V^t, β^t, b^t are learnable parameters, and $\gamma(h_j) = V^t h_j + b^t$. And from the conditional intensity function, the conditional density function for the event time t is:

$$f^*(t) = \lambda^*(t) \exp\left(-\int_{t_j}^{t} \lambda^*(\tau) d\tau\right).$$

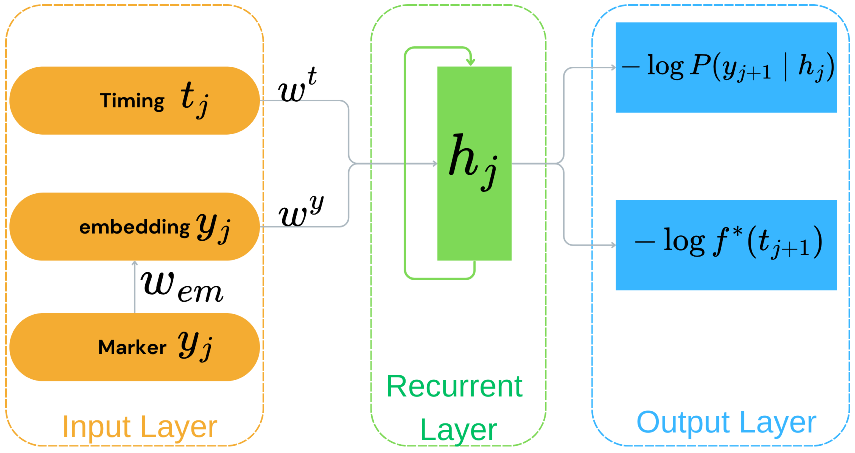


Figure 1: The Structure of Original RMTPP.

Parameter Learning: The parameters are optimized by maximizing the log-likelihood of observed sequences:

$$\ell(\{S^i\}) = \sum_i \sum_j \left(\log P(y^i_{j+1} \mid h_j) + \log f(d^i_{j+1} \mid h_j)\right),$$

where $S^i = \{t^i_j, y^i_j\}$ is the i-th event sequence.

Prediction: Event time and type prediction through the expectation:

$$\hat{t}_{j+1} = \int_{t_j}^{\infty} t f^*(t) dt; \;\; \hat{y}_{j+1} = \arg\max_k \exp(V_k^\top h_j + b_k^y),$$

Or through sampling:

$$\hat{t}_{j+1} \sim f^*(t), \quad \hat{y}_{j+1} \sim P(y|\mathcal{H}_t).$$

3.2 Probabilistic Modeling: The Two Variants of RMTPP

To introduce randomness and uncertainty into the intensity function, we propose two stochastic variants:

3.2.1 Variant 1: Stochastic Latent Function for Intensity

We can add a random latent function between intensity function and h_j and t. So in this variant, the intensity function is governed by a random latent element $X(t)$ and $X(t)$ can be sampled:

$$\lambda^*(t) = \exp(X(t)), \quad X(t)|\mathcal{H}_t \sim \mathcal{N}(\mu(t|\mathcal{H}_t), \sigma^2(t|\mathcal{H}_t)),$$

where:

$$\mu(t|\mathcal{H}_t) = W_\mu h_j + b_\mu + \beta^t_\mu (t - t_j), \quad \log \sigma(t|\mathcal{H}_t) = W_\sigma h_j + b_\sigma + \beta^t_\sigma (t - t_j).$$

Through this approach, for each t, the latent function about t is sampled and then therefore the $\lambda^*(t)$ which is about the latent function is also sampled probabilistically. In the following content we will refer this model as **"RMTPP2"**.

3.2.2 Variant 2: Uncertainty in the Multiplier of Intensity Function

We can see that the intensity function can be decomposed into two parts in original model such that:

$$\lambda^*(t) = \exp(\gamma(h_j)) \exp(\beta^t (t - t_j))$$

Where the multiplier $\exp(\gamma(h_j))$ is depend on h_j deterministically. So our variant incorporates uncertainty by modeling the intensity function multiplier as a random variable that $\gamma(h_j)$ is sampled from a Gaussian distribution:

$$\gamma(h_j) \sim \mathcal{N}(\mu(h_j), \sigma^2(h_j)),$$

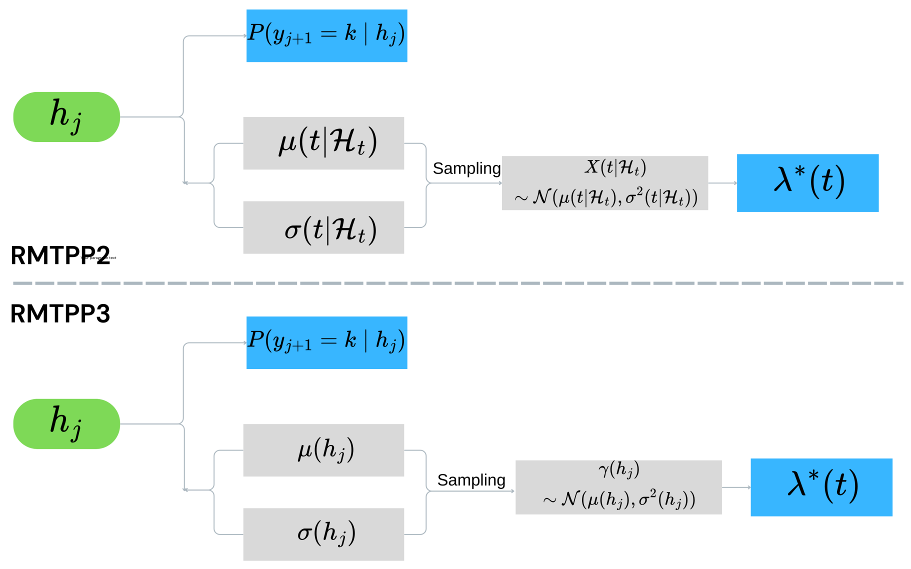

$P(y_{j+1} = k \mid h_j)$
h_j
$\mu(t|\mathcal{H}_t)$
$\sigma(t|\mathcal{H}_t)$
Sampling
$X(t|\mathcal{H}_t) \sim \mathcal{N}(\mu(t|\mathcal{H}_t), \sigma^2(t|\mathcal{H}_t))$
$\lambda^*(t)$
RMTPP2
RMTPP3
$P(y_{j+1} = k \mid h_j)$
h_j
$\mu(h_j)$
$\sigma(h_j)$
Sampling
$\gamma(h_j) \sim \mathcal{N}(\mu(h_j), \sigma^2(h_j))$
$\lambda^*(t)$

Figure 2: The Structure of RMTPP 2 and 3.

and:

$$\mu(h_j) = W_\mu h_j + b_\mu, \quad \log \sigma(h_j) = W_\sigma h_j + b_\sigma + \beta_\sigma(t - t_j).$$

This design achieves probabilistic modeling by introducing randomness into the intensity function multiplier. In the following content we will refer this model as **"RMTPP3"**.

3.3 Incorporating GAN into RMTPP

The two variants above can be considered as inspired by VAE. And we also developed a new RMTPP model using the thought of GAN. We first design and train a GAN model in which the generator can generate new fake event sequences include the event-type sequence and time sequence, and the discriminator can decide whether a event sequence is real or fake. The structure is shown in Figure 3

Generator: The generator is a RNN. The initial event point (y_0, t_0) and initial hidden state h_0 which can be sampled is given to generator-RNN, then the generator can generate (y_1, t_1) through h_1 and goes on this process. And from h_j to get (y_j, t_j) we use intensity function and sampling:

$$t_j \sim \lambda^*(t|h_j), \quad y_j \sim P(y|h_j).$$

And once the point is generated, it will goes into the RNN for generating the next point, which is:

$$h_j = \text{Relu}(W^y y_{j-1} + W^t t_{j-1} + W^h h_{j-1} + b_h),$$

where the y_j, t_j are just generated by previous h_{j-1}. So given the oringinal event point, the generator can generate the whole sequence.

Discriminator: The discriminator is also a RNN, but the input is the whole sequence and the output is a real-or-fake score. After the sequence goes through the RNN, we get the last hidden state h_n and then we have:

$$Real\ or\ fake\ Score = \text{Sigmoid}(W^D h_n + b^D),$$

Parameter Learning: As other GANs, the parameters are optimized by

$$\max_G \min_D \{\mathbb{E}\left[-\log(D(\text{real sequence}))\right] + \mathbb{E}\left[\log(D(\text{fake sequence}))\right]\}$$

Transfer Learning: After tain the GAN, we can transfer the generator to a predictor. Since the generator can gradually produce sequences that are indistinguishable from real ones during its adversarial game with the discriminator, it is capable of learning the pattern characteristics of real sequences. Given an event point as input, the generator should be able to generate the next event

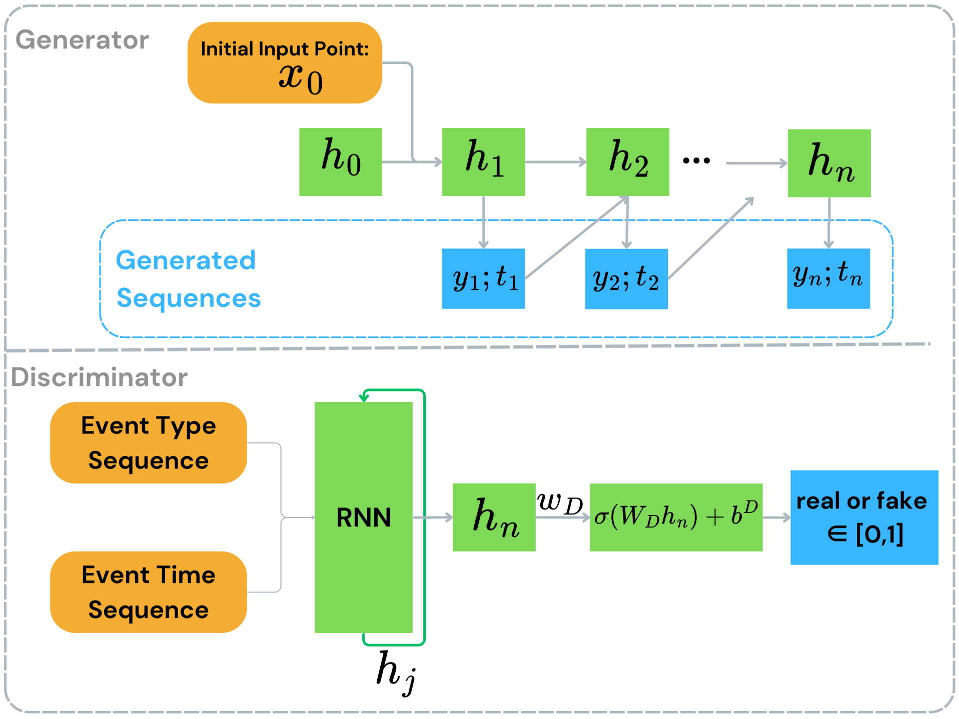


Figure 3: The Structure of GAN-RRMTPP.

point that conforms to the pattern and rules of real sequences, thereby completing the prediction task. In the original generator, the hidden layer information used to generate the next event point is entirely inherited from the previously generated virtual event points. However, for the prediction task, we can input real prior information, i.e., provide the RNN with the real sequence information of $j - 1$ and earlier time steps, enabling it to predict the event point at time j.

Additionally, we observed that the GAN-RNN, with its parameters directly frozen, demonstrated good performance in predicting the time t of the next event but performed poorly in predicting the event type y. To address this issue, we extended the original GAN-RNN by adding a classifier head, where the hidden state h_j is passed through a linear multi-layer perceptron (MLP) to output category predictions. While freezing the parameters of the RNN, we performed additional parameter training for the MLP. In subsequent experimental results, we demonstrate that this transfer learning approach significantly improves both time prediction and event type prediction performance.

4 Experiments

We evaluate our RMTPP variants in large-scale real world data. We compare it to original RMTPP model showing that RMTPP variants and GAN-RMTPP may have better performance compared to oringinal model.

4.1 Data

We conduct experiments on the following two real-world event sequence datasets:

Taxi Data Whong (2014):. Each record in this dataset contains a time-stamped taxi pick-up or drop-off event in one of New York City's five boroughs. The event types are defined as combinations of the borough and the event type (pick-up or drop-off), resulting in K = 10 distinct event types. The dataset used is a randomly sampled subset containing 2,000 drivers, where each driver's sequence has an average length of 39 events.

Taobao Data Tianchi Team (2018). This dataset contains temporal sequences of user-item interactions collected from Taobao's e-commerce platform over a nine-day period spanning November 25 to December 03, 2017. Each interaction event is characterized by a timestamp and the corresponding item category.The event space was conducted by selecting the K = 20 most frequent item categories, where the top 19 categories are preserved as individual event types and the remaining categories are aggregated into a single type. The dataset used is a subset containing n = 4,800 users selected based on activity frequency, with mean sequence length = 150.

4.2 Experimental Results.

We evaluate model performance using two complementary metrics: Root Mean Square Error (RMSE) and Accuracy. RMSE assesses temporal prediction accuracy by quantifying errors in inter-event time intervals (Δt), with lower values indicating better performance. Accuracy measures the model's ability to correctly predict the next event type (k), with higher values reflecting superior categorical predictions. All metrics presented in the figures are computed on the **test** set.

For training, the models are trained using the Adam optimizer with a batch size of 256, an initial learning rate of 0.001, and for a maximum of 200 epochs.

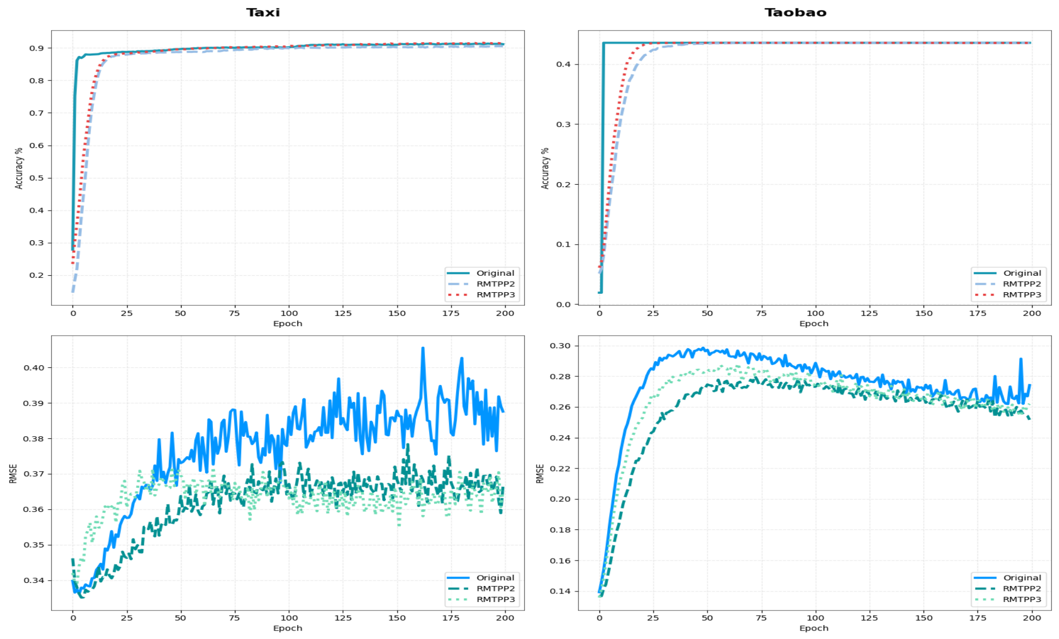


Figure 4: Performance evaluation during training for predicting both type and timing of the next event. The top row presents the Accuracy of predicting the event types, and the bottom row gives the RMSE of predicting the event timings; the left column is Taxi and right is Taobao.

Figure 4 presents the training dynamics of three models (Original, RMTPP2, and RMTPP3) on both datasets. Note that RMTPP-GAN is excluded from this comparison due to the adversarial training processes is different from these three, which makes epoch-wise comparisons impractical. For the Taxi dataset, all models demonstrate rapid convergence in accuracy within the first 25 epochs, ultimately achieving over 90% accuracy. And the RMSE curves show that RMTPP2 and RMTPP3 consistently outperform the original model, maintaining lower RMSE almost throughout the training process and the ultimate RMSE for Original model is about 0.39 while the values for RMTPP2 and RMTPP3 are both about 0.36. For the Taobao dataset, all models have worse performances. The RMTPP2 and RMTPP3 still have the same accuracy as original model but have slightly smaller RMSE than original model.

Figure 5 illustrates the prediction performance results across all variants of the model including GAN-RMTPP. For the Taxi dataset, we observe both RMTPP2 and RMTPP3 achieve comparable accuracy to the baseline, while RMTPP-GAN exhibits marginally lower types prediction perfor-

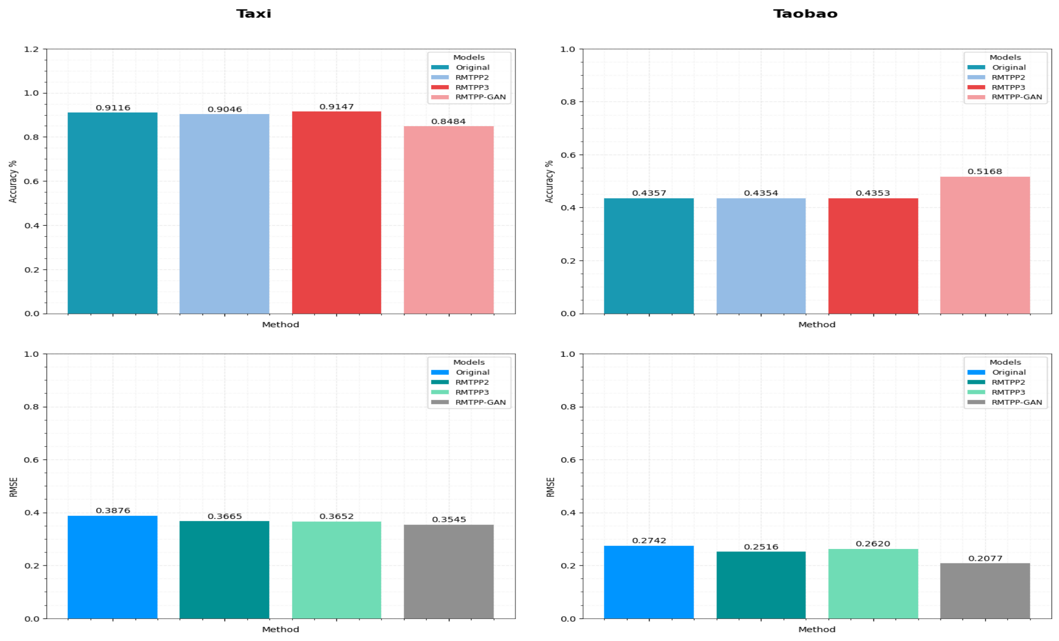
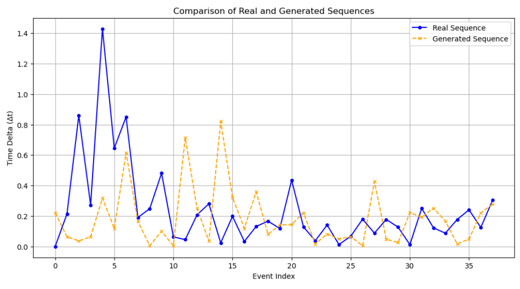


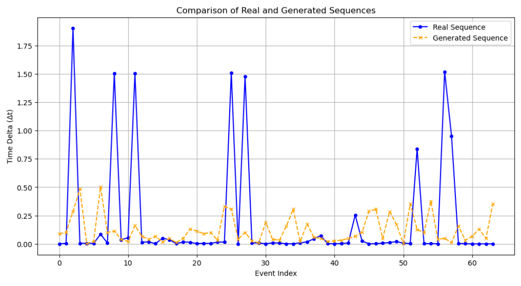
Figure 5: Performance evaluation for predicting both type and timing of the next event. The top row presents the accuracy of predicting the event types, and the bottom row gives the RMSE of predicting the event timings; the left column is Taxi and right is Taobao.

mance (acc=0.85). Notably, RMTPP-GAN demonstrates the strength of temporal modeling capacity with the lowest RMSE (rmse=0.35), representing an improvement in timings prediction accuracy over the baseline. On the Taobao dataset, RMTPP-GAN substantially outperforms all competing methods in both metrics, achieving the highest accuracy (acc=0.52) and lowest RMSE (rmse=0.21) among all models. These empirical results validate that the GAN-RNN framework may enhances both temporal and categorical prediction capabilities.

Additionally, Figure 6 presents examples of sequences generated by the generator of GAN-RMTPP. It can be observed that the generated sequences closely align with the patterns and trends of the real sequences, indicating that the generator has effectively learned the data patterns through adversarial training with the discriminator.



(a) Taxi Dataset



(b) Taobao Dataset

Figure 6: The generated delta time sequences as examples on Taxi and Taobao datasets.

5 Conclusion

This study explores probabilistic modeling methods for Temporal Point Processes (TPPs). By extending the classical Recurrent Marked Temporal Point Process (RMTPP) model, we propose two probabilistic variants (RMTPP2 and RMTPP3) and a novel model incorporating Generative Adversarial Networks (GAN). Our results demonstrate that these enhanced models show some improvements in modeling the uncertainty of event sequences and improving predictive performance.

Specifically, RMTPP2 and RMTPP3 introduce randomness and probabilistic features into the intensity function via sampling, thereby exhibiting better robustness and flexibility in both time and event type predictions. Furthermore, the generator in GAN-RMTPP learns and generates realistic event sequences and can significantly improves the accuracy of time and event type predictions through transfer learning. On the Taxi dataset, GAN-RMTPP achieves the lowest RMSE, showcasing its superior temporal modeling capability. On the Taobao dataset, GAN-RMTPP not only leads in temporal prediction accuracy but also achieves the best performance in event type classification.

By combining RNN modeling with probabilistic reasoning and generative adversarial learning, our approach effectively enhances the performance of temporal point process predictions, offering new research ideas in this domain. Future work could further extend and modify the models and test the models in more datasets to evaluate the models robustness.

References

Nan Du, Hanjun Dai, Rakshit Trivedi, Uttaran Bhattacharya Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1555–1564. ACM, 2016. doi: 10.1145/2939672.2939875.

Tianchi Team. User Behavior Data from Taobao for Recommendation. Tianchi Open Dataset, Alibaba Group, 2018. URL `https://tianchi.aliyun.com/dataset/649`. Dataset of user behaviors from Taobao for recommendation problems with implicit feedback. The dataset is distributed under CC BY-NC-SA 4.0 license. Accessed: 2024-12-06.

Chris Whong. Foil nyc taxi trip data, 2014. URL `https://chriswhong.com/open-data/foil_nyc_taxi/`. Accessed: 2024-12-06.

Shuai Xiao, Junchi Yan, Stephen M. Chu, Xiaokang Yang, and Hongyuan Zha. Modeling the intensity function of point process via recurrent neural networks, 2017. URL `https://arxiv.org/abs/1705.08982`.

Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao Jiang, Chen Pan, James Y. Zhang, Qingsong Wen, Jun Zhou, and Hongyuan Mei. Easytpp: Towards open benchmarking temporal point processes. In *International Conference on Learning Representations (ICLR)*, 2024. URL `https://arxiv.org/abs/2307.08097`.