**A MACHINE LEARNING APPROACH FOR LEARNING TEMPORAL POINT PROCESS**

Andrija Petrović\*2, Aleksa Biserčić1, Boris Delibašić1, Dimitrije Milenković1

1University of Belgrade Faculty of Organizational Sciences

2 Singidunum University

\*Corresponding author, e-mail: apetrovic@singidiunum.ac.rs

***Abstract:*** *Despite a vast application of temporal point processes in infectious disease diffusion forecasting, ecommerce, traffic prediction, preventive maintenance and many others there is no significant development in improving the simulation and prediction of temporal point processes in real world environments. With this problem at hand we propose a novel methodology for learning temporal point processes based on one-dimensional numerical integration techniques. These techniques are used for linearising the negative maximum likelihood (neML) function and to enable backpropagation of the neML derivatives. Our approach is tested on two real-life datasets. Firstly, on high frequency point process data, prediction of highway traffic, and secondly, on a very low frequency point processes dataset, prediction of ski injuries in ski resorts. Four different point process baseline models were compared: second-order Polynomial inhomogeneous process, Hawkes process with exponential kernel, Gaussian process and Poisson process. The results show the ability of the proposed methodology to generalize on different datasets and illustrate how different numerical integration techniques and mathematical models influence the quality of the obtained models. The presented methodology is not limited to these datasets and can be further used to optimize and predict other processes that are based on temporal point processes.*

***Keywords****: temporal point process, Hawkes process, Poisson process, numerical integration, highway traffic prediction, ski injury prediction*

**1. introduction**

Nowadays, one of the most popular research area is focused on modelling event sequences. Event sequencing has become extremely popular in a wide variety of applications such as road traffic estimation (Ryu & Steven, 1998, p. 735), epidemiology prediction (Zahrieh, 2017), network activities (Liu & Wu, 2017, p. 992), bioinformatics (Farajtabar et al., 2017, p. 1305), e-commerce etc. Event data carry information about event occurrence. Additionally, event data can also provide information about classes of an event, event type, participator, etc. This type of point process is known as a marked point process.

A point process is extremely useful in modelling traffic congestion and traffic event occurrences, e.g. arrival of vehicles, pedestrian movement, etc. (Jia, Jiang, Liu, Cui & Shi, 2018, p. 581). Simulating highway traffic and predicting highway congestion is one of the main problems connected with point process modelling (Nguyen, Krishnakumari, Calvert, Vu & Van, 2019, p. 238).

Compared to time-series, event occurrences are treated as random variables generated in an asynchronous manner, which makes them fundamentally different from time series where equal and fixed time intervals are assumed. This property makes them useful in a wide variety of applications where discretizing events to a fixed interval would result in bad prediction performances and high computational cost.

Generally, there are two types of point process models: temporal (univariate) point process and spatial-temporal (multivariate) point process. In the case of the univariate point process, the objective is to model temporally correlated event occurrence, whereas in spatial-temporal point process the event occurrences are correlated in space and time. Multivariate point process is generally mostly used in the analysis of protein patterns (Jacobsen et al., 2007, p. 1289) and financial market predictions (Bowsher, 2007, p. 876). The general formulation of the point process makes them available to model event occurrences continuous or discontinuous (with jumps). Additionally, the point process can be further generalized by stochastic differential equations to stochastic point process.

The main idea behind different types of point processes models is hidden in modelling conditional intensity function (CIF). A CIF can be interpreted heuristically as the expected number of events that are going to occur in an infinitely small timestamp (*dt*). CIF can be modelled as constant (homogeneous process) or as a function of time (inhomogeneous process). Learning an intensity function from a given dataset presents one of the most popular subjects of research (Mei & Eisner, 2017, p. 6754).

In this paper, we present a data-driven approach for learning different types of CIFs used in temporal point process models. Our approach is based on an implementation of numerical integration methods for linearization of negative maximum likelihood (neML) in order to backpropagate derivatives of neML.

We tested our methodology on two real-life datasets that consist of exact timestamps. The first dataset includes highway toll passes recordings, and is a high-frequency dataset. The second dataset includes timestamps when ski injuries occurred, and is a low-frequency dataset. Our methodology showed that it can be successfully used for various types of CIFs. Furthermore, four different baseline models based on neML scores: second-order Polynomial inhomogeneous process, Hawkes with exponential kernel, Gaussian process and Poisson process were compared. The proposed methodology was evaluated on several metrics, amongst which is the minimization of negative log likelihood loss for demonstration of how well models fitted conditional intensity functions, and the mean absolute error for evaluating how well would prediction be for future time events.

The remainder of the paper is structured as follows. In section 2 the related work is reviewed. Background methodology, Point process and Ogata’s modified thinning algorithm are presented in section 3. A novel methodology for learning point process is presented in section 4. Experimental setup and results on real-world applications are presented in sections 5 and 6, respectively. The conclusions are drawn in section 7

**2. RELATED WORK**

We structure the discussion of the related work onto two broad, previously mentioned, categories: intensity approaches and intensity-free approaches. The intensity approaches present methods where a point process is modelled by different functional forms of CIFs (Rasmussen, 2011). Intensity-free method model point processes with some type of unsupervised machine learning algorithms.

**Intensity approaches** present the oldest approaches in point process modelling. They rely on a functional form that completely depends on the CIF. The Poisson process presents the simplest point process where conditional intensity function has a constant value (Last & Penrose, 2017). The more complicated variant of this process is observed when the CIF is modelled as a product of kernels (Kirchner, 2017). Recent research proposed different variants of modelling CIF by deep neural networks (Mei & Eisner, 2017, p. 6754, Xiao, Yan, Yang, Zha & Chu, 2017). Xiao et al. (2017) presented an interesting approach of modelling CIF by a recurrent neural network. However, in this paper authors assumed that integral in negative maximum likelihood is correlated only with the current timestamp. Even though this strong assumption cannot be justified by theoretical properties of point process models the obtained results were significantly better compared to well-known baseline models. Chen et. al (Chen, Rubanova, Bettencourt & Duvenaud, 2018, p. 6571) and Zhang et. al (Zhang et al, 2019) presented an interesting approach for modelling dynamics by deep neural networks. Moreover, the authors presented an interesting example where the point process is modelled by a differential equation and solved using the Euler method. Besides, the authors implemented the backpropagation technique for reducing memory complexity during the training phase.

**Intensity-free approaches** are based on modelling point processes by unsupervised learning techniques (Ghahramani, 2003, p. 72). Compared to intensity approaches these methods can obtain better results, but they are more prone to overfitting due to smaller datasets or large expressive powers of the model. Variational autoencoders (VAE) present unsupervised machine learning algorithms that are mostly used for point process modelling. The Action Point Process variational autoencoder (APP-VAE) presents a variational auto-encoder that can capture the distribution over the times and categories of action sequences (Mehrasa et al., 2019, p. 3165). The APP-VAE obtained state-of-the-art results on the MultiTHUMOS and Breakfast datasets. A declustering based hidden variable model that leads to an efficient inference procedure via a variational autoencoder for solving multivariate highly correlated point process is presented by Yuan et. al. (2019). Besides VAE, generative adversarial networks (GANs) have recently been proposed as a method for describing event occurrences (Xiao et. al., 2017). The authors proposed an intensity-free approach for point process modelling that transforms nuisance processes to a true underlying distribution by using Wasserstein GANs. Experiments on various synthetic and real-world data substantiate the superiority of the proposed point process model over conventional ones.

The model presented in this paper belongs to the class of intensity approaches. Compared to the standard intensity approaches our model has more expressive power, whereas compared to intensity-free approaches it is less prone to overfitting.

**2. Background Methodology**

**2.1 Point process**

In order to be able to define the framework for learning temporal point process, in this section the background methodology of the point process will be presented.

In the term *point process*, the word *point* is used as a representation of the event on the timeline. Furthermore, it is accepted by the authors to think about the temporal point pattern as an ordered array of times when events occurred. Mutual to such events is that there is no information about how many events will occur and when they will happen. Usually, there is a more complex mechanism behind events that explains their nature. To explain this nature and predict future events, it's convenient to use a tool for stochastic process modelling point patterns - a temporal point process.

To describe phenomena over time, the evolutionary character is essential. *Evolutionarity* means that what is happening now only depends on what happened in the past, so future events don’t have any impact on the current state. Based on the evolutionarity assumption, the challenge to describe and predict the temporal point pattern boils down to finding a stochastic model for the time of the next event given the times of previous events. This knowledge of the times of previous events up to but not including current time *t* is given by history :



One of the possible approaches to define a point process is to find the distribution of time length between subsequent events. This time is also known as interevent time. Let  be the conditional density function of the time of the next event  given the history of previous events . Since the density functions specify the distributions of all interevent times, one by one, and according to the evolutionary character of the process, distribution of all events is given by the joint density (respecting the rule that the joint density for a bivariate random variable can be represented as:



Another popular approach involves a conditional intensity function as a function that defines expectation that an event will occur in the infinitesimal interval around *t* given the history *H* at times before time *t*.

The conditional intensity is defined to be the expected rate at which events will tend to occur around time *t* given the :



where *N* denotes the number of events occurred in interval .

Both of the presented approaches assume the fact that the CIF is a more convenient and intuitive way of specifying how the present depends on the past in an evolutionary point process.

Considering the conditional density and its corresponding cumulative distribution function  for any  , Rasmussen (2011) showed that the CIF is defined by:



In other words, the CIF specifies the mean number of events in a region conditional on the past. It’s assumed that there are no overlapping points so that there is either zero or one point in an infinitesimal interval. Therefore the CIF of a point process can be described by the following proposition (Rasmussen, 2011):

**Proposition 1.** A conditional intensity function  uniquely defines a point process if it satisfies the following conditions for any point pattern and any :

1.  is non-negative and integrable on any interval starting at , and

2. for  .

**2.1. Forms of The Conditional Intensity Function**

Based on the true underlying distribution of the point process, a wide variety of functional forms can be used to model event occurrences. In this paper, we consider four different functional forms: second-order Polynomial inhomogeneous process, Hawkes with exponential kernel, Gaussian process and Poisson process.

The form of the *Poisson process* describes a completely random process independent of history. Therefore, conditional intensity function is independent of time, such that:



The common way to introduce a time dependency to a model is to turn it into a polynomial form. Conditional intensity function of a second-order Polynomial inhomogeneous process can be presented as:



where are learnable parameters.

Keeping in mind that a significant number of real world phenomena have a clustered nature, Hawkes suggests a functional form for a clustered point process (Hawkes, 1971).The clustered nature explains that observation of a few points in the recent past increases the chance that there will be new points soon. In other words, the probability of seeing a new event increases due to previous events, in the following manner:



where  are learnable parameters.

The Hawkes form emphasizes that each time a new point arrives in this process, the conditional intensity grows by *α* and then decreases exponentially back towards *µ*. The Hawkes process gives a convenient form for the CIF that can be used as a foundation for adapting to a specific problem. The standard form is presented as:



where represents the kernel function that can be replaced to better suit specific problems. The base form of Hawkes uses an exponential kernel. Another form where the kernel is represented as a sum of Gaussian basis functions (Xu, Farajtabar & Zha, 2016, p. 1717) is known as Gaussian process:



Unlike polynomial forms, Hawkes forms offer a chance to define a wide range of different CIFs that can describe very complex processes. The introduction of dependence on the moments of previous events opens the possibility for other features of the process to be encoded in those times.

**2.2 Point process simulation - Ogata’s modified thinning algorithm**

The basic idea behind Ogata’s modified thinning is to find out when the next event is going to occur. To do this a homogeneous Poisson process on some interval  for some chosen function  (this is the maximum distance we may go forward in time from t and it may be infinite) is simulated. This Poisson process has a chosen constant intensity on , that fulfills



Actually, one needs only to simulate the first point  of this Poisson process. There are now two possibilities: If  , then there is no point in , so one starts again from , but if ,there may be a point in . In the latter case it has to be decided whether to keep this point or not. To get the correct intensity, the point is kept with probability . Whether or not the point is kept, the algorithm is started all over at . The pseudocode of Ogata’s modified thinning algorithm is presented in Fig. 1.

1. Set *t* = 0 and *n* = 0.

2. Repeat until *t* > *T*:

(a) Compute *m(t)* and *l(t)*

(b) Generate independent random variables *s* ∼ Exp*(m(t))* and *U* ∼ Unif([0, 1]).

(c) If *s* > *l(t)*,set *t* = *t* + *l(t)*.

(d) Else if *t* + *s* > *T* or *U* > *λ*∗ (*t* + *s*)/*m(t)*, set *t* = *t* + *s*.

(e) Otherwise, set *n* = *n* + 1, *tn* = t + *s*, *t* = *t* + *s*.

3. Output is {*t1*, . . . , *tn*}.

Figure 1. Ogata’s modified thinning algorithm – pseudocode

**3. Methodology for Learning Point Process**

We here propose the methodology for learning point processes.

**3.1. Poisson process likelihood Function**

For an observed point pattern on  for some given  , a likelihood function is given by:



where  stands for an integrated CIF, such as:



In order to fit the parameters of the point process to the observed event data, it is necessary to define a loss function. The loss function can be defined as a negative log-likelihood:



This form provides an appropriate and powerful instrument to optimize parameters of the defined function with the objective to minimize loss function.

**3.3. A Numerical Approximation of Integration of Loss Function**

Since the function  can take any functional form, it is expected that many of them will not have an analytic solution of the integral.

Moreover, in some cases even though tractable integration would be possible, it is often not feasible due to computational limitations. For example, in the case of Hawkes process the loss function has the following form:



It can be noticed that the loss function is analytically tractable, but it quickly tends to fall into the trap of overflowing due to the memory space of float numbers. The cause of this lies in exponential function integration where for small values of *t*, will tend to infinity.

Therefore, in the case when integration is not possible due to a non-integrable function or due to an overflow problem, it is possible to apply an numerical approximation of the one-dimensional integral. In this paper we used four different numerical approximation techniques for integration: Trapezoidal method, Simpson’s method, Euler's method, and Gaussian quadrature method based on interpolation functions to approximate the intractable integral inside the loss function.

Euler's method uses the idea of local linearity or linear approximation, with short-distance tangents being used to approximate the solution of the problem. Assuming that the value of the conditional intensity  is known at the point , , the integral of CIF between two time steps  and  can be approximated as:



Where *N* is the total number of intervals in range between *t1* and *t2.* The formulation of the Implicit Euler integration method is the same as in the case of the Euler method, the only difference is the way integral is evaluated in the backward manner.

The trapezoidal method approximates the integral by region's graphical function as a trapezoid and calculates its area. The basic form of the trapezoidal method applied to CIF takes the following form:



In the case when difference between time intervals is fixed , this difference is known as integration or discretisation step, abbreviated with *h.*

Simpson's method approximates the integral by replacing the sub integral function on the integration interval  with a quadratic interpolation polynomial, passed through equidistant nodes  which gives the final expression: (Radunović, 2004).



The Gaussian quadrature forms approximate the integral of a function as a weighted sum of the values of the functions at certain points within the domain of integration (Radunović, 2004).

Based on the previously defined CIF, the integration step *h* can be chosen with respect to a desired approximation. Therefore, the integration step defines the precision level of the integral estimation. The higher the number, the more accurate the estimate will be, but the computational cost of training will increase. Hence, this number presents a meta parameter that needs to be predefined with respect to the trade-off between accuracy and training duration.

**4. Framework**

Based on the proposed methodology in this paper we implemented a general framework for learning point processes. The framework consists from three distinct parts: Data cleaning and processing, model and hyperparameter selection, evaluation and simulation part. The framework is presented in Fig. 2.

Data cleaning and preprocessing part consist of methods that are used for cleaning and transforming raw data prior to processing and analysis. It is important step that involves reformatting data, making corrections to data and making time sequences of occurred events. The transformation used in this part depends on problem formulations and raw data formats. Additionally, during this part the dataset is split on training, validation and test set.

The first step in model selection and hyperparameter tuning step is to choose the point process model. In this framework the decision maker must choose between four different kind of models: The Poisson temporal point process, the Gaussian point process, the Poisson polynomial process and the Hawkes process. The choice of a model primary depends on the way events are generated. Therefore, it is advisable to plot approximations of CIF with respect to moving windows and choose the model that best fit to it.

After model selection, the integration step and integration method must be chosen. In this framework three different kinds of integration rules are presented: Implicit Euler, Trapezoidal, Simpsons and Gaussian quadrature method. Based on the integration rule, the integration step must be finely tuned in order to reduce the approximation error of the integral. Depending on the frequency of events, the integration step must be small and sufficiently large when event frequency is high and low, respectively. After model selection the model is trained on training set.

The validation of integration step is applied with respect to log likelihood metric on the validation set. If the decision maker is satisfied with the obtained performances of the selected and validated models, he/she can proceed to the evaluation and simulation phase.

In evaluation and simulation part the model is simulated and results of simulations are used for testing model on test set. Besides, log likelihood metric, mean absolute error is also used to evaluate model performances. Additionally, in combination with simulation, the obtained models can be further used in order to predict occurrence of next events or to summarize some important statistical measures that can provide useful information to the decision maker.

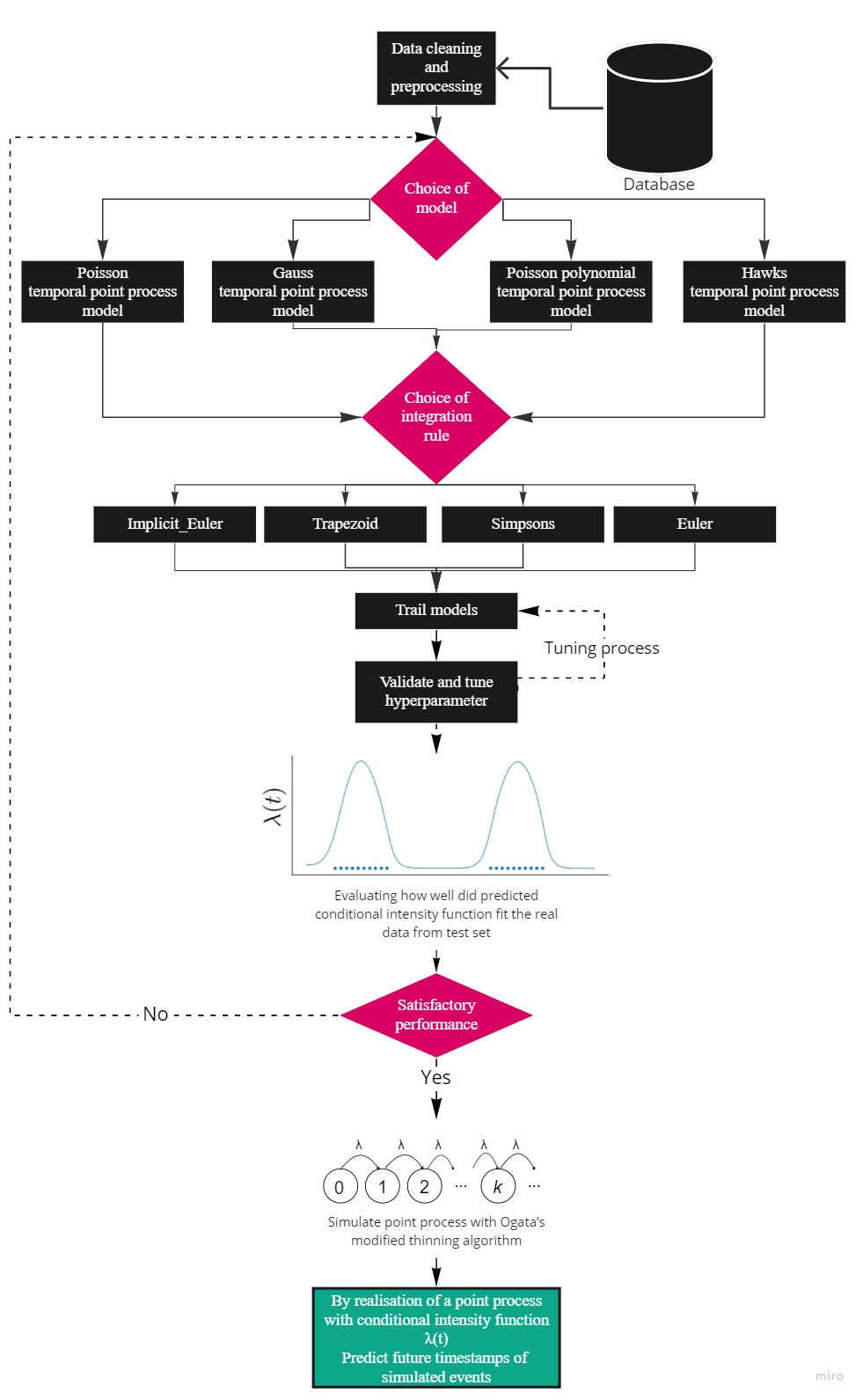


Figure 2. Framework for learning point processes

**5. EXPERIMENTS**

**5.1 EXPERIMENTAL SETUP**

To obtain results reported in Section 5.2. by our base framework architecture presented in Section 4. and depicted in Figure 2 in the experiment we predefined training – test splits of our datasets that are used as a benchmark and made sure to train all model long enough to ensure convergence.

**Datasets.** The presented methodology was tested on two different datasets: a high frequency events dataset, traffic prediction on highway toll dataset, and on low frequency events dataset, prediction of ski injuries in ski resort Kopaonik. Therefore, in experiments we provided the methodology performance to learn event generation from two different types of datasets.

In the case of high frequency events dataset, the sequence of cars arriving at the ramp toll on the E 75 highway was taken as a concrete example of interest. Highway European route E 75 is part of the International E-road network. The observed part connects two large Serbian cities, Belgrade and Niš. More precisely, the goal was to model the process of arrivals on the busiest ramp toll located at Niš from the Belgrade direction. The average time between two passes in one day is about 20 seconds, with a caveat that the time between two passes is highly dependent on time of the day that is observed. Standard 70/10/20 train, validation, test splits were chosen respectively.

As for the low frequency point processes dataset, the observations of ski injuries in ski resort Kopaonik were taken as a concrete events of interest. Ski resort Kopanik is biggest ski resort in Serbia. Dataset consists of records of ski injuries in period from 2005 to 2020. Training and validation were done on period before 2020 and the test and evaluation were done for the year 2020.

**Training.** Due to heterogeneous nature of our benchmark datasets, both in the number of samples and frequency of the events, it was observed that we can get better results by fine-tuning the number of epochs (training time) and framework architecture (base model selection, integration rule, integration step) independently for both datasets. To chose hyperparameters (integration step, learning rate, etc) we split the training data into 80%-20% split.

**Models.** Four different well known point process models were compared, Poisson temporal point process, Gaussian point process, Poisson polynomial process and Hawkes process with three different kinds of integration rules: Implicit Euler, Trapezoidal, Simpsons. All defined models are implemented in Pytorch, an optimized tensor library for deep learning using GPUs and CPUs implemented in Python (Paszke et al., 2019, p. 8026).

All models and integration rules were trained on entire training data, and evaluated on test data. Set of models that performed the overall best were Gaussian point process model and Hawks model which results are presented in Table 1 and Table 2 along with integration rules that were found to yield best performance for respective datasets.

**Optimization.** The Adam optimizer was used in order to fit the parameters of point processes.After hyperparameter tuning it was showed that, all models should be trained by 200 epochs, with a constant learning rate of 0.001, and integration step of 30. Each model was trained with purpose of minimizing negative log likelihood, in order to reconstruct the true underlying event generation process.

**Evaluation metrics.** Each of the models were evaluated on two key benchmark tasks, first, how well did the models fit real conditional intensity functions on the test set, or in other words how well did the models perform minimization of negative log likelihood loss. Secondly, using Ogata’s modified thinning algorithm and conditional intensity function learned during training, we evaluated how well did the models predict future time events. Therefore, predicted time events were separated into bins of 3 different sizes: 5 minutes, 10 minutes, 15 minutes for highway toll dataset and bins of 5 days, 10 days, 15 days for ski injuries dataset, respectively. Then for each binned period mean absolute error (MAE) was calculated between the number of predicted events and real number of events in that period. The visualization of sliding window approach for point process performance evaluation is presented in Fig. 3. In the case of high frequency dataset, due to dense event generation process, events were slide by 1 minute (*σ(t) = 1* min), whereas in the case of low frequency dataset events were slide by 1 day period (*σ(t) = 1* day).

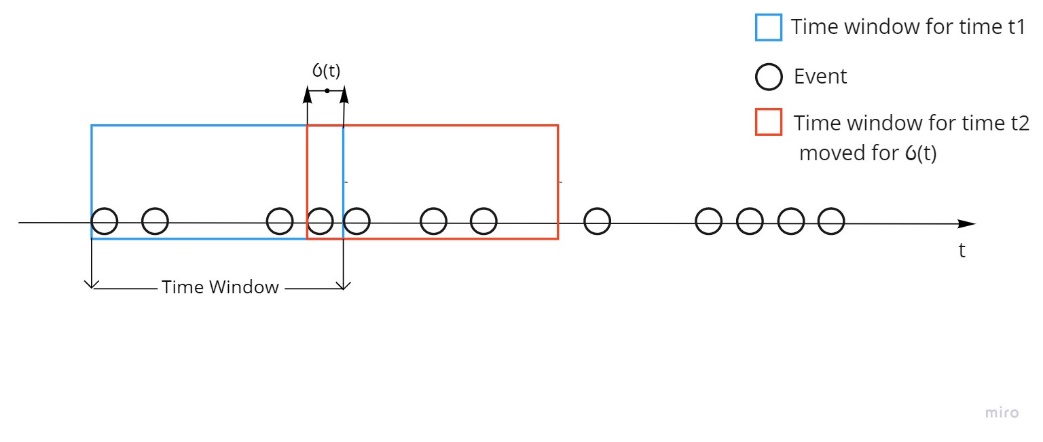


Figure 3. Sliding window algorithm

**5.2 RESULTS AND DISCUSSION**

The prediction performances obtained by fitting different types of point process models with three distinct integration methods on highway car arrivals dataset are presented in Table 1.

Hawkes model with trapezoid numerical integration techniques, had smallest log likelihood loss and the smallest MAE obtained in the case of all bin size. Furthermore, despite small training data and stochastic nature of data generation (i.e., dependency on part of day) it can be concluded that on average, the error of Hawkes model is less than half car per minute compared to the real car arrivals events. In addition, compared to the Hawkes process, Polynomial process obtained the worst results, whereas the results of Poission process are satisfactory, bearing in mind that conditional intensity function is constant.

The results obtained by Hawkes model on highway car arrivals dataset are visualized in Fig 4. Firstly, it can be observed that real and predicted conditional intensity functions are plotted with blue and green lines, respectively, by varying length of sliding window (Fig. 4a – 1 min, Fig. 4b – 5 min, Fig. 4c – 10 min). Additionally, the real and simulated timestamps of car arrivals were visualized as red dots. It can be concluded that despite model being trained on just 70% of data, model was pretty successful in predicting the real conditional intensity function. Moreover, the simulated car arrival events can in the same manner imitate the real world application.

In Table 2 results of models performances on ski injuries dataset are presented. In the same manner, four different point process models performances with respect to three different numerical integration methods are showed. Gaussian point process model with Implicit Euler numerical integration techniques, had smallest log likelihood loss and MAE.

The Polynomial process again obtained the worst results. The Gaussian point process achieved on average MAE of 1.6 in the period of 5 days, which means that the Gaussian point process is going to predict on average 1.6 injuries more or less compared to the real number of injuries. Based on this, it can be emphasized that it is necessary to fit different point process models in order to find the one that best explain the true underlying distribution of event generation.

In addition, the results obtained by Gaussian point process on ski injuries dataset are visualized in Fig 5. Firstly, it can be observed that real and predicted conditional intensity functions are plotted with blue and green lines, respectively, by varying length of sliding window (Fig. 5a – 5 days, Fig. 5b – 10 days, Fig. 5c – 15 days). Additionally, the real and simulated timestamps of ski injuries were visualized as red dots. It can be observed that real and predicted conditional intensity functions looks very similar, and timestamps of simulated events corresponds to the timestamps of real events.

Table 1. Results of models performances

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bin\_size | Model | Integration\_method | MAE | Log likelihood loss (test\_set) |
| 5 | Hawkes | **Trapezoid** | **4.9** | **112.56** |
| Implicit\_Euler | 5.6 | 116.03 |
| Simpson | 5.9 | 126.78 |
| 10 | **Trapezoid** | **8.8** | **112.56** |
| Implicit\_Euler | 10.3 | 116.03 |
| Simpson | 9.7 | 126.78 |
| 15 | **Trapezoid** | **12.3** | **112.56** |
| Implicit\_Euler | 14.07 | 116.03 |
| Simpson | 14 | 126.78 |
| 5 | Gaussian PP | Trapezoid | 6 | 178.85 |
| Implicit\_Euler | 5.9 | 216.44 |
| Simpson | 5.7 | 156.25 |
| 10 | Trapezoid | 11.5 | 178.85 |
| Implicit\_Euler | 11.3 | 216.44 |
| Simpson | 11.1 | 156.25 |
| 15 | Trapezoid | 17.2 | 178.85 |
| Implicit\_Euler | 16.6 | 216.44 |
| Simpson | 15.4 | 156.25 |
| 5 | Polynomial | Trapezoid | 30.8 | 670.65 |
| Implicit\_Euler | 63.4 | 756.04 |
| Simpson | 364 | 1206.34 |
| 10 | Trapezoid | 61.7 | 670.65 |
| Implicit\_Euler | 185.3 | 756.04 |
| Simpson | 729.3 | 1206.34 |
| 15 | Trapezoid | 92.6 | 670.65 |
| Implicit\_Euler | 336 | 756.04 |
| Simpson | 1094.3 | 1206.34 |
| 5 | Poisson | - | 9.1 | 142.12 |
| 10 | Poisson | 17.7 | 142.12 |
| 15 | Poisson | 26.6 | 142.12 |

|  |
| --- |
| *(a)* |
| *(b)(c)* |

Figure 4. Visualization of how conditional intensity function predicted by Hawk’s point process model fitted the real intensity function

Table 2. Experimental results - ski injuries dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bin \_size | Model | Integration\_method | MAE | Log likelihood loss (test\_set) |
| 5 | Hawkes | Trapezoid | 3.2 | 76.20 |
| Implicit\_Euler | 3.3 | 78.81 |
| Simpson | 4 | 77.82 |
| 10 | Trapezoid | 4.8 | 76.20 |
| Implicit\_Euler | 6.2 | 78.81 |
| Simpson | 8 | 77.82 |
| 15 | Trapezoid | 7.6 | 76.20 |
| Implicit\_Euler | 10 | 78.81 |
| Simpson | 10 | 77.82 |
| 5 | Gaussian PP | Trapezoid | 3.8 | 99.64 |
| **Implicit\_Euler** | **1.6** | **75.42** |
| Simpson | 2.8 | 88.56 |
| 10 | Trapezoid | 5.8 | 99.64 |
| **Implicit\_Euler** | **3.8** | **75.42** |
| Simpson | 5.4 | 88.56 |
| 15 | Trapezoid | 9.6 | 99.64 |
| **Implicit\_Euler** | **4.9** | **75.42** |
| Simpson | 7.3 | 88.56 |
| 5 | Polynomial | Trapezoid | 5.1 | 68223.36 |
| Implicit\_Euler | 4.6 | 22480.22 |
| Simpson | 46 | 4262.03 |
| 10 | Trapezoid | 9.8 | 68223.36 |
| Implicit\_Euler | 6.6 | 22480.22 |
| Simpson | 102.4 | 4262.03 |
| 15 | Trapezoid | 14 | 68223.36 |
| Implicit\_Euler | 8 | 22480.22 |
| Simpson | 134 | 4262.03 |
| 5 | Poisson | - | 3.1 | 142.00 |
| 10 | Poisson | 6.4 | 142.00 |
| 15 | Poisson | 12.3 | 142.00 |

|  |
| --- |
| *(a)* |
| *(b)*  *(c)* |

Figure 5. Visualization of how conditional intensity function predicted by Gaussian point process model fitted the real intensity function

**6. Conclusion**

In this paper, we propose a new machine learning approach methodology for learning temporal point process based on the implementation of one-dimensional numerical integration techniques. The likelihood function of the point process has an integral of the CIF given in the limits of data observation. Bearing in mind that the CIF can take any kind of mathematical form, in many cases this integral is analytically intractable. Due to this, in this paper, we present an approach to linearise this integral with standard numerical techniques and to backpropagate the derivative through it. The presented approach was successfully tested on real-life data. The main disadvantage of this approach lies in high computational cost that is connected with backpropagation of derivative through each integration step. Therefore, this approach should be used only in cases when point processes with analytically tractable integrals cannot obtain satisfactory prediction performances.

Furthermore, the methodology was evaluated on four different well-known point process models. In addition, we presented that different numerical techniques for integration can be successfully implemented in this framework. Moreover, we successfully simulated the obtained CIFs and compared them with the observed intensity functions.

Further studies should address using deep neural networks (feed-forward and recurrent networks) as a CIF to better capture dependencies between event occurrences.

**ACKNOWLEDGEMENT**

This work was supported in part by the ONR/ONR Global under Grant N62909-19-1-2008. The authors would like to express gratitude to company Saga NFG d.o.o Belgrade, for supporting this research. The authors would also like to thank public enterprises Roads of Serbia and Ski resorts of Serbia for providing data for this research.

**REFERENCES**

1. Ryu, B., & Steven, B. L. (1998). Point process models for self-similar network traffic, with applications. Communications in statistics. Stochastic models, 14(3), 735-761.

2. Zahrieh, D. (2017). Bayesian point process modeling to quantify excess risk in spatial epidemiology: an analysis of stillbirths with a maternal contextual effect.

3. Liu, S., & Wu, W. (2017). Generalized Mahalanobis depth in point process and its application in neural coding. The Annals of Applied Statistics, 11(2), 992-1010.

4. Farajtabar, M., Wang, Y., Gomez-Rodriguez, M., Li, S., Zha, H., & Song, L. (2017). Coevolve: A joint point process model for information diffusion and network evolution. The Journal of Machine Learning Research, 18(1), 1305-1353.

5. Jacobsen, S., Grove, H., Nedenskov Jensen, K., Sørensen, H. A., Jessen, F., Hollung, K., ... & Søndergaard, I. (2007). Multivariate analysis of 2‐DE protein patterns–Practical approaches. Electrophoresis, 28(8), 1289-1299.

6. Bowsher, C. G. (2007). Modelling security market events in continuous time: Intensity based, multivariate point process models. Journal of Econometrics, 141(2), 876-912.

7. Mei, H., & Eisner, J. M. (2017). The neural hawkes process: A neurally self-modulating multivariate point process. In Advances in Neural Information Processing Systems (pp. 6754-6764).

8. Jia, R., Jiang, P., Liu, L., Cui, L., & Shi, Y. (2018). Data driven congestion trends prediction of urban transportation. IEEE Internet of Things Journal, 5(2), 581-591.

9. Nguyen, T. T., Krishnakumari, P., Calvert, S. C., Vu, H. L., & Van Lint, H. (2019). Feature extraction and clustering analysis of highway congestion. Transportation Research Part C: Emerging Technologies, 100, 238-258.

10. Rasmussen, J. G. (2011). Temporal point processes: the conditional intensity function. Lecture Notes, Jan.

11. Last, G., & Penrose, M. (2017). Lectures on the Poisson process (Vol. 7). Cambridge University Press.

12. Kirchner, M. (2017). An estimation procedure for the Hawkes process. Quantitative Finance, 17(4), 571-595.

13. Xiao, S., Yan, J., Yang, X., Zha, H., & Chu, S. M. (2017, February). Modeling the intensity function of point process via recurrent neural networks. In Thirty-first aaai conference on artificial intelligence.

14. Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. In Advances in neural information processing systems (pp. 6571-6583).

15. Zhang, T., Yao, Z., Gholami, A., Gonzalez, J. E., Keutzer, K., Mahoney, M. W., & Biros, G. (2019). ANODEV2: A Coupled Neural ODE Framework. In Advances in Neural Information Processing Systems (pp. 5151-5161).

16. Ghahramani, Z. (2003, February). Unsupervised learning. In Summer School on Machine Learning (pp. 72-112). Springer, Berlin, Heidelberg.

17. Mehrasa, N., Jyothi, A. A., Durand, T., He, J., Sigal, L., & Mori, G. (2019). A variational auto-encoder model for stochastic point processes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3165-3174).

18. Yuan, B., Wang, X., Ma, J., Zhou, C., Bertozzi, A. L., & Yang, H. (2019, September). Variational Autoencoders for Highly Multivariate Spatial Point Processes Intensities. In International Conference on Learning Representations.

19. Xiao, S., Farajtabar, M., Ye, X., Yan, J., Song, L., & Zha, H. (2017). Wasserstein learning of deep generative point process models. In Advances in Neural Information Processing Systems (pp. 3247-3257).

20. Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. Biometrika, 58(1), 83-90.

21. Xu, H., Farajtabar, M., & Zha, H. (2016, June). Learning granger causality for hawkes processes. In International conference on machine learning (pp. 1717-1726).

22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Desmaison, A. (2019). Pytorch: An imperative style, high-performance deep learning library. In Advances in neural information processing systems (pp. 8026-8037).

23. Palm, C. (1988). Intensity variations in telephone traffic. North-Holland.

24. Radunović, D, Numeričke metode (2004). Akademska misao, Beograd.

25. Euler, L. (1768). Institutionum calculi integralis, vol. 1. imp. Acad. imp. Saent.