Inférence Statistique de Processus de Hawkes et Apprentissage Profond

Nicolas Girard[1]

[1] *M2 IASD, Université Paris Dauphine-PSL, 75016 Paris**

1er septembre 2023

CentraleSupélec

Département : MIDO
Président du Jury : Tristan Cazenave
Encadrant : Ioane Muni-Toke

*Electronic address : nicolas.durat@dauphine.eu

Remerciements

En préambule de ce mémoire, je souhaiterais sincèrement remercier les personnes qui m'ont aidé et soutenu pendant son déroulement ainsi que pour leur suivi constant, qui m'a été très important. Monsieur Ioane Muni-Toke, mon tuteur, a su m'orienter et me suivre dans l'élaboration de ce mémoire en me confiant des missions de qualité. Je le remercie pour son suivi et son investissement personnel dans la réussite de mon stage et de mes projets. Son apport organisationnel se révélera à coup sûr stratégique pour la suite de ma carrière. L'ensemble de mon équipe au sein du laboratoire MICS qui m'a conseillé et permis de mieux comprendre le contexte du monde de la recherche, ainsi que l'importance d'une bonne entente et cohésion d'équipe. Monsieur Tristan Cazenave, mon référent et professeur pour son écoute mais également pour ses conseils et son implication dans l'élaboration et la réalisation de ce mémoire. Son suivi m'a permis de m'ajuster aux exigences attendues pour la réussite de mon année. Je souhaiterais terminer en remerciant mon entreprise d'accueil, CentraleSupélec, qui m'a permis de travailler et de partager leur vision. Cela aura été une expérience enrichissante et valorisante que de travailler avec des personnes qualifiées qui m'ont apporté des connaissances et des méthodes qui me seront utiles durant toute ma vie professionnelle. Je remercie également ma famille et mes proches pour leurs soutiens. Ils m'ont permis de garder ma motivation ainsi qu'une implication totale dans ce mémoire.

Résumé

Les processus de Hawkes sont des processus ponctuels auto-excités très utiles pour la modélisation spatio-temporelle d'événements dans de multiples domaines comme la sismologie, l'écologie ou la finance. Récemment, un fort intérêt est apparu pour l'estimation par apprentissage automatique de ces processus à partir de données de comptage dites « agrégées ». L'objectif de cette thèse est d'étudier différents modèles d'apprentissage profond pour l'estimation de processus de Hawkes agrégés afin de les comparer aux modèles de l'état de l'art. Dans un premier temps, des processus de Hawkes sont calibrés, simulés et discrétisés. Une modélisation de l'intensité de base, du ratio d'endogénéité et de l'intensité conditionnelle des processus agrégés est réalisée à l'aide d'un modèle génératif probabiliste basé sur une distribution de Poisson. Ensuite, différents types de réseaux de neurones sont utilisés pour estimer ces mêmes paramètres. Enfin, les méthodes appliquées sont testées sur un ensemble de paramètres prédéfini et étendu à des processus de Hawkes agrégés plus complexes. Les modèles développés permettent d'obtenir des résultats similaires aux méthodes de l'état de l'art tout en réduisant le temps de calcul.

Keywords : Processus de Hawkes, Apprentissage Profond, RNN, Modèle Génératif Probabilistes

Abstract

Hawkes processes are self-exciting point processes that are very useful for spatio-temporal modeling of events in many fields, such as seismology, ecology or finance. Recently, there has been a strong interest in machine-learning estimation of these processes from so-called "aggregated" count data. The aim of this thesis is to study different deep learning models for the estimation of aggregated Hawkes processes, and to compare them with state-of-the-art models. First, Hawkes processes are calibrated, simulated and discretized. The baseline intensity, endogeneity ratio and conditional intensity of the aggregated processes are modelled using a probabilistic generative model based on a Poisson distribution. Next, different types of neural networks are used to estimate these same parameters. Finally, the applied methods are tested on a predefined set of parameters and extended to more complex aggregated Hawkes processes. The models developed achieve similar results to state-of-the-art methods, while reducing computation time.

Keywords : Hawkes process, Deep Learning, RNN, Probabilistic Generative Model

Table des matières

Table des figures

1 Introduction

Les processus de Hawkes constituent une classe particulière de processus stochastique introduit par Alan G. Hawkes en 1971 [1] et sont appliqués à différents domaines : la sismologie [2], la biologie [3], ou la finance quantitative [4]. Ce sont des processus ponctuels auto-excités où l'occurrence d'un événement augmente temporairement la probabilité que d'autres événements se produisent. Cependant, selon les contraintes des disciplines étudiées les positions exactes des points ne sont pas observés et limitent leur modélisation. Pour cette raison, le comptage d'événements à intervalles réguliers plutôt qu'à un moment précis est courant et favorise l'émergence de techniques d'estimation de processus de Hawkes dit « agrégés ». L'objectif de cette thèse est d'appliquer les méthodes d'apprentissage profond pour estimer les paramètres des processus de Hawkes agrégés. Le problème est abordé de deux façons : une approche discriminante utilisant des réseaux de neurones (MLP, LSTM) et une approche probabiliste exploitant les auto-encodeurs variationnels (VAE) basés sur une distribution de Poisson. Les deux approches permettent d'estimer l'intensité de base et le ratio d'endogénéité des processus agrégés. Les définitions des processus de Hawkes agrégés et des modèles d'apprentissage profond sont données afin de comprendre leurs mécanismes sous-jaçents. Un rappel des notions de bases en statistique et l'étude bibliographique des processus agrégés et non-agrégés sont présentés. Ensuite, la méthodologie décrite dans cette étude expose de manière concise les résultats obtenus et engage une discussion sur les performances.

2 Contexte

2.1 Processus de Hawkes

2.1.1 Processus ponctuels

Un processus ponctuel est une collection aléatoire de points tombant dans un certain espace. Dans la plupart des applications, chaque point représente le moment et/ou le lieu d'un événement. Parmi les exemples d'événements, on peut citer l'observation ou la naissance d'une espèce, de tremblements de terre ou d'éruptions volcaniques. Lors de la modélisation de données purement temporelles, l'espace dans lequel les points tombent est simplement une portion de la ligne réelle. Les processus ponctuels spatio-temporels sont souvent utilisés pour décrire les processus environnementaux ; dans ce cas, chaque point représente le moment et l'emplacement d'un événement dans une région spatio-temporelle. Les réalisations d'un processus ponctuel sont généralement représentées de trois manières :

— Une série temporelle d'événements : $\{t_i\}_{i=1,\dots,n}$;
— Une série de temps d'attente : $\{\tau_i\}_{i=1,\dots,n}$ où $\tau_i = t_i - t_{i-1}$ et $t_0 = 0$;
— Une série de comptages d'événements sur l'intervalle : $(0,t]; \{N(t)\}_{t\geq 0}$.

Definition 1 (Processus de comptage)**.** Un processus de comptage est un processus stochastique ponctuel, $\{N(t)\}_{t\geq 0}$ qui possède les propriétés suivantes [5] :

— $N(t) \geq 0$;
— $N(t) \in \mathbb{Z}$;
— $\forall t \leq s, N(t) \leq N(s)$.

En outre, un processus de comptage peut être représenté par un nombre choisi d'incréments. Un incrément est la valeur du processus de comptage sur des sous-ensembles disjoints de l'intervalle. Ce processus est représenté par le nombre souhaité d'incréments n, d'une longueur $\Delta = \frac{T}{n}$. Les incréments peuvent être définis comme suit :

$$N_t^{(\Delta)} = N(\Delta t) - N(\Delta(t-1)), \quad t = 1, ..., n. \tag{2.1}$$

Les processus ponctuels sont souvent observés comme des incréments d'un processus de comptage lorsque l'observation des temps d'événements est compromise. Cette observation peut être entravée par les limites technologiques ou les coûts d'observation et reste courante lorsque les événements se produisent sur de très petites échelles de temps comme en cybersécurité [6]. En conséquence, il y a une perte d'information inhérente et directement reliée à la taille de l'incrément Δ. Lorsque $\Delta = 0$, le processus agrégés retourne le processus ponctuel sous-jacent. L'exemple le plus connu de processus de comptage est le processus de Poisson. La propriété fondamentale d'un processus de Poisson est que le nombre d'événements sur un intervalle de longueur t, suit une variable aléatoire de Poisson avec un taux λt. Cette propriété est une hypothèse qui s'exprime par :

$$\mathbb{P}\big(N(t) = n\big) = e^{-\lambda t}\frac{(\lambda t)^n}{n!}. \tag{2.2}$$

L'équation (2.2) définit un processus de Poisson homogène en raison de son paramètre de taux constant, $\lambda > 0$. Ce taux correspond à l'intensité du processus de Poisson. Ce qui donne :

Definition 2 (Processus de Poisson homogène)**.** Un processus de comptage est un processus de Poisson homogène si les propriétés suivantes sont respectées [5] :

- $N(0) = 0$;
- $N(t)$ a des incréments indépendants ;
- Le nombre d'événements suit une variable aléatoire de Poisson avec une intensité λt.

Dans de nombreux cas, l'hypothèse d'un processus de Poisson homogène est déraisonnable. Cela débouche sur un processus de Poisson inhomogène dont l'intensité variable est définie par une fonction du temps $\lambda(t)$. Pour un processus de Poisson inhomogène, le nombre d'événements sur un intervalle $(0, t)$, suit une variable aléatoire de Poisson avec un taux égal à l'intégrale de la fonction d'intensité sur l'intervalle : $\int_0^t \lambda(s)\,\mathrm{d}s$. Par substitution :

$$\mathbb{P}\big(N(t) = n\big) = e^{-\int_0^t \lambda(s)\,\mathrm{d}s}\frac{(\int_0^t \lambda(s)\,\mathrm{d}s)^n}{n!}. \tag{2.3}$$

Definition 3 (Processus de Poisson inhomogène)**.** Un processus de comptage est un processus de Poisson inhomogène si les propriétés suivantes sont respectées [5] :

- $\lambda(t)$ est une fonction intégrable ;
- $N(0) = 0$;
- $N(t)$ a des incréments indépendants ;
- $\forall t \in [0, \infty)$, $\mathbb{P}\big(N(t+\Delta) - N(t) = 0\big) = 1 - \lambda(t)\Delta + o(\Delta)$;
- $\forall t \in [0, \infty)$, $\mathbb{P}\big(N(t+\Delta) - N(t) = 1\big) = \lambda(t)\Delta + o(\Delta)$;
- $\forall t \in [0, \infty)$, $\mathbb{P}\big(N(t+\Delta) - N(t) \geq 2\big) = o(\Delta)$.

2.1.2 Processus ponctuels auto-excités

Le processus de Hawkes est un processus ponctuel auto-excité. Lorsqu'un point apparaît à un instant t alors la probabilité d'apparition d'un autre point augmente. Par conséquent, il permet de modéliser des clusters de points aléatoires dans le temps et l'espace. Les processus de Hawkes sont caractérisés par leur intensité conditionnelle qui correspond à la probabilité d'une arrivée pendant un intervalle de temps conditionné par l'historique du processus $\mathcal{H}_t$. Cet historique représente toutes les arrivées jusqu'au temps t exclu. L'intensité conditionnelle peut s'écrire en fonction de la combinaison de la densité conditionnelle $f(t|\mathcal{H}_t)$ et de la distribution conditionnelle $F(t|\mathcal{H}_t)$.

Definition 4 (Intensité conditionnelle)**.** Considérons un processus de comptage, $N(t)$, avec un historique, $\mathcal{H}_\sqcup$. Sachant la densité conditionnelle, $f(t|\mathcal{H}_t)$, et la distribution conditionnelle, $F(t|\mathcal{H}_t)$, la fonction d'intensité conditionnelle du processus ponctuel est définie comme suit [5] :

$$\lambda^*(t) = \frac{f(t|\mathcal{H}_t)}{1 - F(t|\mathcal{H}_t)}, \tag{2.4}$$

$$\lambda^*(t) = \lim_{\Delta \to a} \frac{\mathbb{E}[N(t+\Delta) - N(t)|\mathcal{H}_t]}{\Delta}. \tag{2.5}$$

$\lambda^*(t)$ est une fonction non négative qui définit l'intensité du processus ponctuel sur la base de l'historique des points observés jusqu'à aujourd'hui. De façon non-exhaustive, les processus ponctuels peuvent être auto-excités ou auto-correctifs. La première catégorie concerne un processus dont l'intensité augmente en cas d'arrivée et la seconde concerne un processus dont l'intensité diminue en cas d'arrivée.

Definition 5 (Processus de Hawkes)**.** Considérons un processus de comptage, $N(t)$, auquel sont associés un historique $\mathcal{H}_t$, et une fonction d'intensité conditionnelle, $\lambda^*(t)$. On suppose que pour tout $t \in [0, \infty)$ la règle suivante s'applique [5] :

- $\forall t \in [0, \infty)$, $\mathbb{P}\big(N(t+\Delta) - N(t) = 0\big) = 1 - \lambda(t)\Delta + o(\Delta)$;
- $\forall t \in [0, \infty)$, $\mathbb{P}\big(N(t+\Delta) - N(t) = 1\big) = \lambda(t)\Delta + o(\Delta)$;
- $\forall t \in [0, \infty)$, $\mathbb{P}\big(N(t+\Delta) - N(t) \geq 2\big) = o(\Delta)$.

Si la fonction d'intensité conditionnelle, $\lambda^*(t)$, est de la forme :

$$\lambda^*(t) = \mu + \int_0^t k(t-u)\mathrm{d}N(u), \tag{2.6}$$

où $\mu \geq 0$ et $k : (0, \infty) \to [0, \infty)$, alors $N(t)$ est un processus de Hawkes.

La forme de la fonction d'intensité conditionnelle est sa caractéristique principale. Elle est divisée en deux termes. Le premier, μ, est l'intensité de base, et le second $\mathrm{k}(\star)$, est la fonction d'excitation. L'intensité de base est une contribution constante à l'intensité conditionnelle, indépendamment des événements passés. La fonction d'excitation représente la contribution relative des événements passés à l'intensité conditionnelle. Elle prend souvent la forme d'un noyau invariant. Lorsque la fonction d'excitation est nulle, on obtient une fonction d'intensité conditionnelle constante, équivalente à un processus de Poisson homogène. Ainsi, le processus de Hawkes est vu comme un processus d'immigration et de naissance. Les points sont classés comme immigrants ou descendants. Les immigrants sont des événements attribués à l'intensité de base. Les descendants se produisent grâce à la propriété d'auto-excitation, ils sont considérés comme des naissances. La forme de la fonction noyau définit les propriétés du processus de Hawkes. La forme de base est exponentielle :

$$k(t-u) = \alpha e^{(-\beta(t-u))}. \tag{2.7}$$

La compréhension des propriétés d'auto-excitation qui sont impliquées par le choix de la fonction noyau est un domaine d'étude important. Dans les sections suivantes, le rapport de branchement ou ratio d'endogénéité, η, est particulièrement intéressant. Il est défini comme le nombre attendu de descendants de la première génération causé par un événement.

Il est égal à l'intégrale de la fonction noyau sur l'intervalle $[0, \infty)$ qui est la suivante :

$$\int_0^\infty \alpha e^{(-\beta(s))\mathrm{d}s} = \frac{\alpha}{\beta} = \eta. \tag{2.8}$$

Pour la simulation des processus de Hawkes, il est nécessaire que $\eta < 1$. La raison est évidente si l'on considère le nombre total attendu de descendants d'un événement. On a $\frac{\eta}{1-\eta}$ quand $\eta < 1$ et ∞ lorsque $\eta \geq 1$ [7]. Par la suite, on suppose que η est inférieur à 1. Parallèlement à la définition du nombre de descendants attendus, le nombre total d'événements attendus est :

$$\mathbb{E}(\lambda^*(t)) = \frac{\mu}{1-\eta}. \tag{2.9}$$

2.2 Modèles discriminants

2.2.1 Multilayer perceptron (MLP)

Les réseaux neuronaux sont la représentation non linéaire $F(X)$ sur un espace d'entrée à haute dimension à l'aide de couches hiérarchiques d'abstractions [8]. Un exemple de réseau neuronal est un réseau feedforward - une séquence de L couches formées via des compositions.

Definition 6 (Réseau Feedforward)**.** Un réseau feedforward est une fonction de la forme :

$$\hat{Y}(X) := F_{W,b}(X) = \left(f^{(L)}_{W^{(L)},b^{(L)}} \dots \circ f^{(1)}_{W^{(1)},b^{(1)}}\right)(X) \tag{2.10}$$

où :

- $f^{(l)}_{W^{(l)},b^{(l)}}(X) := \sigma_l(W_l X + b_l)$ est une fonction semi-affine ;
- σ_l est une fonction d'activation non linéaire continue comme $\max(\star, 0)$ ou $\tanh(\star)$;
- $W = (W^{(1)}, ..., W^L)$ and $b = (b^1, .., b^{(L)})$ sont des matrices de poids et des biais.

Sur la figure ci-dessous, on observe un exemple d'un réseau feedforward avec deux couches cachées, deux valeurs d'entrée et trois valeurs de sortie. Généralement, les classificateurs de réseau d'apprentissage profond ont plus de couches et utilisent un plus grand nombre d'entrées et de sorties.

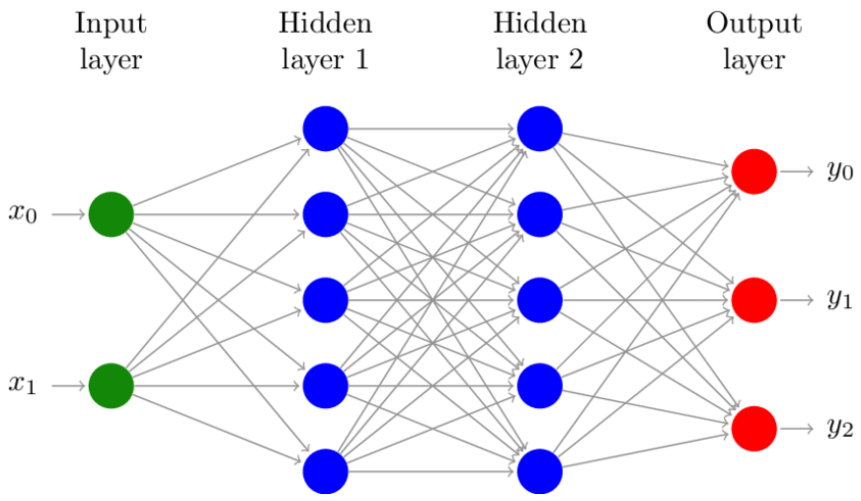

FIGURE 1 – MLP Architecture - Source : [9]

Chaque neurone d'une couche est connecté à chaque neurone de la couche précédente. En effet, on utilise une combinaison linéaire des valeurs des neurones qui lui sont connectés. Après l'évaluation de la combinaison linéaire de chaque couche, une transformation est appliquée. Cette transformation est appelée fonction d'activation. Si toutes les fonctions d'activations sont linéaires, $F_{W,b}$ est juste une régression linéaire peu importe le nombre de couches L et le nombre de couches cachées. De manière informelle, le principal effet de la fonction d'activation est d'introduire de la non-linéarité dans le modèle et en particulier dans les termes d'interactions entre les entrées. Les réseaux neuronaux utilisent la minimisation d'une fonction de perte pour apporter des modifications itératives aux poids et aux biais. Cela permet au réseau d'apprendre la fonction qu'il essaie de reproduire. Ce processus est connu sous le nom de rétropropagation qui est un algorithme d'optimisation basé sur le gradient. Ce dernier est calculé de manière récursive sur chaque couche, ce qui permet de calculer le gradient en remontant le réseau.

2.2.2 Long-Short-Term-Memory (LSTM)

Le modèle LSTM est un réseau de neurones récurrents conçu pour surmonter les problèmes d'explosion et de disparition du gradient lors de l'apprentissage des dépendances à long terme. Globalement, ce problème peut être évité en utilisant un carrousel d'erreurs constantes (CEC), qui maintient le signal d'erreur dans la cellule de chaque unité [11]. Ces cellules sont elles-mêmes des réseaux récurrents, dont l'architecture intéressante est que le CEC est étendu avec des caractéristiques supplémentaires, à savoir la porte d'entrée et la porte de sortie, formant ainsi la cellule mémoire. Les connexions autorécurrentes indiquent un retour d'information avec un décalage d'un pas de temps.

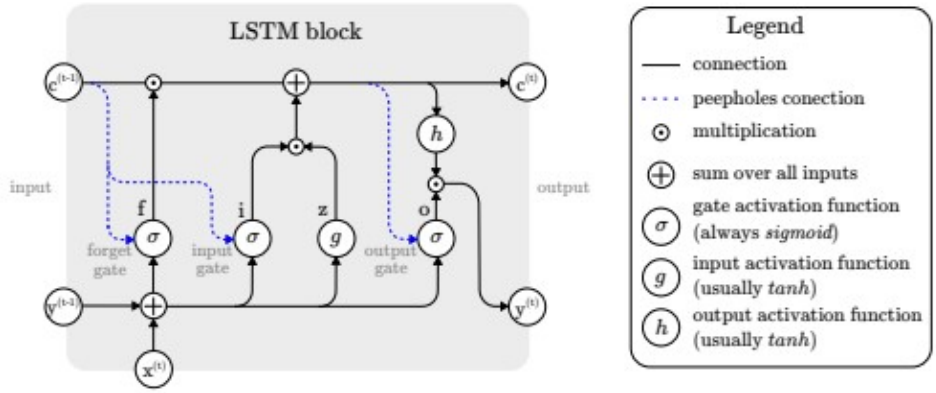

FIGURE 2 – LSTM Architecture - Source : [11]

Afin de clarifier le fonctionnement du LSTM, on suppose un réseau de N blocs et de M entrées. La forward pass est décrite selon 6 composantes : block input, input gate, forget gate, cell memory gate, output gate, block output. D'abord, l'entrée du bloc, qui combine l'entrée actuelle $x(t)$ et la sortie de cette unité LSTM $y^{(t-1)}$ lors de la dernière itération, est mise à jour. L'équation en jeu s'écrit [11] :

$$z^{(t)} = g(W_z x^{(t)} + R_z y^{(t-1)} + b_z), \tag{2.11}$$

où W_z et R_z sont les poids associés à $x^{(t)}$ et à $y^{(t-1)}$, respectivement, tandis que b_z représente le biais. Ensuite, la porte d'entrée qui combine l'entrée actuelle $x^{(t)}$ et la sortie de cette unité LSTM $y^{(t-1)}$ et la valeur de la cellule $c^{(t-1)}$ lors de la dernière itération est mise à jour. La procédure est la suivante [11] :

$$i^{(t)} = \sigma(W_i x^{(t)} + R_i y^{(t-1)} + p_i \odot c^{(t-1)} + b_i), \tag{2.12}$$

où $\odot$ désigne la multiplication de deux vecteurs, W_i, R_i et p_i sont les poids associés à $x^{(t)}$, $y^{(t-1)}$ et $c^{(t-1)}$, tandis que b_i représente le vecteur de biais associé à cette composante. Dans les étapes précédentes, la

couche LSTM détermine quelles informations doivent être conservées dans les états cellulaires du réseau $c^{(t)}$. Cela inclut la sélection des valeurs $z^{(t)}$ qui peuvent potentiellement être ajoutées aux états des cellules, et les valeurs d'activation $i^{(t)}$ des portes d'entrée. Au niveau de la porte d'oubli, l'unité LSTM détermine quelles informations doivent être supprimées de ses états cellulaires précédents $c^{(t-1)}$. Par conséquent, les valeurs d'activation $f^{(t)}$ des portes d'oubli au pas de temps t sont calculées sur la base de l'entrée actuelle $x^{(t)}$ les sorties $y^{(t-1)}$ et de l'état $c^{(t-1)}$ des cellules de mémoire au pas de temps précédent $(t-1)$. Cela peut s'exprimer de la manière suivante [11] :

$$f^{(t)} = \sigma(W_f x^{(t)} + R_f y^{(t-1)} + p_f \odot c^{(t-1)} + b_f), \tag{2.13}$$

où W_f, R_f et p_f sont les poids associés à $x^{(t)}$, $y^{(t-1)}$ et $c^{(t-1)}$, tandis que b_f désigne le biais. L'étape de la cellule mémoire, permet de calculer la valeur de la cellule, qui combine l'entrée du bloc $z^{(t)}$, la porte d'entrée $i^{(t)}$ et la porte d'oubli $f^{(t)}$ avec la valeur de la cellule précédente. Cette opération peut être réalisée comme indiqué ci-dessous [11] :

$$c^{(t)} = z^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)}. \tag{2.14}$$

Après coup, la porte de sortie qui combine l'entrée actuelle $x^{(t)}$, la sortie de cette unité LSTM $y^{(t-1)}$ et la valeur de la cellule $c^{(t-1)}$ lors de la dernière itération est calculée. Elle se traduit par [11] :

$$o(t) = \sigma(W_o x^{(t)} + R_o y^{(t-1)} + p_o \odot c^{(t)} + b_o), \tag{2.15}$$

où W_o, R_o et p_o sont les poids associés à $x^{(t)}$, $y^{(t-1)}$ et $c^{(t-1)}$, tandis que b_o désigne le biais. Enfin, on calcule la sortie du bloc, qui combine la valeur actuelle de la cellule $c(t)$ et la valeur actuelle de la porte de sortie comme suit [11] :

$$y^{(t)} = g(c^{(t)}) \odot o^{(t)}. \tag{2.16}$$

La sigmoïde logistique $\sigma(x) = \frac{1}{1+e^{1-x}}$ est utilisée comme fonction d'activation, tandis que la tangente hyperbolique $g(x) = h(x) = \tanh(x)$ est la fonction d'activation de l'entrée et de la sortie du bloc.

2.3 Modèles probabilistes

2.3.1 Auto-encodeur variationnel (VAE)

L'auto-encodeur est un réseau de neurones qui apprend une fonction d'identité de manière non supervisée afin de reconstruire l'entrée originale tout en comprimant les données [10]. Le but est d'obtenir une représentation plus efficace et comprimée. Il se compose de deux réseaux : le réseau encodeur qui traduit l'entrée de haute dimension en une représentation latente de basse dimension. La taille de l'entrée est supérieure à celle de la sortie. Le réseau décodeur récupère les résultats de la représentation latente avec des couches de sortie de plus en plus grandes pour reconstruire les données d'entrée. Un auto-encodeur variationnel est un modèle génératif basé sur la vraisemblance. L'inférence est effectuée via inférence bayésienne variationnelle pour approximer la distribution postérieur du modèle. Au lieu de convertir l'entrée en un vecteur fixe, on la convertit en une distribution. Cette distribution $p_\theta(\mathbf{z})$ est paramétrée par θ. La relation entre les données d'entrée et le vecteur d'encodage latent est entièrement définie par la distribution préalable $p_\theta(\mathbf{z})$, la vraisemblance $p_\theta(\mathbf{x}|\mathbf{z})$, la distribution postérieure $p_\theta(\mathbf{z}|\mathbf{x})$. Afin de générer un échantillon qui ressemble à un point de données réel $\mathbf{x}^{(i)}$ on peut suivre les étapes suivantes :

— Échantillonner $\mathbf{z}^{(i)}$ à partir d'une distribution préalable
— Une valeur $\mathbf{x}^{(i)}$ est générée à partir d'une distribution conditionnelle $p_{\theta^*}(\mathbf{x}|\mathbf{z} = \mathbf{z}^{(i)})$

Le paramètre optimal θ^* maximise la probabilité de générer des échantillons de données réelles :

$$\theta^* = \arg\max_\theta \prod_{i=1}^{n} p_\theta(\mathbf{x}^{(i)}). \tag{2.17}$$

Ensuite, on actualise l'équation afin d'illustrer la génération des données et l'encodage :

$$p_\theta(\mathbf{x}^{(i)}) = \int p_\theta(\mathbf{x}^{(i)}|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z}. \tag{2.18}$$

Il n'est pas facile de calculer $p_\theta(\mathbf{x}^{(i)})$ car il est très coûteux de vérifier toutes les valeurs possibles de $\mathbf{z}$ et de les additionner. Pour réduire l'espace des valeurs afin de faciliter une recherche plus rapide, on introduit une nouvelle fonction d'approximation pour déterminer quel est le code probable compte tenu d'une entrée $\mathbf{x}$, $q_\phi(\mathbf{z}|\mathbf{x})$ paramétrée par ϕ.

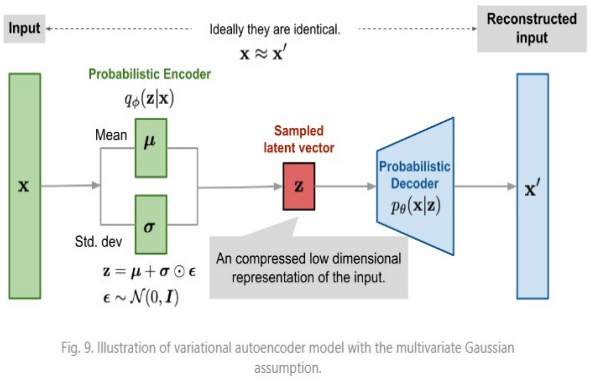

FIGURE 3 – Architecture VAE - Source : [13]

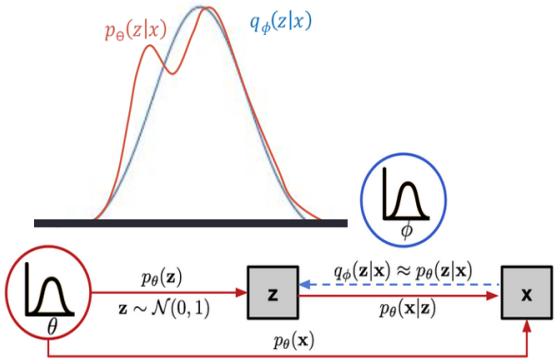

FIGURE 4 – Approximation VAE - Source : [13]

La structure ressemble à un auto-encodeur :

— $p_\theta(\mathbf{x}|\mathbf{z})$ définit un modèle génératif, similaire au décodeur probabiliste.
— $q_\phi(\mathbf{z}|\mathbf{x})$ est l'encodeur probabiliste, jouant un rôle similaire de $g_\phi(\mathbf{z}|\mathbf{x})$.

La distribution postérieur estimée $q_\phi(\mathbf{z}|\mathbf{x})$ doit être très proche de la valeur réelle. On utilise la divergence de Kullback-Leibler pour quantifier la distance entre ces deux distributions. La divergence de KL mesure la quantité d'informations perdues si la distribution Y est utilisée pour représenter X. Dans ce cas, on veut minimiser $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})p_\theta(\mathbf{z}|\mathbf{x}))$ par rapport ϕ. Pour les méthodes bayésiennes variationnelles, la fonction de perte utilisée est connue sous le nom de limite inférieure variationnelle ou de limite inférieure d'évidence (ELBO) [12]. La partie borne inférieure du nom provient du fait que la divergence KL est toujours non négative et donc que $-L_{\mathrm{VAE}}$ est la borne inférieure de $\log p_\theta(\mathbf{x})$:

$$-L_{\mathrm{VAE}} = \log p_\theta(\mathbf{x}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) \leq \log p_\theta(\mathbf{x}) \tag{2.19}$$

En minimisant la perte, on maximise la limite inférieure de la probabilité de générer des échantillons de données réelles. Le terme d'espérance dans la fonction de perte fait appel à la génération d'échantillons à partir de $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$. L'échantillonnage est un processus stochastique et on ne peut pas rétropropager le gradient. Pour le rendre entraînable, il est possible d'exprimer la variable aléatoire comme une variable déterministe $\mathbf{z} = \mathcal{T}_\phi(\mathbf{x}, \boldsymbol{\epsilon})$ où $\boldsymbol{\epsilon}$ est une variable aléatoire indépendante auxiliaire, et la fonction de transformation $\mathcal{T}_\phi$ paramétrée par ϕ convertit $\boldsymbol{\epsilon}$ en $\mathbf{z}$. Par exemple, un choix courant de la forme de $q_\phi(\mathbf{z}|\mathbf{x})$ est une gaussienne multivariée avec une structure de covariance diagonale :

$$\begin{aligned} &\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\boldsymbol{I}) \\ &\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \text{ où } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}) \end{aligned} \tag{2.20}$$

L'astuce du reparamétrage fonctionne pour d'autres types de distributions, et pas seulement pour les distributions gaussiennes. Dans le cas de la distribution gaussienne multivariée, le modèle est entraînable en apprenant la moyenne et la variance de la distribution, μ et σ, explicitement à l'aide de l'astuce de reparamétrage, tandis que la stochasticité demeure dans la variable aléatoire $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$.

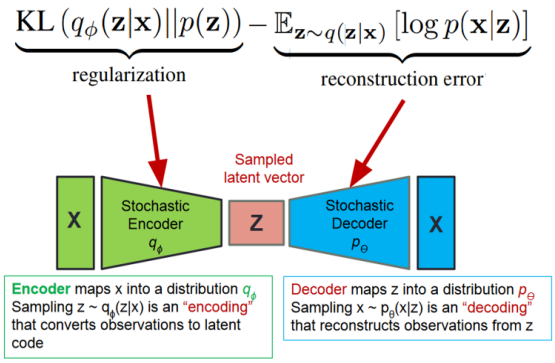

FIGURE 5 – VAE Connexion - Source : [13]

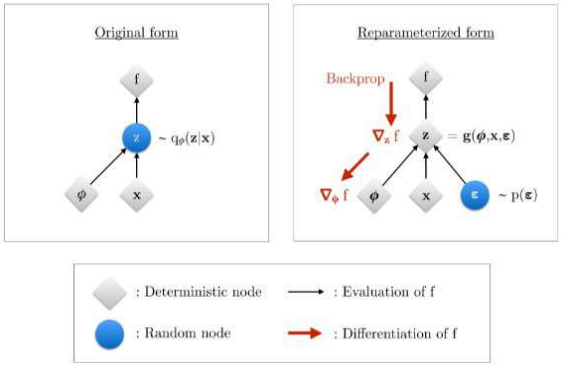

FIGURE 6 – Reparamétrage - Source : [13]

3 État de l'art

Un processus de Hawkes est défini par la forme de sa fonction d'intensité conditionnelle. Une estimation précise des paramètres de cette fonction est intéressante car elle permet de comprendre la fréquence des événements. Elle permet notamment de comprendre l'effet des influences et des événements endogènes (naissances) et exogènes (immigrants). L'estimation des paramètres est une tâche difficile lorsque les temps d'événements d'un processus sont disponibles. Elle est encore plus difficile lorsque les temps d'événements ne sont pas disponibles. Dans ce cas, des informations sont perdues en raison d'une observation intermittente (discrétisation) ou d'une observation imprécise (arrondi). Cette revue de littérature couvre les méthodes appliquables sur des données continues et des données discrètes. Les données continues se présentent sous la forme de temps d'événements précis et les données discrètes sous la forme d'un processus de comptage incrémental à intervalles d'observation réguliers.

3.1 Estimation des paramètres en temps continu

Des méthodes sont accessibles pour les données continues, avec de nombreuses techniques basées sur l'estimation non-paramétrique des processus de Hawkes. Ces méthodes non paramétriques sont flexibles car elles permettent d'estimer la fonction d'intensité conditionnelle sans aucune restriction sur sa forme.

3.1.1 Estimation paramétrique

L'estimation du maximum de vraisemblance est la norme pour l'estimation des paramètres des processus de Hawkes avec une gamme de noyaux paramétriques [14]. Elle a été décrite pour la première fois par Ozaki [15]. La log-vraisemblance a été donnée en termes de fonction noyau générale par Rubin [16]. La log-vraisemblance générale est définie par :

Definition 7 (Log-vraisemblance & Processus ponctuel)**.** Soit un processus de Hawkes avec des temps d'observation, $\{t_i\}_{i=1,\dots,n}$ pour un intervalle $[0, T]$, la log-vraisemblance d'un processus de Hawkes avec une fonction d'intensité conditionnelle, $\lambda^*(t)$, est donnée par :

$$l(t_1, ..., t_n) = -\int_0^T \lambda^*(s)\mathrm{d}s + \int_0^T \log\lambda^*(s)\mathrm{d}N(s) \tag{3.1}$$

Grâce à l'équation (3.1), les gradients et la matrice hessienne de la log-vraisemblance sont dérivés [15]. Ensuite, l'optimisation non linéaire a été réalisée en utilisant la méthode de Newton-Raphson. Cette version du MLE est quadratique en matière de temps de calcul. Une méthode récursive pour le noyau exponentiel, qui est linéaire en temps de calcul, a été dérivée [15] et présentée en détail par Laub [7]. Une méthode récursive n'est disponible que pour le noyau exponentiel, ce qui signifie que cette accélération n'existe pas pour d'autres choix de noyaux. Avec $A(i) = e^{(-\beta(t_i - t_i - 1))}(1 + A(i-1))$, la méthode s'écrit :

$$l(t_1, ..., t_n) = \sum_{i=1}^{n} \log(\mu + \alpha A(i)) - \mu t_n + \frac{\alpha}{\beta} \sum_{i=1}^{n} e^{-\beta(t_n - t_i))-1} \tag{3.2}$$

Un algorithme paramétrique de maximisation des espérances (EM) a été étudié pour le noyau exponentiel [17] en suivant une méthode dérivée de Veen et Schoenberg [18]. Cette méthode repose sur une étape d'espérance pour calculer la probabilité qu'un événement i soit un descendant de l'événement j ou un immigrant. Cette étape est suivie d'une étape de maximisation qui met à jour les estimations actuelles de la fonction d'intensité conditionnelle. Un autre algorithme EM paramétrique est décrit par Olson et Carley [19]. Aucune comparaison entre la performance de ces méthodes ou leur performance par rapport à l'estimateur du maximum de vraisemblance n'est disponible. Il convient toutefois de noter que l'estimation du maximum de vraisemblance semble être privilégiée pour les estimations paramétriques.

3.1.2 Estimation non-paramétrique

L'estimation non paramétrique pour les processus de Hawkes englobe les méthodes qui ne supposent pas une forme de la fonction noyau dans la fonction d'intensité conditionnelle. Dans la section précédente, les méthodes paramétriques supposent que la forme du noyau est connue. L'absence d'hypothèse signifie que les méthodes non paramétriques sont plus flexibles, mais cela se fait au détriment de la précision. Si la forme du noyau est connue, il est généralement préférable d'utiliser une méthode paramétrique.

Une première méthode non paramétrique sous la forme d'un algorithme EM du nom de Model Independent Stochastic Declustering (MISD) est proposée par Marsan et Lengline [20]. Cette méthode est dérivée pour des intensités de base homogènes et des noyaux non paramétriques. Elle est similaire à celle de Veen et Schoenberg [18], car elle comprend l'estimation de la probabilité qu'un événement i soit un descendant de l'événement j ou un immigrant. L'étape de maximisation nécessite l'estimation constante de l'intensité de base et une estimation constante par morceaux de la fonction noyau non paramétrique est calculée. L'opération est répétée jusqu'à convergence. Un autre algorithme EM non paramétrique a été développé [17]. Il utilise une méthode de vraisemblance pénalisée et a été conçu pour estimer un processus de Hawkes avec une base variable en conjonction avec la fonction de noyau non paramétrique. La technique ajoute un terme de pénalité à la fois à la fonction optimisée pour la variation de l'intensité de base et le noyau non paramétrique. Ces termes de pénalité permettent de reformuler le problème de minimisation sous la forme d'une équation différentielle ordinaire qui est discrétisée et résolue de manière itérative. La flexibilité de l'approche la rend intéressante lorsque l'on ne dispose d'aucune information sur la forme de l'intensité de base ou de la fonction noyau.

Une méthode non paramétrique pour les processus de Hawkes avec des matrices à noyau symétrique est proposée par Bacry et al. [14]. Une méthode ultérieure ne reposant pas sur la symétrie de la matrice à noyau est exposée par Bacry et Muzy [21] et étendue [23]. La première méthode consiste à relier la matrice d'autocovariance à un décalage donné à la matrice du noyau par le biais d'une transformée de Fourier. La méthode nécessite le choix de deux hyperparamètres pour l'estimation de la matrice de covariance. Les hyperparamètres sont le décalage et l'échelle de la matrice de covariance. Le décalage se rapporte au décalage de l'auto-covariance et l'échelle est la discrétisation de la fonction noyau. Dans [14], il est suggéré que cette méthode est fiable pour les séries de plus de 105 événements et qu'elle peut être utilisée comme une première exploration de la forme du noyau avant de passer à une méthode paramétrique telle que la MLE. Les extensions apportées ont permis d'abandonner l'exigence de noyau symétrique [22]. Cette méthode utilise la matrice de l'espérance conditionnelle, qui peut être estimée empiriquement selon les résultats [14], et nécessite l'utilisation d'une méthode de Nystrom sur les équations résultantes.

3.2 Estimation des paramètres en temps discret

Cette section présente les méthodologies qui peuvent être appliquées au problème de l'estimation des paramètres des processus de Hawkes agrégés. On retrouve l'estimation non paramétrique à l'aide de séries chronologiques auto-régressives, le maximum de vraisemblance par intervalle et les simulations MC-EM.

3.2.1 Séries chronologiques autorégressives

Dans [24], la convergence faible d'une série auto-régressive à valeur entière (INAR) et retards infinis vers un processus de Hawkes a été démontrée. Elle conduit à une méthode non paramétrique qui repose sur la discrétisation du processus de Hawkes en petits intervalles Δ [25]. Le terme auto-régressif est approximé à l'aide de p retards auto-régressifs. La discrétisation est effectuée sur les réalisations d'un processus de Hawkes, mais il en résulte une méthode qui est utilisée sur des données discrètes. L'approximation s'écrit :

$$\mathbb{E}(N_i^{(\Delta)}|\mathcal{H}_{i-1}^{(\Delta)}) \approx \Delta\mu + \sum_{n=1}^{p} \Delta k(\Delta n) N_{i-n}^{\Delta}, \tag{3.3}$$

où $\mathcal{H}_{i-1}^{(\Delta)}$ représente l'historique des comptages incrémentaux jusqu'à l'intervalle $i-1$. À chaque approximation correspond une source d'erreur documentée dans [25]. La plus grande source d'erreur provient de la première approximation, où la fonction noyau est supposée être constante par morceaux. Ainsi, les événements qui se produisent dans la même intervalle ne peuvent pas être conditionnels les uns aux autres. On ne peut pas dire qu'un événement dans une intervalle déclenche un autre événement. Pour estimer les paramètres du processus INAR(p), la méthode des moindres carrés conditionnels est utilisée, ce qui donne le processus d'estimation suivant pour $\{N_i^{(\Delta)}\}_{i=1,\dots,n}$:

1. Choisir un grand p, avec $p < n$;
2. Calculer les estimateurs des moindres carrés conditionnels ;
3. Récupérer les p premières entrées et dernières entrées pour obtenir les estimations ;
4. Choisir un type de lissage pour l'estimation de la fonction noyau.

L'effet de la valeur de p et de Δ est discuté dans [25]. L'étude suggère que le plus grand nombre dans le processus discrétisé soit égal à 1. En temps continu, cela peut être contrôlé par la valeur de Δ. Ce choix est un compromis biais-variance où un Δ plus petit favorise le biais mais aussi un compromis biais-temps de calcul, car le processus d'estimation repose sur une inversion de matrice. Dans le cas de données discrètes, ce choix n'est pas possible. Il convient de noter qu'une méthode semi-paramétrique a été suggérée. Toutefois, Kirchner et Bercher ont montré qu'elle est moins performante même lorsque le modèle paramétrique est égal au noyau d'intensité sous-jacent [26] . Les performances de cette méthode sont comparées au MLE qui est plus performant. Cela est dû à l'élément non paramétrique de la méthode. Le principal avantage est le temps de calcul linéaire par rapport au temps du MLE qui est quadratique.

3.2.2 Maximum de vraisemblance par intervalles

Cette méthode développée par Shlomovich et al. [27] est l'application de l'estimation du maximum de vraisemblance pour des données discrètes. La fonction d'intensité conditionnelle est désignée comme $\lambda^{(\Delta)}(i) \equiv \lambda^{(\Delta)}(i\Delta|\mathcal{H}_{i-1}^{(\Delta)})$. Cela conduit à l'approximation de log-vraisemblance suivante :

$$l = \sum_{i=1}^{n} N_i^{(\Delta)} \log[\Delta\lambda^{(\Delta)}(i)] - \Delta\lambda^{(\Delta)}(i) \tag{3.4}$$

L'hypothèse d'une fonction d'intensité conditionnelle constante par morceaux équivaut à supposer que $N_i^{(\Delta)} \sim \text{Poisson}(\Delta\lambda^{(\Delta)})$. Cette méthode présente les mêmes limites que la méthode INAR car les événements à l'intérieur d'une intervalle ne peuvent pas déclencher d'événements. Pour estimer les paramètres, la log-vraisemblance a été maximisé à l'aide d'une optimisation sous contrainte.

3.2.3 Monte Carlo - Expectation Maximisation

La méthode principale dans Shlomovich et al. [28] était un algorithme de maximisation des espérances de Monte Carlo (MC-EM) pour les processus de Hawkes. Un algorithme MC-EM est utilisé lorsque l'espérance de la distribution postérieure est analytiquement intraitable. Cela peut-être résolu en utilisant l'intégration de Monte Carlo. Cependant, l'intégration de Monte Carlo nécessite un échantillon de la distribution postérieure qui n'est pas disponible. Il utilise l'échantillonnage par importance pour tirer un ensemble de points temporels sachant le processus observé. Cette nouvelle méthode est appliquée de manière séquentielle. Ainsi, en maximisant la PDF tronquée conjointe d'événements sachant l'historique, il est possible de générer un échantillon qui maximise la vraisemblance. Ces échantillons sont utilisés pour approximer l'étape d'espérance à l'aide de l'échantillonnage d'importance, puis pour passer à l'étape de maximisation. Cette méthode est plus performante que la méthode INAR et la méthode de vraisemblance découpée en intervalle. Une analyse relative à la taille de la discrétisation montre que l'algorithme MC-EM a le biais le plus faible pour toutes les tailles de pas de 0.2 à 2.

3.3 Estimation des paramètres par apprentissage automatique

3.3.1 Approche discriminante

Simon et al. [29] utilisent les réseaux neuronaux récurrents (RNN) combinés à un réseau feedforward pour modéliser l'intensité conditionnelle de processus ponctuels sur des données sismiques. Trois types de processus ponctuels sont testés : processus de Poisson homogène, processus auto-excité et auto-correctif. Concernant l'évaluation, la RMSE est utilisée entre l'intensité évaluée à chaque nouvelle arrivée et la valeur effective. Le processus de Poisson homogène obtient le meilleur score suivi du processus auto-correctif et auto-excité. Ensuite, il propose de comparer le RNN à un modèle paramétrique dont les paramètres ont été estimés par MLE. La MAE augmente jusqu'à environ 4 heures avec un biais positif ce qui indique que les deux modèles sous-estiment l'heure d'arrivée du prochain séisme de 4 heures en moyenne. Ainsi, le prochain séisme se produit en moyenne 4 heures après l'heure prévue. Le RNN obtient des résultats similaires à ceux du modèle paramétrique. Cependant, alors que le modèle paramétrique nécessite l'historique complet des événements, le RNN n'a été entraîné que sur les 10 derniers événements sismiques. Il n'a fallu que quelques secondes au RNN pour fournir la séquence de prédiction de l'arrivée suivante, alors que le modèle paramétrique nécessite 15 minutes. Aucune forme de fonction d'intensité spécifique n'est requise dans le RNN et les ressources informatiques modestes méritent d'être mentionnées.

Un type particulier de RNN a été appliqué aux processus de Hawkes par K. Lee [30]. En effet, il compare les résultats du modèle long short-term memory (LSTM) par rapport au MLE sur des données financières synthétiques et empiriques. Les LSTM sont utiles pour les tâches saisissant les dépendances à long terme, car le mécanisme de blocage permet au réseau de conserver les informations antérieures importantes de la séquence tout en rejetant les informations non pertinentes. Grâce à ce dernier, il estime les paramètres des processus de Hawkes. La capacité du modèle à faire des prédictions est évaluée grâce à l'erreur quadratique moyenne (MSE). Les résultats montrent que la MLE est légèrement plus performante, mais que le LSTM donne également des résultats raisonnables. Par ailleurs, la méthode numérique de la MLE nécessite de nombreuses itérations et prend beaucoup plus de temps que le LSTM. Pour comprendre ses propriétés, il étudie sa distribution d'échantillonnage. Dans l'ensemble, la méthode MLE est légèrement plus performante que le LSTM. Néanmoins, ses performances générales sont également très bonnes. Cette étude montre qu'un LSTM peut estimer de manière fiable les paramètres d'une série temporelle, avec une précision similaire à la méthode MLE et un calcul beaucoup plus rapide.

3.3.2 Approche probabiliste

Zhao et al. [31] ont décrit un VAE qui utilise une probabilité binomiale négative (NBVAE) pour modéliser des données textuelles présentées sous la forme de séquences de mots dans un document. Chaque mot possède un emplacement dans un vecteur. Une observation est alors représentée par un vecteur contenant des valeurs entières indiquant le nombre d'occurrences du mot selon sa position. Les données encodées sont similaires aux données agrégées d'un processus de Hawkes. Le NBVAE code l'auto-excitation et l'excitation croisée des mots dans un document. L'auto-excitation se réfère à l'occurrence d'un mot qui entraîne d'autres occurrences du même mot. L'excitation croisée se rapporte à l'excitation d'autres mots apparentés à l'occurrence d'un mot. La propriété auto-excitante correspond à l'auto-excitation produite dans une intervalle et l'auto-excitation croisée représente l'auto-excitation générée au sein d'un intervalle ultérieur. Ainsi, le NBVAE peut encoder l'auto-excitation de processus de Hawkes.

Un cadre d'inférence bayésienne a été décrit par Mishra et al. [32] qui suggère qu'étant donné un nouveau point de données y, et un VAE entièrement entraîné sur les données $\mathcal{D}$, un échantillon de la distribution postérieure $z|y$, peut être créé à l'aide d'un échantillonnage MCMC. En échantillonnant à partir de cette distribution, le décodeur convertis l'échantillon en un échantillon de la distribution postérieure prédite $\hat{y}|y$. Cela génère de nouveaux échantillons de processus de Hawkes. Cette méthode est appliquée dans le cadre d'un double décodeur développé par Seybold et al. [33]. Ce cadre repose sur deux décodeurs avec des sorties différentes utilisant la même variable latente. Cela a permis d'encoder dans l'espace latent des informations qui n'auraient pas été récupérées par un seul décodeur. Ainsi, il y a des spécifications distinctes de la perte de reconstruction pour les différents décodeurs. La méthodologie suggère que le second décodeur doit avoir un objectif différent de celui du premier décodeur, sinon cela équivaut simplement à repondérer la perte de reconstruction dans un VAE standard. Cette technique améliore l'inférence bayésienne avec un second décodeur qui reconstruit $\theta = \{\alpha, \beta, \mu\}$.

4 Méthode

La méthodologie utilisée pour générer les données s'inspire du travail de T. Keane [5] sur les réseaux de neurones (MLP) et les modèles génératifs (VAE) appliqués aux processus de Hawkes agrégés. Dans cette thèse, la même procédure est étendue aux réseaux de neurones récurrents (LSTM). La génération des données est décrite pour les entraînements et les tests des modèles. Ensuite, leurs architectures sont présentées. La partie consacrée aux modèles génératifs et leurs estimations est disponible en annexes.

4.1 Simulation des processus de Hawkes agrégés

4.1.1 Génération des données

Pour les données d'entraînement et des tests, des processus agrégés avec une fonction kernel exponentielle sont simulés sous la forme : $\{N_j^{(\Delta)}\}_{j=1,\dots,\frac{T}{\Delta}}$. Pour ce faire, 5 paramètres doivent être spécifiés :

- Le taux d'auto-excitation : α
- Le taux de décroissance : β
- L'intensité de base : μ
- L'horizon temporel : T
- La taille du pas de discrétisation : Δ

Sachant $\eta = \frac{\alpha}{\beta}$, on retrouve facilement α ou β s'il n'est pas spécifié. Dans cette thèse, la sélection de β et de η est réalisée. Pour agréger les processus, la longueur de l'intervalle d'observation Δ compte le nombre d'événements survenus dans chaque intervalle. Pour générer les données agrégées des modèles, il faut donc générer l'ensemble d'hyperparamètres $\{\beta, \eta, \mu\}$. Ces simulations sont réalisées avec la librairie Python « Hawkes » de Takahiro Omi [34] qui utilise la méthodologie d'Ogata [35]. La particularité de cette librairie est la forme de la fonction exponentielle qui permet d'obtenir $\alpha = \eta$ dans les calculs. Ainsi, la méthode utilisée pour générer n processus de Hawkes avec un niveau d'activité attendu E s'écrit :

1. Échantillonner une variable normale aléatoire $\epsilon \sim \mathcal{N}(E, \sigma^2) \quad \forall i = 1, ..., n$,
2. Échantillonner $\eta_i \sim \mathcal{U}(a, b)$ où $0 < a < b < 1$ sont fixés au préalable,
3. Calculer $\mu_i = \frac{\epsilon_i}{T}(1 - \eta_i)$,
4. Échantillonner $\beta_i \sim \mathcal{U}(p, q)$ où $0 < p < q$ sont fixés au préalable,
5. Calculer $\alpha_i = \eta_i$,
6. Générer les processus d'entrainement,
7. Discrétiser les temps d'événements à des intervalles de longueur Δ.

La spécification préalable pour β et η est une distribution uniforme, car elle ne fait qu'inférer les valeurs minimales et maximales à générer. Dans une application réelle, les valeurs de β et η sont inconnues. La spécification des paramètres doit idéalement contenir les vraies valeurs dans leur plage. L'utilisation des connaissances du domaine pour fixer les valeurs limites du nombre de descendants attendus est une bonne méthode. La plage de η peut être large, sans affecter négativement les performances. De surcroit, la spécification de β et η n'est intrinsèquement pas limitée à une distribution uniforme.

4.1.2 Génération des données d'entrainement

Pour l'entraînement, on suit la méthode et on l'applique à 100,000 processus d'entrainement :

Paramètres	Valeurs
Nombre de processus (N)	100,000
Taux de décroissance (β)	$\mathcal{U}(p = 1, q = 3)$
Taux de branchement (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Pas de discrétisation (Δ)	1
Écart type (σ)	10

TABLE 1 – Valeurs par défaut des paramètres d'entrainement

4.1.3 Génération des données de test

Pour les tests, on suit la méthode et on l'applique à 20,000 processus de tests :

Paramètres	Valeurs
Nombre de processus (N)	20,000
Taux de décroissance (β)	$\mathcal{U}(p = 1, q = 3)$
Taux de branchement (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Pas de discrétisation (Δ)	1
Écart type (σ)	10

TABLE 2 – Valeurs par défaut des paramètres de test

Dans les applications réelles, les hyperparamètres sous-jaçents qui ont généré les processus de Hawkes agrégés qui nous intéressent ne sont pas connus. Étant donné m processus de Hawkes agrégés, $\{y_i\}_{i=1,\dots,m}$, avec un horizon temporel T et un pas de discrétisation Δ, une estimation de l'activité attendue de ces processus doit être calculée. En supposant que les m processus sont générés par les mêmes paramètres, cela se fait en calculant la moyenne du nombre total d'événements observés pour chaque processus y_i :

$$\hat{y} = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{\frac{T}{\Delta}} (N_j^{(\Delta)})_i \tag{4.1}$$

Avec $E = \hat{y}$, on procède aux étapes 1 à 7 décrites plus haut pour générer les données d'apprentissage.

4.2 Estimation de l'intensité de base et du ratio d'endogéneité

Dans ce chapitre, les architectures du MLP/LSTM sont présentées. On prend l'hypothèse que les réseaux de neurones peuvent prédire η et μ comme une tâche de régression en prenant les données agrégées en entrée et les estimations en sortie. Au total, on entraîne dix réseaux de neurones : 5 MLP et 5 LSTM. Le nombre de modèle entraîné varie en fonction du nombre de pas de discrétisation Δ à tester.

4.2.1 Architecture du MLP

Le premier réseau de neurone est un MLP qui prend en données d'entrées les données agrégées et en données de sortie les estimations de η et μ. Sachant le jeu de données d'entrainement détaillé précédemment et la taille des batchs randomisés $B = N \times 0.1 = 10,000$, le MLP est entraîné sur 95% et validé sur 5% de ces données. L'architecture du modèle regroupe les paramètres suivants :

Paramètres	Valeurs
Dimension d'entrée	T / Δ
Dimension cachée	T / Δ
Dimension de sortie	2
Nombre de couches cachées	6
L2 Régularisation	0.001
Taux d'apprentissage	0.001
Nombre d'epochs	500
Early stopping patience	25
Early stopping δ	0.01

TABLE 3 – Paramètres du MLP

La couche d'entrée et la couche de sortie sont des couches linéaires. Les couches cachées sont des couches linéaires composées de fonction d'activation ReLU. Le réseau a été formé pour un maximum de 500 epochs, l'arrêt prématuré étant défini comme une réduction inférieure à 0.01 de la perte de validation sur 25 epochs. L'optimiseur est Adam et le critère optimisé est la MSELoss qui mesure l'erreur quadratique moyenne (norme L2 au carré). Une régularisation L2 est appliquée durant l'entraînement.

4.2.2 Architecture du LSTM

Le second réseau de neurones utilisé est un LSTM qui prend aussi en données d'entrées les données agrégées et en données de sortie les estimations de η et μ. Sachant le jeu de données d'entrainement détaillé précédemment et la taille des batchs randomisés $B = N \times 0.1 = 10,000$, le MLP est entraîné sur 95% et validé sur 5% de ces données. L'architecture du modèle regroupe les paramètres suivants :

Paramètres	Valeurs
Dimension d'entrée	T / Δ
Dimension cachée	64
Dimension de sortie	2
Nombre de couches	2
L2 Régularisation	0.001
Taux d'apprentissage	0.001
Nombre d'epochs	500
Early stopping patience	25
Early stopping δ	0.01

TABLE 4 – Paramètres du LSTM

La couche d'entrée est directement une couche LSTM avec une fonction d'activation sigmoïde interne et la couche de sortie est une couche linéaire. Le réseau a été formé pour un maximum de 500 epochs, l'arrêt prématuré étant défini comme une réduction inférieure à 0.01 de la perte de validation sur 25 epochs. L'optimiseur est Adam et le critère optimisé est la MSELoss qui mesure l'erreur quadratique moyenne (norme L2 au carré). Une régularisation L2 est appliquée durant l'entraînement.

5 Résultats

Les résultats sont segmentés en trois parties : l'évaluation de la méthode MLE, l'évaluation des méthodes d'apprentissage supervisé (MLP, LSTM), et leurs comparaisons respectives. L'évaluation de des modèles génératifs (Poisson-VAE, Poisson-VAE à double décodeur) est disponible en annexes.

5.1 Evaluation des méthodes de l'état de l'art

5.1.1 Méthodologie des tests

Le premier test compare l'erreur et l'erreur relative des estimations d'un MLE non-randomisé par rapport aux données réelles. Le but est de voir si les estimations et la dispersion des erreurs autour de la médiane sont initialement corrects. Le second test compare l'erreur et l'erreur relative des estimations d'un MLE randomisé par rapport aux paramètres Δ, E, η, β. L'objectif est d'analyser l'évolution des erreurs selon la valeur des paramètres. Les valeurs des paramètres par défaut sont :

Paramètres	Valeurs
Nombre de processus (N)	5,000
Taux de décroissance (β)	$\mathcal{U}(p = 1, q = 3)$
Taux de branchement (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Pas de discrétisation (Δ)	0
Écart type (σ)	10

TABLE 5 – Valeurs par défaut des paramètres du test n°1

Paramètres	Valeurs
Nombre de processus (N)	5,000
Taux de décroissance (β)	$\mathcal{U}(p = 1, q = 3)$
Taux de branchement (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Pas de discrétisation (Δ)	1
Écart type (σ)	10

TABLE 6 – Valeurs par défaut des paramètres du test n°2

Une gamme de valeurs a été sélectionnée pour tester chaque paramètre parmi Δ, E, β, η :

Taux de décroissance (β)	Taux de branchement (η)	Pas de discrétisation (Δ)	Niveau d'activité attendu (E)
$[0.5, 2.5]$	$[0.1, 0.4]$	0.25	50
$[1.75, 3.75]$	$[0.3, 0.6]$	0.5	100
$[3, 5]$	$[0.5, 0.8]$	1	250
$[0.5, 3]$	$[0.1, 0.6]$	2	500
$[1.5, 4]$	$[0.2, 0.7]$	5	1000
$[2.5, 5]$	$[0.3, 0.8]$		
$[0.5, 4]$	$[0.05, 0.6]$		
$[1.5, 5]$	$[0.05, 0.7]$		
$[0.5, 5]$	$[0.05, 0.8]$		

TABLE 7 – Valeurs variables des paramètres du test n°2

Pour obtenir les erreurs sachant le tableau ci-dessus, on teste le MLE pour chaque valeur des paramètres variables en prenant en compte le reste des valeurs par défaut des paramètres du test n°2.

5.1.2 Comparaison des erreurs du MLE

Dans cette partie sont présentés les erreurs et erreurs relatives des prédictions du MLE non-randomisé pour η et μ. Les distributions des erreurs sont décrites sous la forme de boîtes à moustache. La ligne horizontale rouge au milieu de la boîte représente la médiane des erreurs. Cela indique la valeur centrale de la distribution des erreurs et sépare l'ensemble des erreurs en deux parties égales. Les deux bords de la boîte représentent le premier quartile (Q1) et le troisième quartile (Q3). La distance entre Q1 et Q3 définit l'écart interquartile (IQR), qui représente la dispersion des erreurs autour de la médiane. La boîte englobe la région où se situe la majorité des erreurs (50 % des données) et donne une idée de la dispersion des erreurs autour de la médiane. Plus la boîte est large, plus la distribution des erreurs est étalée. Les lignes verticales qui s'étendent à partir de la boîte représentent les moustaches. Elles s'étendent jusqu'au minimum (ou 1.5 fois l'IQR en dessous de Q1) et jusqu'au maximum (ou 1.5 fois l'IQR au-dessus de Q3). Une majorité des résultats expliqués dans cette thèse sont affichés sans les valeurs aberrantes.

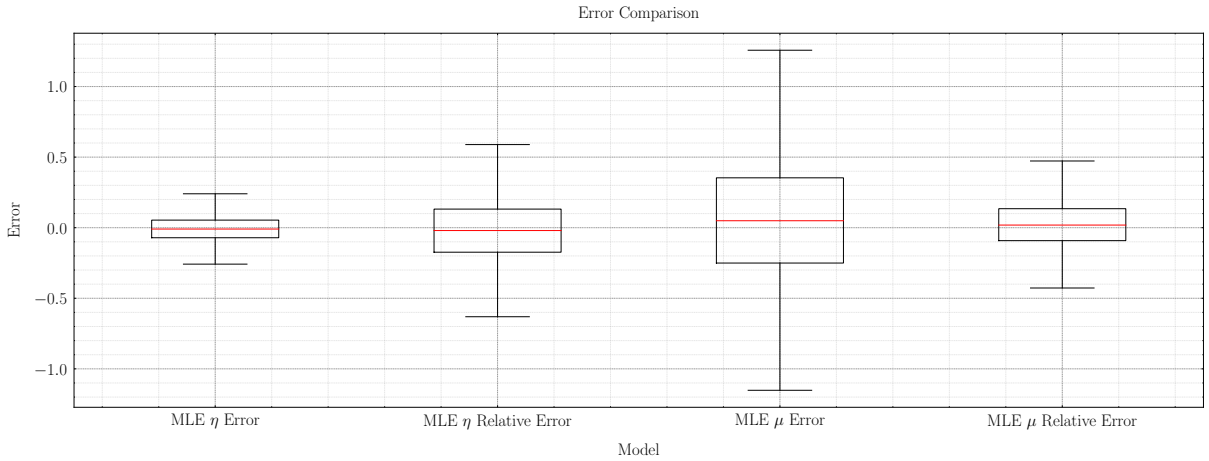

FIGURE 7 – Comparaison des erreurs - MLE ($\Delta = 0.0$)

Sur la figure 7, on observe la comparaison des erreurs du MLE non-randomisé pour les paramètres η et μ. On constate que toutes les erreurs sont centrées autour de 0 et que la distribution des erreurs pour η est moins étalée que pour μ. En moyenne les moustaches inférieures et supérieures sont évaluées autour de -0.5 et 0.5. Globalement cela prouve que les valeurs prédites sont proches des valeurs réelles.

5.1.3 Impacts des paramètres

Les résultats des erreurs du MLE randomisé en fonction des différents paramètres : Δ, E, η, β sont présentés dans cette sous-partie. Les erreurs et erreurs relatives de η et μ sont toujours évaluées par des boîtes à moustache. Concernant les paramètres : Δ, E, η, β, les valeurs testées sont données dans le tableaux 7. Dans l'ordre on retrouve la comparaison des erreurs en fonction de Δ, E, η et β.

Sur la figure 8, on constate que les erreurs et erreurs relatives sont de moins en moins concentrées autour de 0 et leurs dispersions augmentent positivement ou négativement en fonction de Δ. Les estimations sont plus précises lorsque $\Delta \to 0$ et inversement. Pour η, l'étalement négatif des erreurs lorsque $\Delta \to \infty$ implique que la prédiction est plus souvent inférieure à la valeur réelle. Pour μ, l'étalement positif des erreurs lorsque $\Delta \to \infty$ implique que la prédiction est plus souvent supérieure à la valeur réelle. Par conséquent, lorsque Δ croît, le nombre d'événement agrégé par intervalle d'observation augmente ce qui atténue la précision des estimations. Les données et les résultats du MLE perdent en granularité. Ainsi, on s'aperçoit que les erreurs d'estimation de η et de μ s'accentuent en fonction de l'augmentation de Δ.

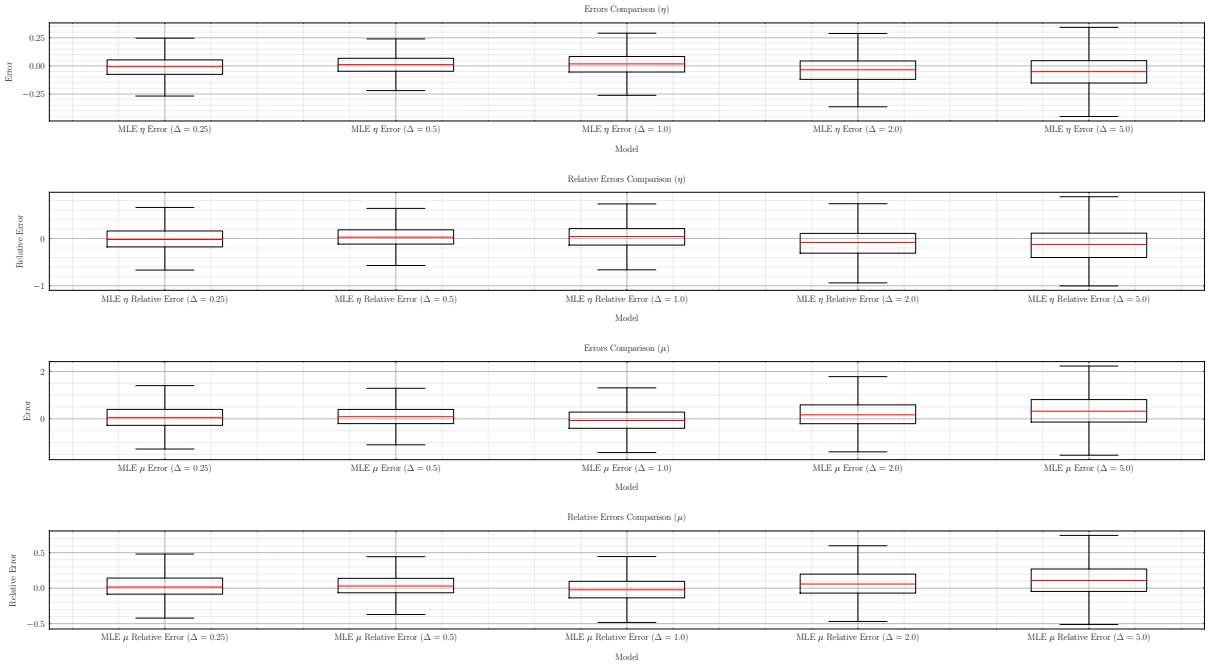

FIGURE 8 – Comparaison des erreurs en fonction de Δ - MLE

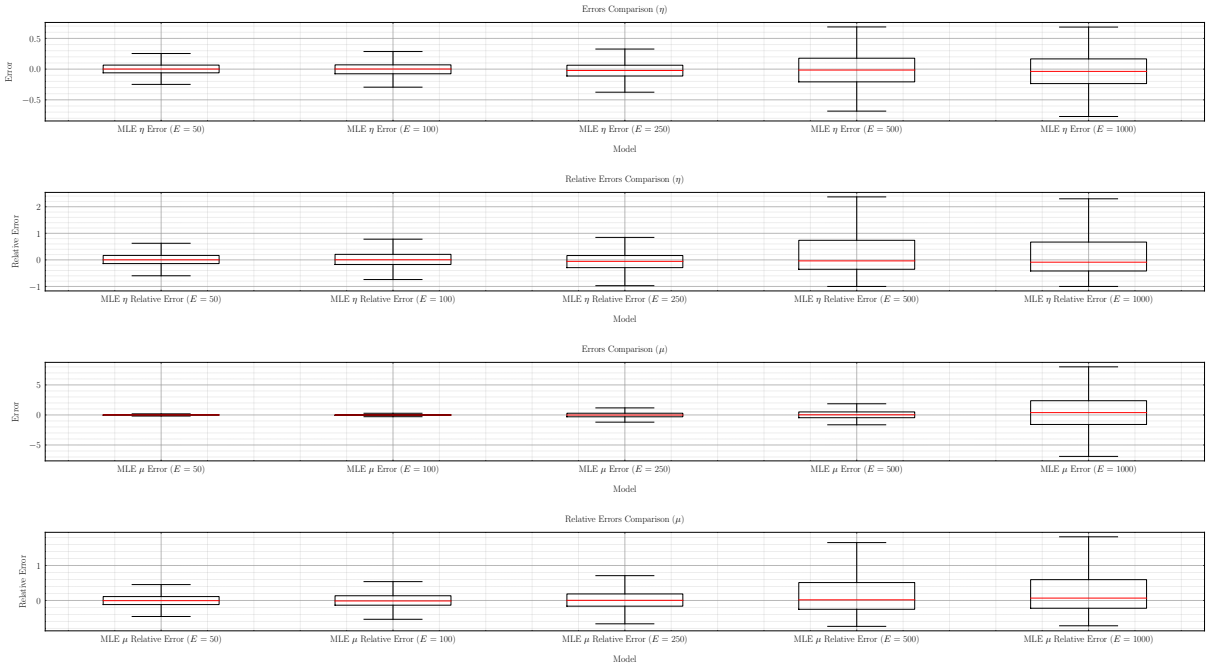

FIGURE 9 – Comparaison des erreurs en fonction de E - MLE ($\Delta = 1.0$)

Sur la figure 9, on compare les erreurs en fonction du niveau d'activité attendu E (nombre moyen d'événements par processus). On observe que les erreurs et erreurs relatives gravitent de moins en moins autour de 0 et leurs dispersions augmentent positivement ou négativement en fonction de E. En effet, les erreurs d'estimation de η et de μ augmentent lorsque $E \to \infty$ et inversement. Pour η, l'étalement négatif des erreurs lorsque $E \to \infty$ implique que la prédiction est plus souvent inférieure à la valeur réelle. Pour μ, l'étalement positif des erreurs lorsque $E \to \infty$ implique que la prédiction est plus souvent supérieure à la valeur réelle. Ainsi, lorsque le niveau d'activité attendu E augmente, l'intensité de base μ et le ratio de branchement η sont plus susceptibles de varier rendant leurs estimations moins précises. On conclut que les erreurs d'estimation de η et de μ s'amplifient en fonction de l'augmentation de E.

Sur la figure 10, on compare les erreurs en fonction de η. Les résultats tournent autour de 0 sans différences notables entre les intervalles. Cependant, les erreurs varient moins pour les intervalles $\eta \in [0.3, 0.8]$ et $\eta \in [0.5, 0.8]$. Ainsi, lorsque l'intervalle de η est restreinte et que $\eta \to 0.8$, les estimations de η et μ sont meilleures. On en déduit qu'une taille réduite et des valeurs précises pour l'intervalle de η donnent des estimations plus fiables selon le domaine d'expertise et le jeu de données.

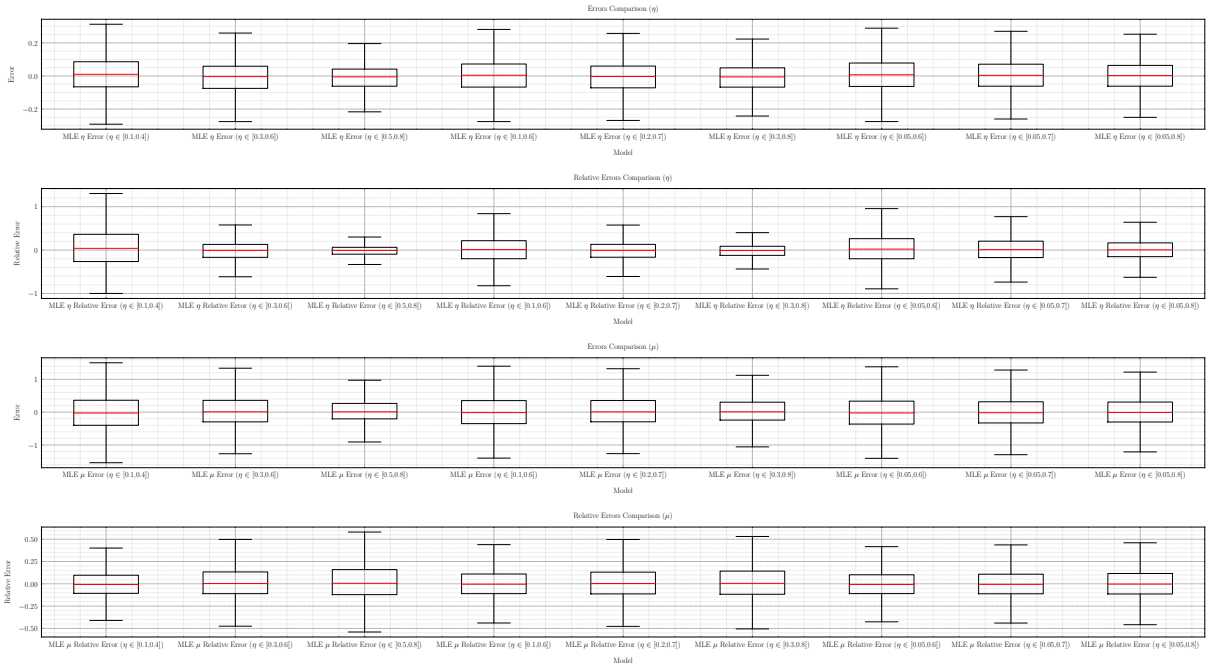

FIGURE 10 – Comparaison des erreurs en fonction de η - MLE ($\Delta = 1.0$)

Sur la figure 11, on compare les erreurs en fonction de β. Les erreurs se concentrent en 0 pour chaque intervalle. Néanmoins, les erreurs sont plus faibles pour les intervalles $\beta \in [2.5, 5]$ et $\beta \in [3, 5]$. Ainsi, lorsque l'intervalle de β est restreint et que $\beta \to 5$, les estimations sont meilleures. On conclut qu'une taille réduite et des valeurs précises pour l'intervalle de β donnent des estimations plus précises.

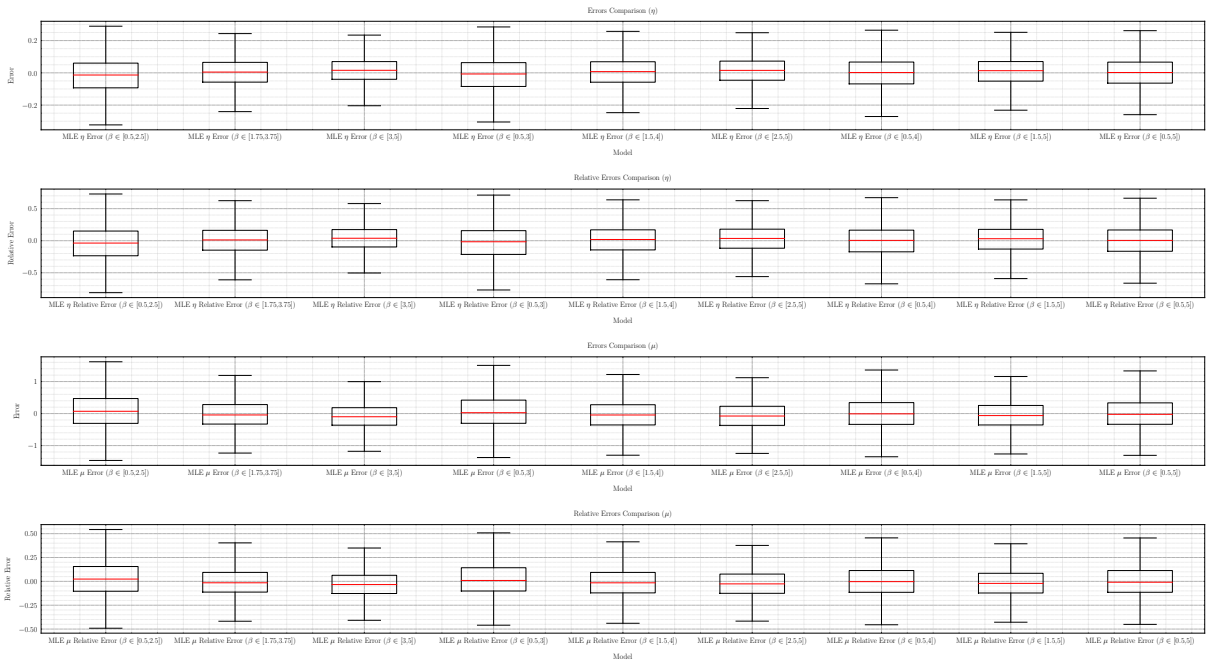

FIGURE 11 – Comparaison des erreurs en fonction de β - MLE ($\Delta = 1.0$)

5.2 Évaluation des méthodes d'apprentissage supervisé

5.2.1 Méthodologie des tests

Dans cette section, trois tests sont réalisés afin d'évaluer le modèle le plus adéquate à la prédiction de processus de Hawkes agrégés entre les FFNN (MLP) et RNN (LSTM). Le premier test correspond à l'évaluation des taux de convergences d'entraînement et de validation. Ce test donne des informations sur la vitesse et la précision de convergence d'un algorithme. De plus, le test permet potentiellement de réfuter l'hypothèse du sous-apprentissage ou sur-apprentissage du modèle. Les deux autres tests et leurs objectifs sont similaires à ceux effectués sur le MLE randomisé. Les gammes de valeurs sélectionnées pour tester chaque paramètre variable sont les mêmes et les valeurs des paramètres par défaut sont :

Paramètres	Valeurs
Nombre de processus (N)	5,000
Taux de décroissance (β)	$\mathcal{U}(p = 1, q = 3)$
Taux de branchement (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Pas de discrétisation (Δ)	1
Écart type (σ)	10

TABLE 8 – Valeurs par défaut des paramètres des tests

5.2.2 Comparaison des erreurs du MLE/LSTM

Sur la figure 12, est affiché les taux de convergence d'entrainement et de validation des modèles. Sur 500 epochs, on constate que le MLP converge plus rapidement pendant l'entraînement pour atteindre la perte la plus basse au alentour de 0.05. Pendant la validation, il converge aussi rapidement pour se stabiliser autour de 0.12 de perte. Les deux courbes sont proches au début et ne divergent pas beaucoup l'une de l'autre vers la fin de l'exécution. En outre, la perte de validation est supérieure à celle d'entrainement, il n'y a donc à priori pas de sous-apprentissage ou de sur-apprentissage. Les taux de convergence du LSTM sont assez similaires et regroupés. Durant l'entraînement, le modèle converge moins vite que le MLP mais finit par se rapprocher de ses résultats de perte. Durant la validation, le LSTM converge plus rapidement que le MLP et obtient un meilleur résultat avec une perte de 0.11. De même que le MLP, les deux courbes sont proches et convergent vers la fin, il n'y a donc pas de risques de sous-apprentissage ou de sur-apprentissage. En terme de taux de convergence, le LSTM est meilleur et plus rapide que le MLP.

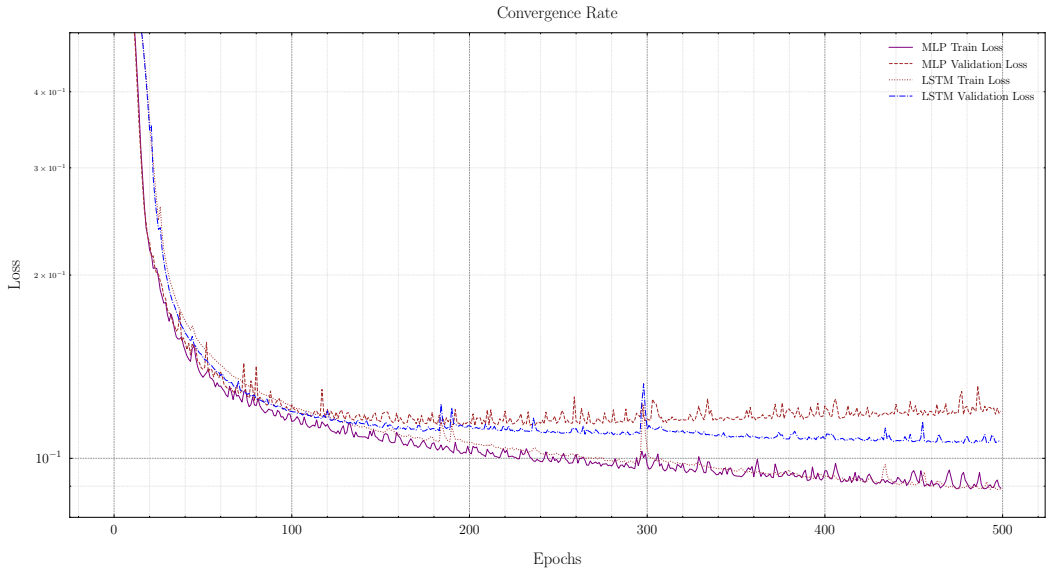

FIGURE 12 – Taux de convergence - MLP/LSTM

Sur la figure 13, on compare les erreurs et les erreurs relatives entre MLP et LSTM. De prime abord, les résultats pour η et μ semblent tous centrés en 0 avec un léger décalage négatif de la médiane pour l'erreur MLP de μ par rapport à celle du LSTM. L'asymétrie moindre pour les deux modèles indique une distribution plus symétrique autour de la médiane et donc une bonne précision. Néanmoins, en analysant les médianes des erreurs dans le tableau 19, on constate que le LSTM est plus précis et efficace :

Erreur standard				Erreur relative			
Taux de branchement η		Intensité de base μ		Taux de branchement η		Intensité de base μ	
MLP	LSTM	MLP	LSTM	MLP	LSTM	MLP	LSTM
0.008	**-0.0004**	-0.073	**0.006**	0.02	**-0.0012**	-0.028	**0.002**

TABLE 9 – Erreur Standard/Relative du MLP/LSTM

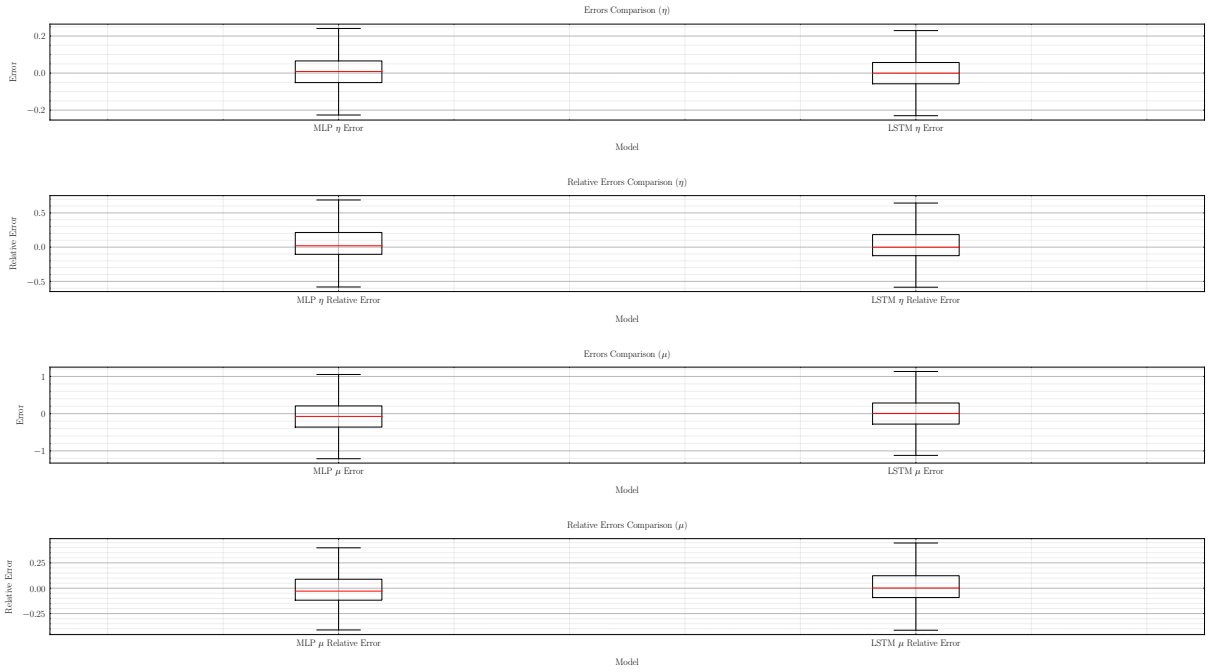

FIGURE 13 – Comparaison des erreurs - MLP/LSTM ($\Delta = 1.0$)

5.2.3 Impacts des paramètres - MLP

Dans cette sous-partie, on étudie les erreurs et les erreurs relatives de η et μ pour le modèle MLP à l'instar du MLE. Sur la figure 14, on observe que les erreurs sont plutôt stables selon Δ. Néanmoins, on note une forte concentration des erreurs autour de la médiane pour $\Delta = 1.0$ et $\Delta = 2.0$. Pour ces valeurs de Δ, il y a un bon compromis biais-variance et l'estimation est optimale. Par conséquent, lorsque Δ augmente, le nombre d'événements agrégés par intervalle d'observation augmente mais la précision des estimations reste constante. Les données et les résultats du MLP ne perdent pas en granularité. Ainsi, on remarque que les erreurs d'estimation pour η et μ persistent en fonction de l'augmentation de Δ.

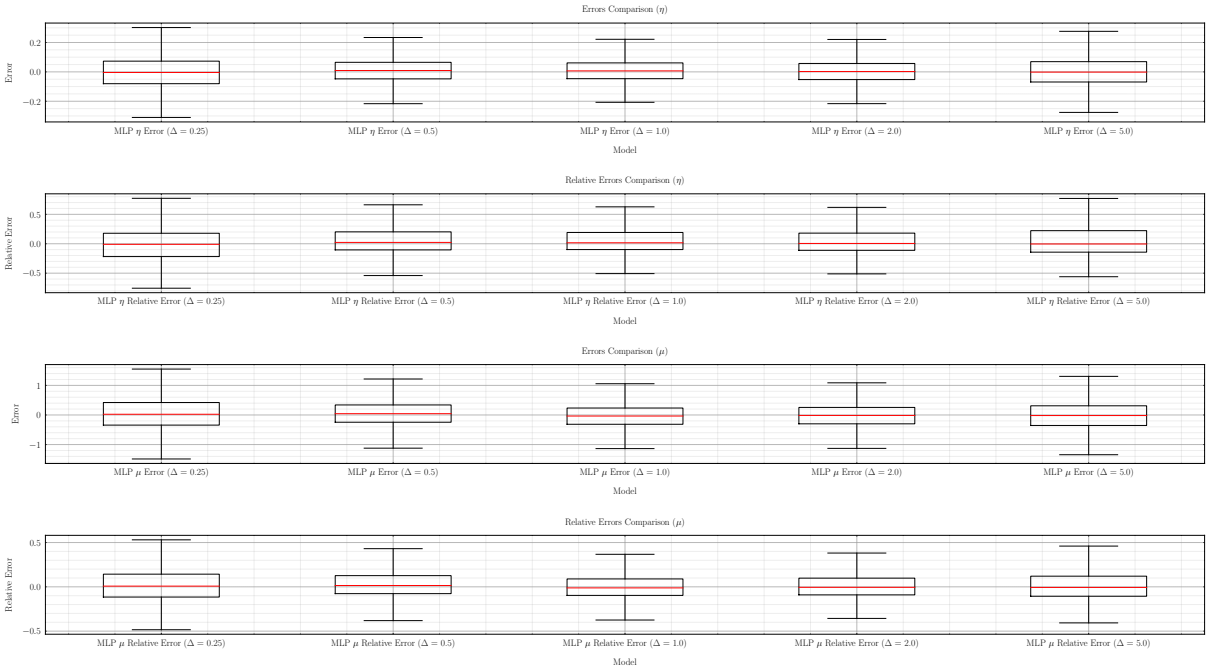

FIGURE 14 – Comparaison des erreurs en fonction de Δ - MLP

Sur la figure 15, on compare les erreurs en fonction du niveau d'activité attendu E. On retrouve approximativement les mêmes résultats qu'avec le MLE. Les erreurs et erreurs relatives gravitent de moins en moins autour de 0 et leurs dispersions augmentent en fonction de E. En effet, les erreurs d'estimation de η et de μ augmentent lorsque $E \to \infty$ et inversement. Ainsi, lorsque le niveau d'activité attendu E augmente, η et μ sont plus susceptibles de varier rendant leurs estimations moins précises. On en déduit que les erreurs d'estimation de η et de μ s'intensifient en fonction de l'augmentation de E.

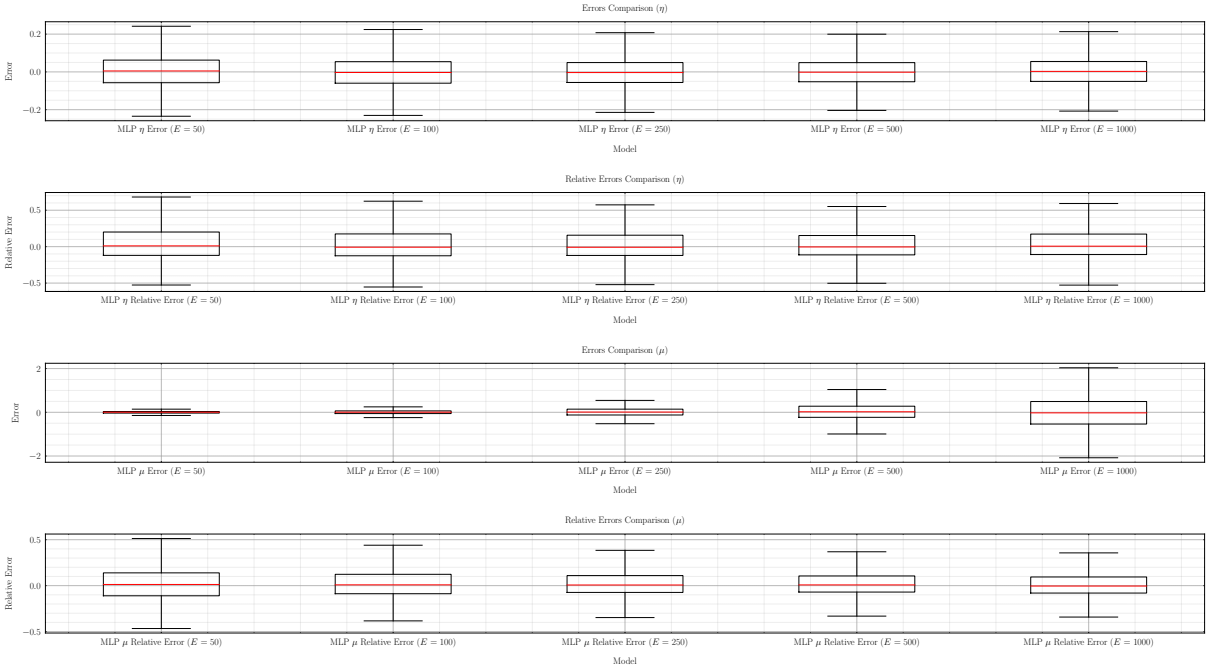

FIGURE 15 – Comparaison des erreurs en fonction de E - MLP ($\Delta = 1.0$)

Sur la figure 16, on compare l'évolution des erreurs en fonction de η. Les résultats sont homogènes avec des erreurs autour de 0 pour chaque intervalle de η. Cependant, les erreurs varient moins pour les intervalles $\eta \in [0.3, 0.8]$ et $\eta \in [0.5, 0.8]$. Ainsi, lorsque l'intervalle de η est restreinte et que $\eta \to 0.8$, les estimations de η et μ sont meilleures. On en déduit qu'une taille réduite et des valeurs précises pour l'intervalle de η donnent des estimations plus robustes selon le domaine d'expertise et le jeu de données.

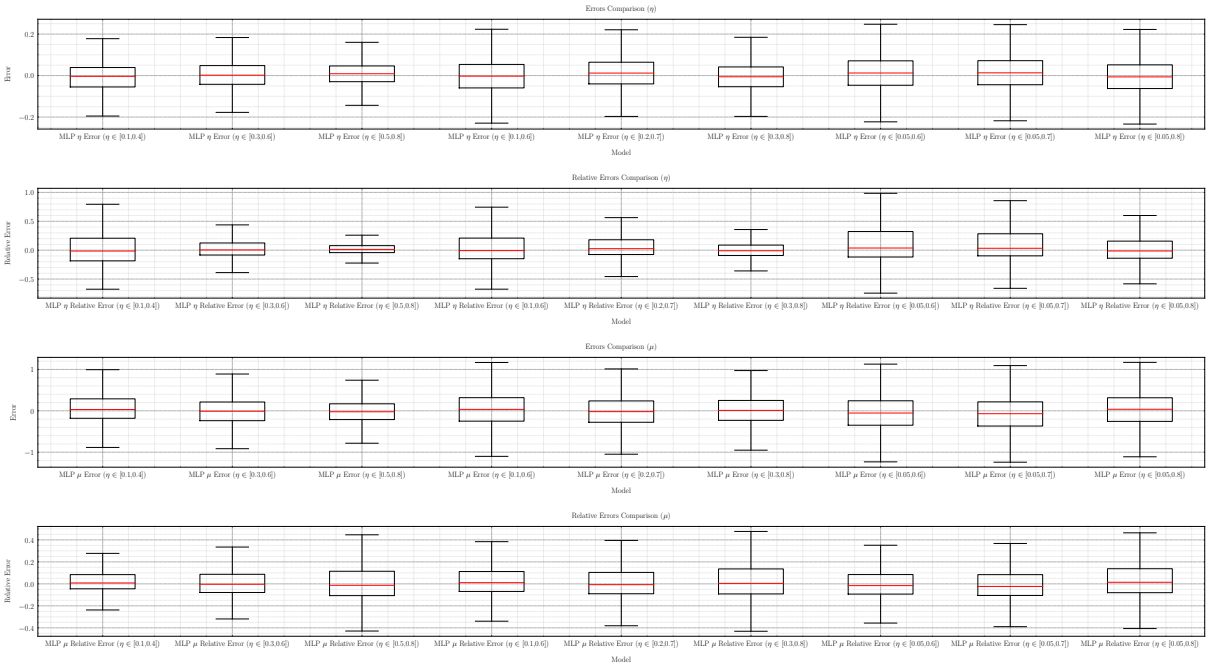

FIGURE 16 – Comparaison des erreurs en fonction de η - MLP ($\Delta = 1.0$)

Sur la figure 17, on compare les erreurs en fonction de β. Les erreurs se concentrent en 0 et sont dispersées approximativement de la même manière pour chaque intervalle. Comme pour le MLE, on note qu'une taille réduite et des valeurs précises pour l'intervalle de β donnent des estimations plus précises.

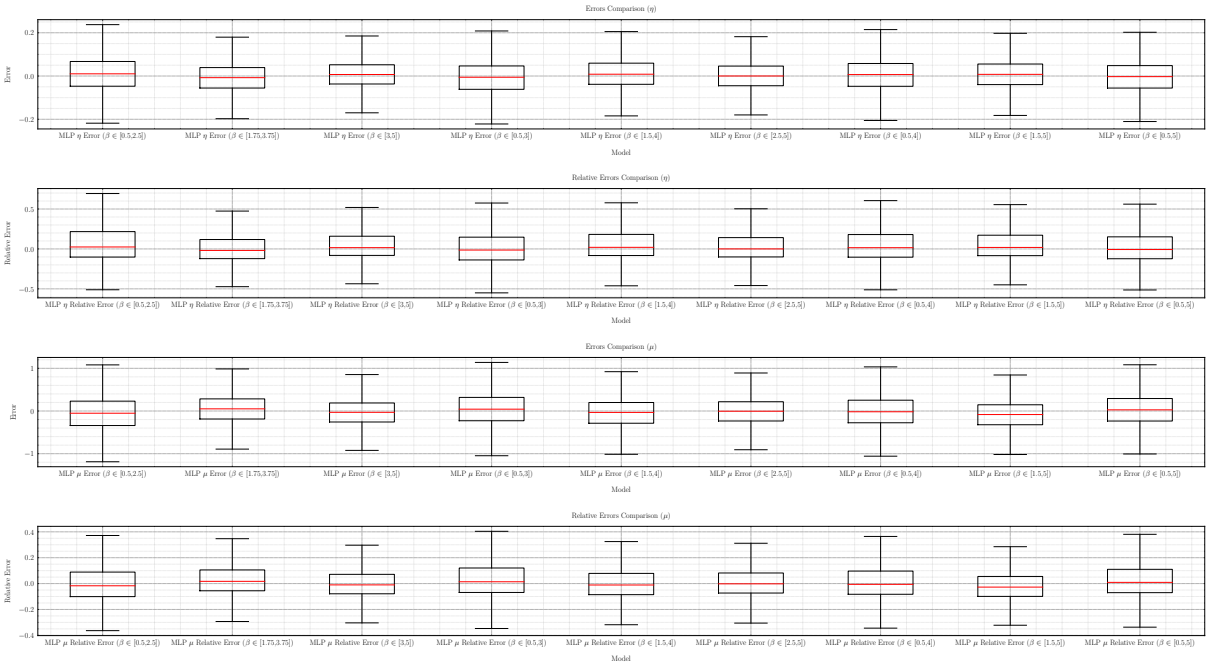

FIGURE 17 – Comparaison des erreurs en fonction de β - MLP ($\Delta = 1.0$)

Seul les erreurs en fonction de β varient légèrement. En guise de correction, modifier les modèles ou allonger l'horizon T pour des intervalles de β plus élevés peut être envisagé. De manière générale, les erreurs et les erreurs relatives ne sont majoritairement pas altérées par les différents paramètres.

5.2.4 Impacts des paramètres - LSTM

Dans ce chapitre, on s'intéresse aux résultats du LSTM en fonction de Δ, E, η, β. Sur la figure 18, on examine l'influence de Δ sur les erreurs du LSTM. On constate que les erreurs sont stables selon Δ et cohérentes avec celles du MLP. On observe une plus légère concentration des erreurs pour $\Delta = 1.0$ et $\Delta = 2.0$. Lorsque Δ augmente, le nombre d'événements agrégés par intervalle d'observation augmente mais la précision des estimations reste stable. Les données et les résultats du MLP ne perdent pas en granularité. Ainsi, on remarque que les erreurs persistent en fonction de l'augmentation de Δ.

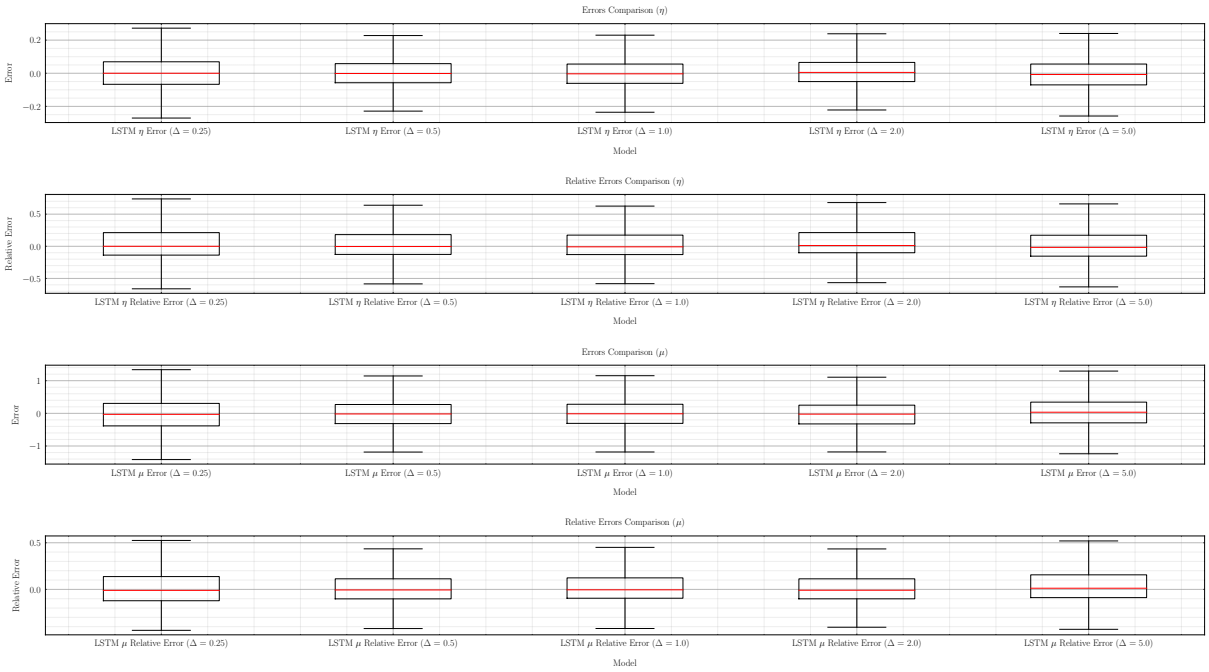

FIGURE 18 – Comparaison des erreurs en fonction de Δ - LSTM

Sur la figure 19, on analyse les erreurs du LSTM en fonction du niveau d'activité attendu E. On retrouve approximativement les mêmes résultats qu'avec le MLP. Les erreurs et erreurs relatives gravitent autour de 0 mais leurs dispersions augmentent en fonction de E. On remarque que les erreurs d'estimation de η et de μ augmentent lorsque $E \to \infty$ et inversement. Ainsi, lorsque le niveau d'activité attendu E augmente, η et μ sont plus susceptibles de varier rendant leurs estimations moins précises. On en déduit que les erreurs d'estimation de η et de μ s'intensifient en fonction de l'augmentation de E.

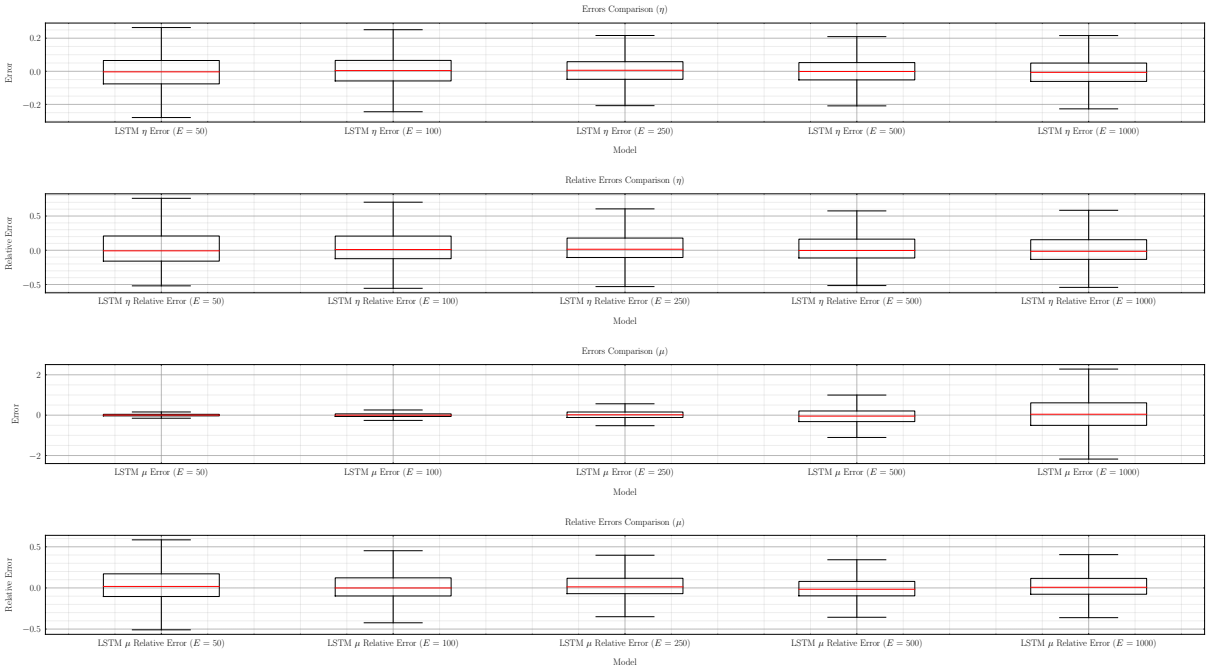

FIGURE 19 – Comparaison des erreurs en fonction de E - LSTM ($\Delta = 1.0$)

Sur la figure 20, on examine les résultats du LSTM en fonction de η. On observe que les erreurs sont très faibles et proches de 0 pour chaque intervalle. Les boîtes et moustaches sont respectivement fines et courtes marquant des distributions plus compacts que celles du MLP. Les erreurs varient moins pour les intervalles $\eta \in [0.3, 0.8]$ et $\eta \in [0.5, 0.8]$. Ainsi, lorsque l'intervalle de η est restreinte et que $\eta \to 0.8$, les estimations de η et μ sont meilleurs. On en déduit qu'une taille réduite et des valeurs précises pour l'intervalle de η donnent des estimations plus robustes selon le domaine d'expertise et le jeu de données.

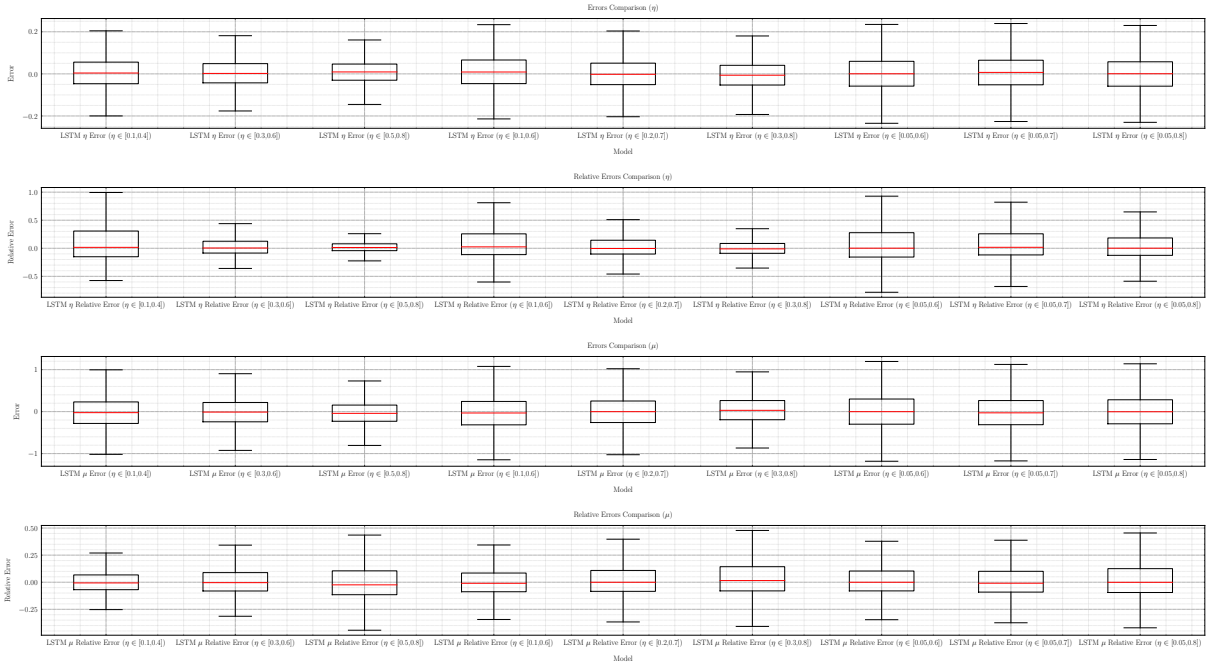

FIGURE 20 – Comparaison des erreurs en fonction de η - LSTM ($\Delta = 1.0$)

Sur la figure 21, on étudie les erreurs du LSTM en fonction de β. Les erreurs tournent autour de 0 et sont étalées majoritairement de la même façon pour chaque intervalle. Comme pour le MLP, on note qu'une taille réduite et des valeurs précises pour l'intervalle de β donnent des estimations plus fiables.

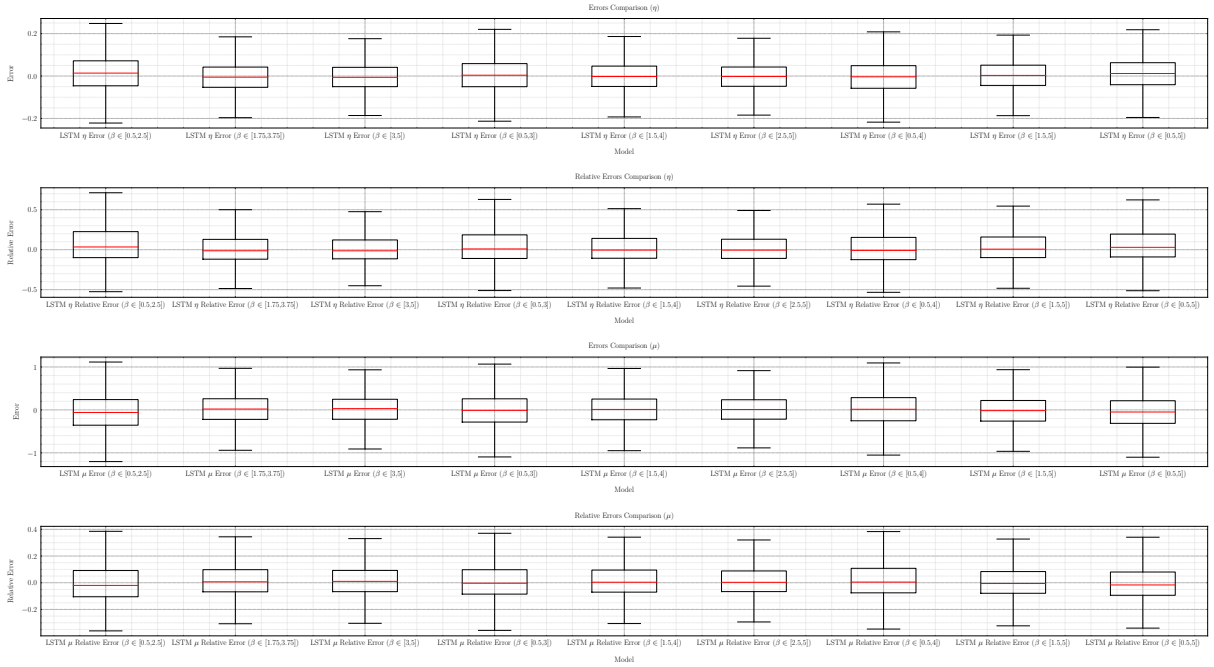

FIGURE 21 – Comparaison des erreurs en fonction de β - LSTM ($\Delta = 1.0$)

A l'instar du MLP, les résultats du LSTM varient très peu selon les paramètres. De même, modifier l'architecture du RNN ou allonger l'horizon T semblent être une bonne façon de corriger ces différences.

5.3 Comparaison des méthodes

5.3.1 Méthodologie des tests

Dans cette section, on s'intéresse à la comparaison des erreurs et des prédictions entre chaque modèle. Le premier test compare l'erreur et l'erreur relative des estimations par rapport aux données réelles. Le but est de voir si les estimations et la dispersion des erreurs autour de la médiane sont corrects. Le second test compare les prédictions de η et μ en fonction de Δ pour un η et β donné. Le but est d'analyser les évolutions des prédictions selon Δ, η et β. Les valeurs des paramètres par défaut des deux tests sont :

Paramètres	Valeurs
Nombre de processus (N)	5,000
Taux de décroissance (β)	$\mathcal{U}(p = 1, q = 3)$
Taux de branchement (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Pas de discrétisation (Δ)	1
Écart type (σ)	10

TABLE 10 – Valeurs par défaut des paramètres du test n°1

Paramètres	Valeurs
Nombre de processus (N)	100
Taux de décroissance (β)	2
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Écart type (σ)	10

TABLE 11 – Valeurs par défaut des paramètres du test n°2

Pour le test n°2, une gamme de valeurs a été sélectionnée pour chaque paramètre parmi Δ et η :

Taux de branchement (η)	Pas de discrétisation (Δ)
0.2	0.25
0.5	0.5
0.8	1
	2
	5

TABLE 12 – Valeurs variables des paramètres du test n°2

Les modèles sont entraînés avec les valeurs par défaut des paramètres d'entraînement et sont testés avec les valeurs par défaut et variables des paramètres des tests. Par exemple, si on souhaite obtenir les prédictions de η et μ sachant le tableau ci-dessus, on entraîne chaque modèle pour chaque Δ selon les valeurs par défaut des paramètres d'entraînement. Puis on teste ces modèles pour chaque η en prenant en compte les valeurs par défaut des paramètres du test. Pour le test n°2, les valeurs réelles de μ (4.0, 2.5, 1.0) représentent la moyenne arrondie des valeurs de μ pour chaque Δ et η fixés (0.2, 0.5, 0.8).

5.3.2 Comparaison des erreurs du MLE/MLP/LSTM

Les médianes des erreurs pour le MLE, MLP et le LSTM sont données dans le tableau suivant :

Erreur standard						Erreur relative					
Taux de branchement η			Intensité de base μ			Taux de branchement η			Intensité de base μ		
MLE	MLP	LSTM	MLE	MLP	LSTM	MLE	MLP	LSTM	MLE	MLP	LSTM
0.0012	0.008	**-0.0004**	-0.013	-0.073	**0.006**	0.002	0.02	**-0.0012**	-0.005	-0.028	**0.002**

TABLE 13 – Erreur Standard/Relative du MLE/MLP/LSTM

Sur la figure 22, on observe les erreurs et erreurs relatives de η et μ pour le MLE, MLP et LSTM. On constate que les erreurs sont centrées en 0 avec une faible dispersion pour tous les modèles. Le LSTM semble légèrement plus précis que le MLE et le MLP en terme de médiane, variabilité d'écart interquartile et symétrie. Dans l'ensemble, l'estimation de η est vaguement meilleur que celle de μ.

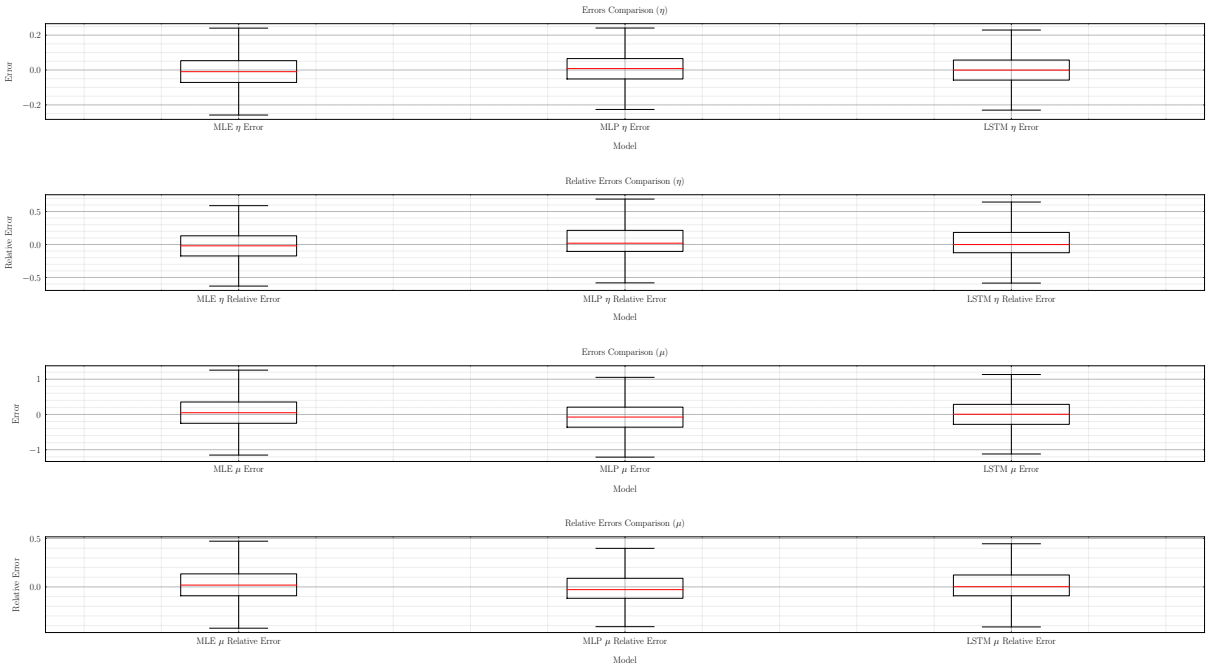

FIGURE 22 – Comparaison des erreurs - MLE/MLP/LSTM ($\Delta = 1.0$)

5.3.3 Comparaison des prédictions et impacts des paramètres

Dans cette partie, on étudie les prédictions de η et μ à travers les variations de Δ. Sur la figure 23, on explore les prédictions de η sachant $\eta = 0.2$ et $\beta = 2.0$ pour chaque modèle et valeur de Δ. On constate que les médianes et les prédictions du MLE s'éloignent de la valeur effective de η lorsque Δ augmente. En effet, le nombre d'événements agrégés par intervalle d'observation augmente ce qui diminue la précision des estimations. Les données et les résultats du MLE perdent en granularité. Pour le MLP et le LSTM, les résultats gravitent autour de 0.2 sans grandes variations. Lorsque Δ augmente, le nombre d'événements agrégés par intervalle d'observation augmente mais la précision des estimations reste stable. Les données et les résultats ne perdent pas en granularité. Ainsi, les résultats du MLP et du LSTM sont plutôt stables selon l'augmentation de Δ alors que ceux du MLE se dégradent lorsque $\Delta \to \infty$.

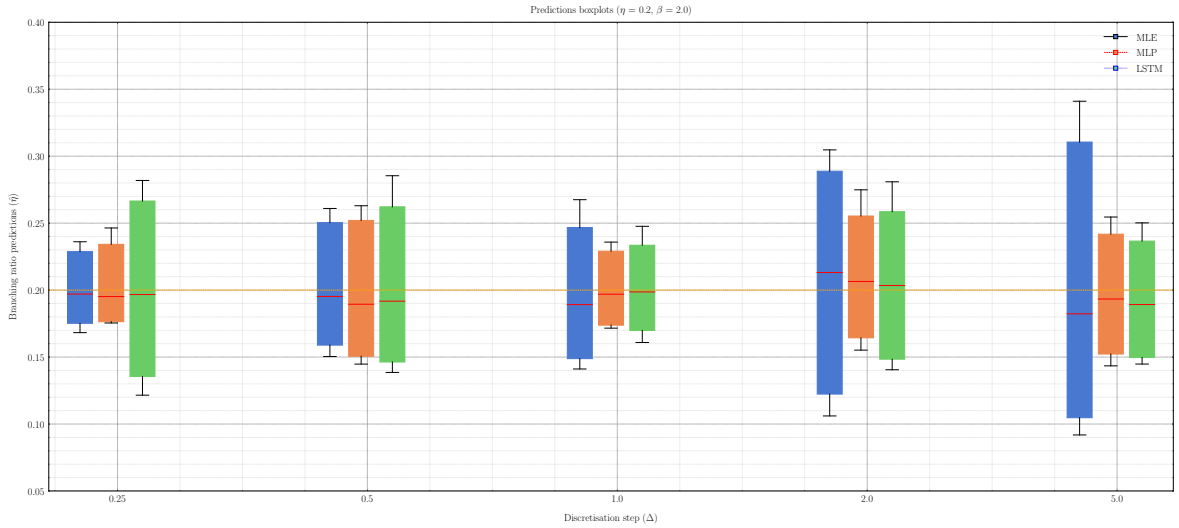

FIGURE 23 – Comparaison des prédictions de η en fonction de Δ, η et β ($\eta = 0.2$, $\beta = 2.0$)

Sur la figure 24, on observe les prédictions de η sachant $\eta = 0.5$ et $\beta = 2.0$ pour chaque modèle et valeur de Δ. Globalement, les prédictions en fonction de Δ sont plus dispersées qu'avec $\eta = 0.2$. On constate que les médianes et les prédictions du MLE s'éloignent de la valeur effective de η lorsque Δ augmente. La précision des estimations diminue quand le nombre d'événements agrégés dans l'intervalle d'observation augmente. Cependant, cette détérioration des résultats du MLE est moins marquée qu'avec $\eta = 0.2$. Pour les deux modèles d'apprentissage supervisé, les résultats sont plus étalés et tournent autour de 0.5 sans grandes différences. L'augmentation du nombre d'événements agrégés par intervalle d'observation ne semble pas déstabiliser la précision des estimations. Par conséquent, les résultats du MLP et du LSTM sont constants en fonction de Δ alors que ceux du MLE se dégradent modérément lorsque $\Delta \to \infty$.

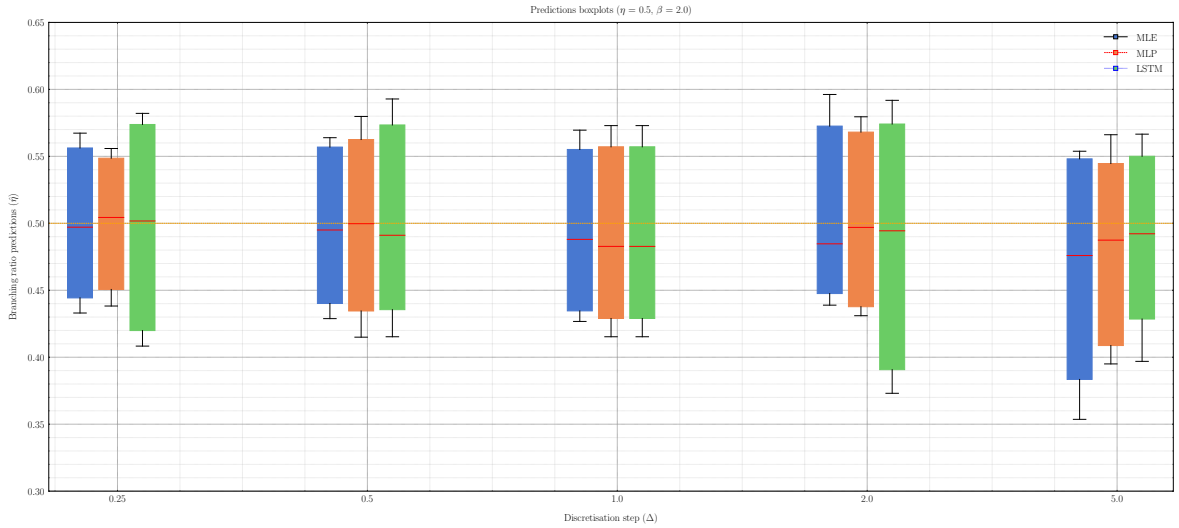

FIGURE 24 – Comparaison des prédictions de η en fonction de Δ, η et β ($\eta = 0.5$, $\beta = 2.0$)

Sur la figure 25, on observe les prédictions de η sachant $\eta = 0.8$ et $\beta = 2.0$ en fonction de Δ. Les résultats sont hétérogènes et les étalements des prédictions entre les modèles plus fluctuants. On note que les médianes et les prédictions du MLE s'éloignent de la valeur effective de η lorsque Δ augmente. Ainsi, la précision des estimations diminue lorsque Δ et le nombre d'événements agrégés augmentent. Cette altération est moins marquée qu'avec $\eta = 0.2$ mais plus qu'avec $\eta = 0.5$. Pour le MLP et le LSTM, les prédictions se concentrent autour de 0.8 avec une dispersion notable pour $\Delta = 0.5$ et $\Delta = 2.0$. L'augmentation de Δ n'ébranle pas la précision des modèles. En conséquence, les résultats du MLP et du LSTM sont plutôt stables en fonction de Δ alors que ceux du MLE se détériorent lorsque $\Delta \to \infty$.

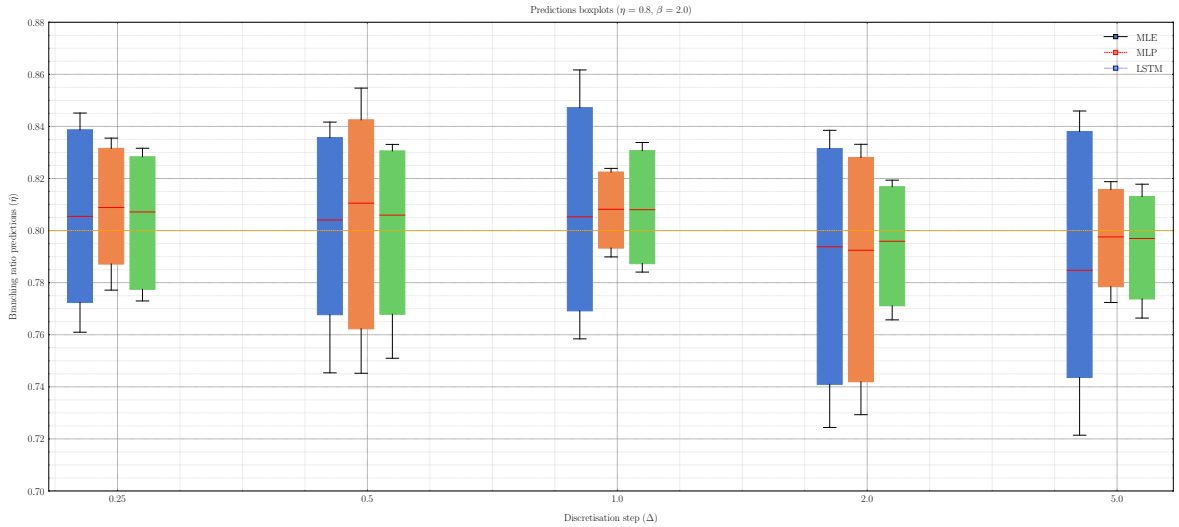

FIGURE 25 – Comparaison des prédictions de η en fonction de Δ, η et β ($\eta = 0.8$, $\beta = 2.0$)

Sur la figure 26, on observe les prédictions de μ sachant $\eta = 0.2$ et $\beta = 2.0$ pour chaque modèle et valeur de Δ. Les médianes des prédictions de μ sont moins proches de la valeur réelle par rapport à celles des prédictions de η. En outre, on note que l'étalement des prédictions du MLE s'agrandit lorsque Δ augmente. En effet, la précision des estimations diminue avec l'augmentation du nombre d'événements agrégés et de l'intervalle d'observation. Pour le MLP et le LSTM, les prédictions gravitent autour de 4.0 avec une dispersion marquée pour $\Delta = 0.5$ et $\Delta = 2.0$. L'augmentation du nombre d'événements agrégés par intervalle d'observation ne détériore pas la précision des estimations. En définitive, les résultats du MLP et du LSTM sont fiables en fonction de Δ alors que ceux du MLE se dégradent lorsque $\Delta \to \infty$.

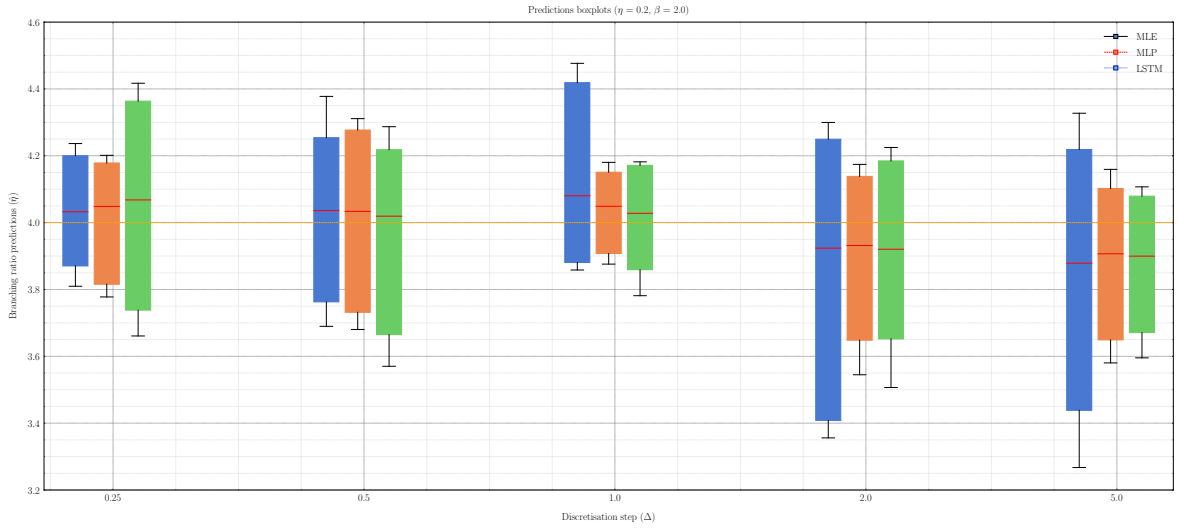

FIGURE 26 – Comparaison des prédictions de μ en fonction de Δ, η et β ($\eta = 0.2$, $\beta = 2.0$)

Sur la figure 27, on observe les prédictions de μ sachant $\eta = 0.5$ et $\beta = 2.0$ en fonction de Δ. Les résultats sont homogènes et les dispersions sont similaires aux prédictions de η sachant $\eta = 0.5$. En effet, les étalements sont moins flottants. On aperçoit encore que les médianes et les prédictions du MLE s'éloignent de la valeur réelle de μ lorsque Δ augmente. La précision des estimations diminue lorsque Δ et le nombre d'événements agrégés s'accentuent. Cette altération est moins marquée qu'avec $\eta = 0.2$. Pour les deux modèles d'apprentissage supervisé, les prédictions se concentrent autour de 2.5 avec une grande dispersion pour chaque Δ. L'augmentation du nombre d'événements agrégés par intervalle d'observation n'affaiblit pas la précision des estimations. Ainsi, les résultats du MLP et du LSTM sont plutôt stables en fonction de Δ alors que ceux du MLE se détériorent modérément lorsque $\Delta \to \infty$.

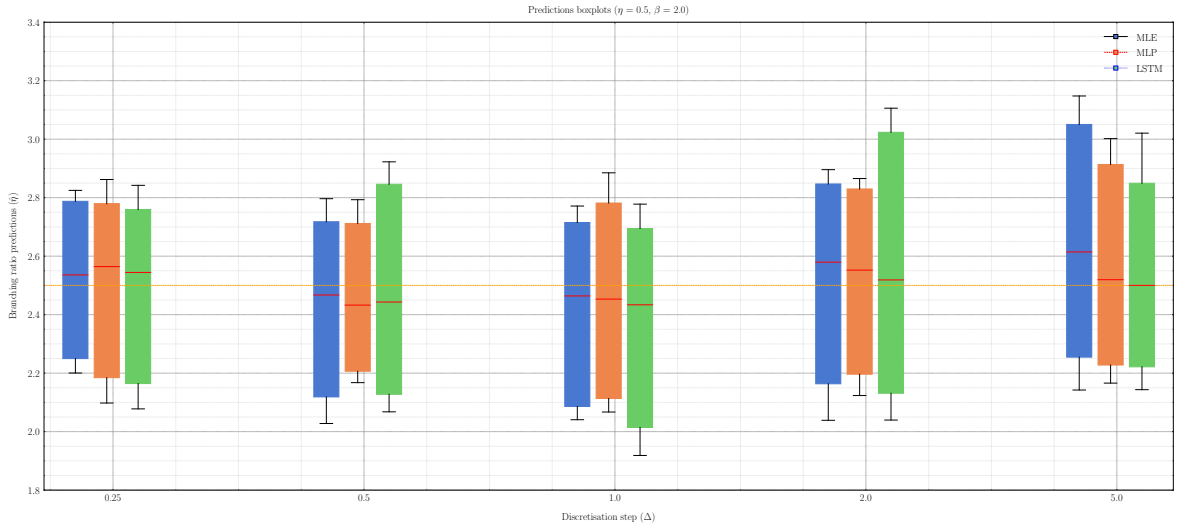

FIGURE 27 – Comparaison des prédictions de μ en fonction de Δ, η et β ($\eta = 0.5$, $\beta = 2.0$)

Sur la figure 28, on observe les prédictions de μ sachant $\eta = 0.8$ et $\beta = 2.0$ pour chaque modèle et valeur de Δ. Comme pour les prédictions de η, les résultats sont plus hétérogènes qu'avec $\eta = 0.2$ et $\eta = 0.5$. En effet, les étalements entre les modèles sont plus variables. On constate que les médianes et les prédictions du MLE s'éloignent de la valeur réelle de μ lorsque Δ augmente. Ainsi, la précision du MLE est déjà plutôt faible pour des intervalles d'observations réduites et diminue lorsque cet intervalle et le nombre d'événements agrégés augmentent. Cette variation des résultats du MLE se démarque moins qu'avec $\eta = 0.2$ mais plus qu'avec $\eta = 0.5$. Pour le MLP et le LSTM, les prédictions se concentrent autour de 1.0 avec une dispersion notable pour $\Delta = 1.0$ et $\Delta = 5.0$. L'augmentation du nombre d'événements agrégés modifie légèrement la précision des estimations. Par conséquent, les résultats du MLP et du LSTM sont robustes en fonction de Δ alors que ceux du MLE déclinent lorsque $\Delta \to \infty$.

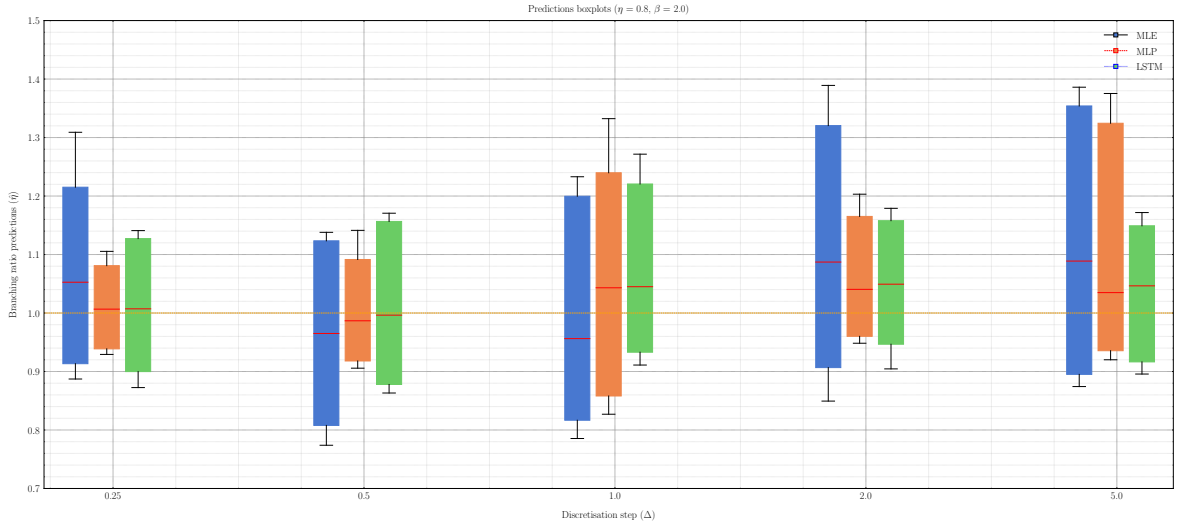

FIGURE 28 – Comparaison des prédictions de μ en fonction de Δ, η et β ($\eta = 0.8$, $\beta = 2.0$)

En conclusion, quelque soit le paramètre estimé et les valeurs réelles fixées, les prédictions du MLP et du LSTM sont globalement stables alors que les prédictions du MLE se dégradent lorsque Δ augmente. Tout modèle confondu, les estimations sont plus précises sachant $\eta = 0.2$ et plus dispersées sachant $\eta = 0.5$. En effet, lorsque $\eta = 0.5$, la précision des estimations est plus faible même pour les valeurs de $\Delta \to 0$. En outre, plus les estimations sont robustes plus les résultats entre le MLE et les modèles d'apprentissage supervisé se démarquent en fonction de Δ. En conséquence, lorsque $\Delta \to 0$, le MLE est souvent meilleur que les autres modèles, mais lorsque $\Delta \to \infty$ alors le MLP et le LSTM sont plus fiables.

6 Discussion

Dans la première partie des résultats, les performances de reconstruction de l'intensité conditionnelle par le Poisson-VAE et le Poisson-VAE à double décodeur sont insuffisantes pour passer à l'inférence bayésienne. Ce constat est surprenant dans la mesure où la méthodologie utilisée est tirée de la thèse de T. Keane [5] dans laquelle les résultats du Poisson-VAE et Poisson-VAE à double décodeur sont très prometteurs. Les erreurs de reconstructions sont plus faibles et l'intensité décodée moins lisse. Le seul point commun est l'effet limité du Poisson-VAE à double décodeur. Pour améliorer les résultats, il est possible de revoir la méthodologie, l'architecture des modèles ou de modifier le schéma de calcul de l'erreur de reconstruction. Par exemple, si le poids de la perte de reconstruction θ a été réduit avec le calcul de la divergence KL, il est probable que les décodeurs surpassent le Poisson-VAE en termes de reconstruction d'intensité. C'est un domaine de recherche suggéré, car une reconstruction performante s'avère très utile dans les cas où la quantité de données est faible, étant donné que l'utilisation de l'inférence bayésienne permet d'échantillonner des processus de Hawkes agrégés tout en préservant les paramètres d'intensité conditionnelles sous-jaçents. En outre, l'entraînement des deux Poisson-VAE a été très difficile car la taille de l'espace latent limite fortement les performances. La dimension latente de 15 a été choisie pour offrir la flexibilité nécessaire à la reconstruction des données d'entrée. La reproduction de ces résultats avec un espace latent de dimension inférieure est un sujet qui mérite d'être développé.

Dans la deuxième partie des résultats, les performances des trois modèles pour estimer η et μ sont nuancées. Pour des valeurs de $\Delta \to 0$, le MLE est meilleur que les modèles d'apprentissage supervisé. Cependant, lorsque Δ augmente, les résultats du MLE se détériorent au profit des résultats plus stables du MLP et du LSTM. Entre ces deux derniers, les résultats sont équivalents même si le LSTM est plus précis lorsque la valeur effective fixée de $\eta \to 1$. En outre, pour chaque modèle les erreurs s'accentuent lorsque le niveau d'activité attendu E augmente. De même, pour un intervalle fixé de η ou de β, les estimations sont plus robustes lorsque l'intervalle est de taille réduite et les valeurs précises. Globalement, les performances des modèles en fonction de Δ, E, η, β sont tous approximativement similaires avec de plus fortes variations pour les erreurs et les prédictions du MLE. Pour approfondir, il serait pertinent d'allonger l'horizon T ou d'essayer d'autres plages de paramètres. En particulier des valeurs extrêmes ou des intervalles étendus. Pour le MLP et le LSTM, une source d'amélioration possible peut être de modifier l'architecture, la régularisation et les paramètres d'entrainement. Dans l'ensemble, l'étude montre que l'estimation d'un MLP ou d'un LSTM est fiable avec une précision analogue au MLE et un temps de calcul plus rapide. En effet, on effectue un test en dix secondes avec un réseau de neurones alors qu'il faut dix minutes avec un MLE. Afin d'affiner la comparaison des modèles, des recherches poussées sur de nouvelles techniques d'apprentissage automatique pourraient être réalisées. Notamment, sur les mécanismes d'auto-attention [38] et les « transformers » [39]. Ces derniers résument l'influence des événements antérieurs et calculent la probabilité du prochain événement tout en étant rapide et efficace.

7 Conclusion

Le problème principal de cette thèse était d'estimer les paramètres de processus de Hawkes à partir de données de comptage dites « agrégées » où les temps d'événements sont censurés par l'imprécision d'une observation. Cette inexactitude est un problème grandissant dans les domaines de la finance quantitative pour la conduite de l'inférence statistique. Le but de cette thèse était d'étudier différents modèles d'apprentissage profond pour l'estimation de processus de Hawkes agrégées afin de les comparer aux modèles de l'état de l'art et d'optimiser ce problème d'inférence. Dans un premier temps, des processus de Hawkes ont été simulés à partir de données de comptage. Ensuite, une modélisation de l'intensité de base μ, du ratio d'endogénéité η et de l'intensité conditionnelle λ des processus agrégés a été réalisé à l'aide d'un Poisson-VAE. Ensuite, un MLP et un LSTM ont été utilisés pour estimer ces mêmes paramètres. Les données agrégées sont des données où le comptage d'événements a été effectué à intervalles réguliers plutôt qu'à un moment précis. Ces données discrétisées ont été simulées grâce à plusieurs hyperparamètres et une fonction de noyau exponentielle déjà développés dans la littérature. Ensuite, les travaux sur les auto-encodeurs variationnelles avec une distribution de Poisson et un potentiel double décodeur ont démontré leur incapacité à reconstruire l'intensité conditionnelle λ d'une manière non supervisée. Malgré la prise en compte de η et μ dans l'espace latent en plus de λ, le doute sur cette méthodologie subsiste encore pour réaliser l'inférence statistique des paramètres d'un processus de Hawkes agrégé. Cependant, la structure de l'encodeur et des décodeurs est un élément qui peut bénéficier d'améliorations plus poussées. Par exemple, les RNN peuvent apporter de la valeur à cette méthodologie. De même, jouer sur le niveau de compression est intéressant pour de futurs études. L'application novatrice des modèles d'apprentissage supervisé pour l'estimation de η et μ s'est avérée fiable sur de nombreuses plages de paramètres. La robustesse de l'estimation entre les différents niveaux de discrétisation, d'activité attendue et de taux de décroissance est remarquable. Malgré les distributions d'erreur à longue queue observées, le MLP et le LSTM ont donné des résultats proches, voir parfois meilleurs, que le MLE avec des avantages distincts en termes de temps de calcul, d'accessibilité et de flexibilité des données. Le temps de calcul réduit des méthodes d'apprentissage automatique permet des gains de temps significatifs, notamment dans le cas de processus à forte activité attendue. La flexibilité concernant les données et l'accessibilité accrue garantissent de nombreux cas d'utilisation. Les domaines d'approfondissements suggérés comprennent l'utilisation de nouveaux modèles d'apprentissage supervisé, des mécanismes d'auto-attention et « transformers ». Dans cette thèse a été établi trois nouvelles méthodologies pour effectuer l'inférence statistique des paramètres des processus de Hawkes agrégés : le Poisson-VAE, le MLP et le LSTM. Ce domaine d'étude s'est développé récemment en tant qu'extension des processus de Hawkes. Les méthodes présentées constituent une bonne contribution aux travaux réalisés dans [27] et [5].

Références

[1] A. G. Hawkes, *Spectra of Some Self-exciting and Mutually Exciting Point Processes*, Biometrika, vol. 58, no. 1, pp. 83–90, 1971.

[2] Y. Ogata, *Space-Time Point-Process Models for Earthquake Occurrences*, Annals of the Institute of Statistical Mathematics, vol. 50, no. 2, pp. 379–402, 1998.

[3] P. Reynaud-Bouret et S. Schbath *Adaptive estimation for Hawkes processes ; application to genome analysis*, Annals of the Institute of Statistical Mathematics, vol. 38.5, no. 1, pp. 2781—2822, 2010.

[4] E. Bacry, I. Mastromatteo et J.-F. Muzy, *Hawkes Processes in Finance. Market Microstructure and Liquidity*, Market Microstructure and Liquidity, vol. 1, 1550005, 2015.

[5] T. Keane *Statistical Inference for Hawkes Processes with Deep Learning*, Imperial College London, Department of Mathematics, 2020.

[6] M. J. Turcotte, A. D. Kent et C. Hash, *Unified Host and Network Data Set*, arXiv preprint arXiv :1708.07518, 2017.

[7] P. J. Laub, T. Taimre et P. K. Pollett, *Hawkes Processes*, arXiv preprint arXiv :1507.02822, 2015.

[8] M. F. Dixon, I. Halperin, P. Bilokon, *Machine Learning in Finance : From Theory to Practice*, Springer Nature Switzerland AG, 2020.

[9] N. Discacciatia, J. S. Hesthavena, D. Ray, *Controlling oscillations in high-order Discontinuous Galerkin schemes using artificial viscosity tuned by neural networks*, Journal of Computational Physics vol. 409, 109304, 2020.

[10] D. P. Kingma et M. Welling, *Auto-encoding Variational Bayes*, arXiv preprint arXiv :1312.6114, 2013.

[11] G. Van Houdt, C. Mosquera, G. Napoles *A Review on the Long Short-Term Memory Model*, Artificial Intelligence Review, vol. 53, no. 1, 2020.

[12] D. P. Kingma et M. Welling, *An Introduction to Variational Autoencoders*, arXiv preprint arXiv :1906.02691, 2019.

[13] L. Weng *From Autoencoder to Beta-VAE*, https ://lilianweng.github.io/posts/2018-08-12-vae/, 2018.

[14] E. Bacry, K. Dayri et J.-F. Muzy, *Non-Parametric Kernel Estimation for Symmetric Hawkes Processes. Application to High Frequency Financial Data*, The European Physical Journal B, vol. 85, no. 5, pp. 1–12, 2012.

[15] T. Ozaki, *Maximum Likelihood Estimation of Hawkes' Self-exciting Point Processes*, Annals of the Institute of Statistical Mathematics, vol. 31, no. 1, pp. 145–155, 1979.

[16] I. Rubin, *Regular Point Processes and their Detection*, IEEE Transactions on Information Theory, vol. 18, no. 5, pp. 547–557, 1972.

[17] E. Lewis et G. Mohler, *A Nonparametric EM Algorithm for Multiscale Hawkes Processes*, Journal of Nonparametric Statistics, vol. 1, no. 1, pp. 1–20, 2011.

[18] A. Veen et F. P. Schoenberg, *Estimation of Space–Time Branching Process Models in Seismology using an EM-type Algorithm*, Journal of the American Statistical Association, vol. 103, no. 482, pp. 614–624, 2008.

[19] J. F. Olson et K. M. Carley, *Exact and Approximate EM Estimation of Mutually Exciting Hawkes Processes*, Statistical Inference for Stochastic Processes, vol. 16, no. 1, pp. 63–80, 2013.

[20] D. Marsan et O. Lengline, *Extending Earthquakes' Reach through Cascading*, Science, vol. 319, no. 5866, pp. 1076–1079, 2008.

[21] E. Bacry et J.-F. Muzy, *Hawkes Model for Price and Trades High-Frequency Dynamics*, Quantitative Finance, vol. 14, no. 7, pp. 1147–1166, 2014.

[22] E. Bacry *Second Order Statistics Characterization of Hawkes Processes and Non-Parametric Estimation*, arXiv preprint arXiv :1401.0903, 2014.

[23] E. Bacry and J.-F. Muzy. *First- and Second-Order Statistics Characterization of Hawkes Processes and Non-Parametric Estimation*, IEEE Transactions on Information Theory, 62(4) :2184–2202, 2016.

[24] M. Kirchner, *Hawkes and INAR(∞) Processes*, Stochastic Processes and their Applications, vol. 126, no. 8, pp. 2494–2525, 2016.

[25] M. Kirchner, *An Estimation Procedure for the Hawkes Process*, Quantitative Finance, vol. 17, no. 4, pp. 571–595, 2017.

[26] M. Kirchner et A. Bercher, *A Nonparametric Estimation Procedure for the Hawkes Process : Comparison with Maximum Likelihood Estimation*, Journal of Statistical Computation and Simulation, vol. 88, no. 6, pp. 1106–1116, 2018.

[27] L. Shlomovich, E. Cohen, N. Adams et L. Patel, *A Monte Carlo EM Algorithm for the Parameter Estimation of Aggregated Hawkes Processes*, 2020, 2020.

[28] L. Shlomovich, E. Cohen, N. Adams et L. Patel, *Parameter Estimation of Binned Hawkes Processes. Journal of Computational and Graphical Statistics*, 31(4), 990-1000, 2022.

[29] P. Simon, R. Stoica, F. Sur *An application of neural point processes to geophysical data*, Ring Meeting 2021, pp.1-13, 2021.

[30] K. Lee *Recurrent neural network based parameter estimation of Hawkes model on high-frequency financial data*, Finance Research Letters, vol. 55, 103922, 2023.

[31] H. Zhao, P. Rai, L. Du, W. Buntine, D. Phung et M. Zhou, *Variational Autoencoders for Sparse and Overdispersed Discrete Data*, International Conference on Artificial Intelligence and Statistics, pp. 1684–1694, 2020.

[32] S. Mishra, S. Flaxman et S. Bhatt, *π-VAE : Encoding Stochastic Process Priors with Variational Autoencoders*, arXiv preprint arXiv :2002.06873, 2020.

[33] B. Seybold, E. Fertig, A. Alemi et I. Fischer, *Dueling Decoders : Regularizing Variational Autoencoder Latent Spaces*, arXiv preprint arXiv :1905.07478, 2019.

[34] T. Omi, *Hawkes : Python package for simulation and inference of Hawkes processes*, https ://github.com/omitakahiro/Hawkes, Python package version 1.0.

[35] Y. Ogata, *On Lewis' Simulation Method for Point Processes*, IEEE Transactions on Information Theory, vol. 27, no. 1, pp. 23–31, 1981.

[36] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz et L. Carin, *Cyclical Annealing Schedule : A Simple Approach to Mitigating kl Vanishing*, arXiv preprint arXiv :1903.10145, 2019.

[37] H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini et R. Murray-Smith, *Bayesian Parameter Estimation using Conditional Variational Autoencoders for Gravitational-Wave Astronomy*, arXiv preprint arXiv :1909.06296, 2019.

[38] Q. Zhang, A. Lipani, O. Kirnap, E. Yilmaz *Self-Attentive Hawkes Process*, arXiv preprint arXiv :1907.07561, 2019.

[39] S. Zuo, H. Jiang, Z. Li, T. Zhao, H. Zha *Transformer Hawkes Process*, arXiv preprint arXiv :2002.09291, 2020.

Annexes

Le code informatique utilisé pour simuler les résultats de cette thèse est disponible ici.

Estimation de l'intensité conditionnelle

Dans cette partie, on exploite les modèles génératifs pour la reconstruction et l'estimation de l'intensité conditionnelle. Deux types de VAE sont abordés : le Poisson-VAE et le Poisson-VAE à décodeur double. Les modèles sont analysées avant de décrire le framework bayésien nécessaire à l'inférence de l'intensité conditionnelle. Enfin les tests et la comparaison des deux modèles sont détaillés.

Architecture du VAE

L'auto encoder variationnel est un modèle composé de deux réseaux de neurones. Le premier réseaux encode les données agrégées dans un espace latent continue de plus basse dimension. Le deuxième réseaux de neurones décode les données de l'espace latent de façon à reconstruire l'intensité conditionnelle. Sachant le jeu de données d'entraînement initial et la taille des batchs B = N × 0.1 = 10,000, le VAE est entraîné sur 95% et validé sur 5% de ces données. L'architecture du modèle est constituée de :

Paramètres	Valeurs
Dimension d'entrée	T / Δ
Dimension latent	15
Dimension intermédiaire	(T / Δ) × 0.75
Taux d'apprentissage	0.001
Nombre d'epochs	10,000
KL start	2000
KL steep	1000
Anneal target	1
Nombre de cycle	8

TABLE 14 – Paramètres du VAE

L'encodeur qui prend en donnée d'entrée les données agrégées est composé de deux couches linéaires aux fonctions d'activation ReLU. La moyenne et la variance logarithmique sont les variables latentes représentées par deux couches linéaires respectives. Enfin le décodeur est constitué de trois couches linéaires dont les deux premières sont suivis de fonctions d'activation ReLU et la dernière d'une fonction d'activation Softplus de la forme : $g(x) = \frac{1}{\beta} \times \log(1 + e^{(\beta \times x)})$. L'ensemble forme l'architecture :

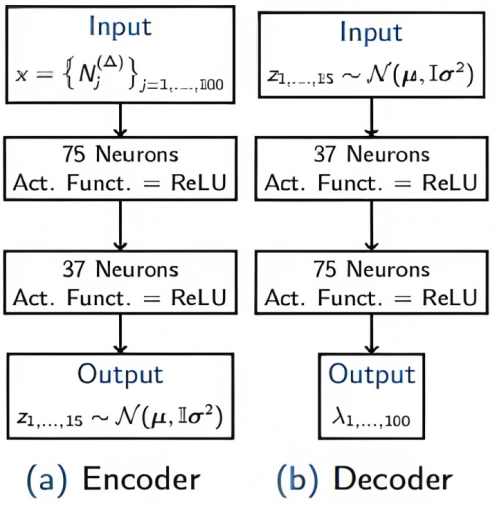

FIGURE 29 – Architecture du VAE [5]

Le but de ce VAE est de reconstruire l'entrée $x_i = \{N_j^{(\Delta)}\}_{j=1,\dots,\frac{T}{\Delta}}$. Sachant le lien entre les processus de Hawkes et les processus de Poisson, la vraisemblance d'une distribution de Poisson est bon choix de vraisemblance pour $N_j^{(\Delta)}$. Ainsi, la log-vraisemblance s'écrit :

$$\log(p(x_i|z_i)) = \sum_{i=1}^{n} x_i \log(\lambda_i) - \log(x_i!) \quad \text{où } \lambda_i = \text{Décodeur}(z_i) \tag{.1}$$

λ_i est un processus de Poisson inhomogène qui représente l'activité attendue sur la période d'intervalle i. Il est égale à l'intégrale de la fonction d'intensité conditionnelle sur l'intervalle $((i-1)\Delta, i\Delta])$. Cette possibilité d'interprétation est le principal avantage d'une vraisemblance de Poisson. Le niveau de discrétisation est fixé à $\Delta = 1$, ce qui donne une dimension d'entrée de 100. La dimension de l'espace latent est ajusté en examinant les performances de la reconstruction et de l'encodage de l'espace latent. On constate que 15 dimensions latentes sont suffisantes, car la perte de reconstruction diminue lorsque la dimension est supérieure à 15. Le poisson-VAE est entraîné avec un optimiseur Adam sur 10,000 epochs. Cet entraînement comporte un schéma de calcul cyclique [36] pour s'assurer que la divergence KL est minimisée après que la perte de reconstruction ai été minimisée. En particulier, on constate que si le terme de divergence KL n'est pas recalculé, le Poisson-VAE trouve un minimum local dans lequel la sortie reconstruite prédit l'activité attendue par unité de temps pour toutes les intervalles. Le poids du calcul est introduit à 2000 epochs, après quoi il augmente pendant 500 epochs jusqu'à $\frac{1}{8}$. Il est maintenu constant pendant 500 epochs avant de revenir à 0. Au cours des 500 epochs suivantes, le poids est recalculé jusqu'à $\frac{2}{8}$. Cette opération est répétée jusqu'à ce que le poids atteigne 1, ensuite il est entraîné pendant 500 epochs supplémentaires. Cela permet d'introduire le poids plus lentement au début, ce qui garantit que la forme de l'espace latent qui minimise la perte de reconstruction n'est pas perdue. Le choix de la fonction d'activation est fait en raison de l'explosion des gradients. Cependant, cela n'est pas suffisant et un gradient clipping pour les valeurs supérieures à 1000 est mis en œuvre.

Architecture du Dueling Decoder

Le Poisson-VAE à décodeur double [37] est construit de la même manière que le Poisson-VAE présenté précédemment à l'exception qu'il est constitué d'un deuxième décodeur. L'objectif du premier décodeur est de reconstruire $\theta_i = \{\eta_i, \mu_i\}$ à partir des données d'entrée x_i. Cette tâche est possible grâce aux résultats préliminaires du Poisson-VAE, où θ_i peut être codé. L'objectif du premier décodeur est donc de reconstruire θ_i avec l'erreur quadratique moyenne (MSE) comme perte de reconstruction. Ce qui équivaut à supposer une vraisemblance gaussienne. Étant donné la variable latente, l'objectif du second décodeur est de reconstruire x_i. En raison des performances, une vraisemblance de Poisson est utilisé et la structure correspond à celle du Poisson-VAE. En mettant en œuvre un second décodeur, on suppose la distribution postérieur de θ. Le premier décodeur veille à ce que la variable latente ait des informations sur θ explicitement encodées, ce qui n'était pas une garantie dans le cadre du Poisson-VAE. Le deuxième décodeur s'assure que la variable latente représente l'entrée dans une dimension inférieure, ce qui est nécessaire pour le cadre inférentiel. En raison de la structure à double décodeur, la perte de reconstruction est composée de deux termes distincts. Le premier décodeur contribue par le biais de la MSE et le second décodeur par le biais de la log-vraisemblance de Poisson. Ces deux termes sont de magnitude différente. La log-vraisemblance de Poisson est la somme de 100 valeurs de vraisemblance indépendantes, tandis que le terme MSE est une erreur moyenne sur un vecteur bidimensionnel. Par conséquent, la perte de reconstruction du décodeur gaussien doit être repondérée. Cette repondération n'est pas exacte, mais un facteur 250 a été choisi sur la base de l'importance de l'erreur de reconstruction observée pour le Poisson-VAE. La log-vraisemblance de Poisson a également été pondérée en utilisant un facteur de $\frac{1}{1+\text{KL weight}}$. Ainsi, on s'assure que le VAE préserve la prédiction de θ lors du calcul du terme de KL divergence. Ce résultat est obtenu en réduisant le gradient de la perte de reconstruction du second décodeur. Par conséquent, toute augmentation de la perte de reconstruction causée par le calcul de la distribution préalable provient du décodeur secondaire. Concernant l'architecture et l'entraînement, on retrouve les mêmes éléments que le Poisson-VAE avec une régularisation L2 en plus :

Paramètres	Valeurs
Dimension d'entrée	T / Δ
Dimension latent	15
Dimension intermédiaire	$(T$ / $\Delta) \times 0.75$
L2 Régularisation	0.001
Taux d'apprentissage	0.001
Nombre d'epochs	10,000
KL start	2000
KL steep	1000
Anneal target	1
Nombre de cycle	8

TABLE 15 – Paramètres du Dueling Decoder

L'encodeur qui prend en donnée d'entrée les données agrégées est composé de deux couches linéaires aux fonctions d'activation ReLU. La moyenne et la variance logarithmique sont les variables latentes représentées par deux couches linéaires respectives. Enfin les deux décodeurs sont constitués de trois couches linéaires dont les deux premières sont suivis de fonctions d'activation ReLU et la dernière d'une fonction d'activation Softplus de la forme : $g(x) = \frac{1}{\beta} \times \log(1 + e^{(\beta \times x)})$. L'ensemble forme l'architecture :

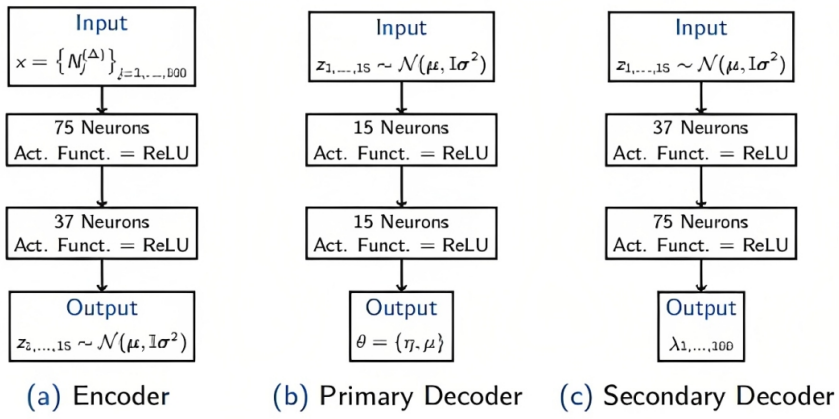

FIGURE 30 – Architecture du Dueling Decoder [5]

Inférence Bayésienne

Malheureusement, en raison des résultats négatifs obtenus par le Poisson-VAE et le Poisson-VAE à double décodeur le schéma d'inférence bayésienne n'a pas pu être appliqué. Il permet potentiellement d'échantillonner la distribution postérieure $p(z|y, \mathcal{D})$. En effet, on utilise la distribution postérieure non normalisé et l'échantillonnage MCMC. La distribution postérieure non normalisé est formulée ainsi :

$$p(z|y, \mathcal{D}) \propto p(y|z)p(z) \tag{.2}$$

La forme de la vraisemblance, $p(y|z)$, et de la distribution préalable, $p(z)$, est spécifiée pendant la formation de la VAE. Un échantillon de la distribution postérieure $p(z|y, \mathcal{D})$ peut être converti en un échantillon de la distribution postérieure prédictive à l'aide d'un décodeur. Celui-ci convertit l'échantillon postérieur en un échantillon de $p(\hat{y}|y, \mathcal{D})$. La distribution postérieure non normalisé dans le cadre du Poisson-VAE et du Poisson-VAE à double décodeur a la forme suivante :

$$p(z|y, \mathcal{D}) \propto \prod_j \lambda_j^{y_j} e^{-\lambda_j} \prod e^{(-0.5(z_i^2))} \quad \text{où} \quad \lambda_j = (\text{Décodeur}(z))_j \tag{.3}$$

En utilisant le Poisson-VAE, le décodeur est appliqué à l'échantillon résultant, ce qui permet d'obtenir un échantillon d'intensités de processus de Hawkes agrégés. Dans le cas d'un Poisson-VAE avec double décodeur, le premier décodeur est utilisé pour créer un échantillon à partir de $p(\hat{y}|y, \mathcal{D})$.Le second décodeur peut être appliqué à l'échantillon postérieur pour échantillonner les intensités correspondantes.

Tests et Comparaisons des modèles

La performance de reconstruction du Poisson-VAE et du Poisson-VAE à double décodeur est testée en comparant les intensités de Poisson reconstruites à l'intégrale de la véritable intensité conditionnelle sur des intervalles de longueur Δ. Pour comparer l'intensité reconstruite et l'intensité conditionnelle réelle intégrée, l'erreur quadratique moyenne normalisée (NRMSE) est utilisée. Cette méthode est la suivante :

$$\text{NRMSE}(y) = \frac{\text{RMSE}(y)}{\max y - \min y} \tag{.4}$$

La NRMSE a été choisi plutôt que la RMSE normale car l'intensité intégrée et donc l'ampleur de l'erreur augmentent avec l'ampleur de α. Par conséquent, la normalisation de l'erreur permet de tenir compte de la différence de magnitude. Des tests ont été simulés pour un processus agrégés afin d'évaluer la différence entre intensité réelle intégrée et intensité décodée pour différentes valeurs de β et η. Les valeurs des paramètres par défaut pour les tests de reconstruction sont :

Paramètres	Valeurs
Nombre de processus (N)	1
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Pas de discrétisation (Δ)	1
Écart type (σ)	10

TABLE 16 – Valeurs par défaut des paramètres du test VAE n°1

Une gamme de valeurs a été sélectionnée pour tester chaque paramètre parmi β et η :

Paramètres	Test 1	Test 2	Test 3	Test 4
Taux de décroissance (β)	1	3	1	3
Taux de branchement (η)	0.2	0.2	0.7	0.7

TABLE 17 – Valeurs variables des paramètres du test VAE n°1

En outre, des tests ont été simulés pour 100 processus de Hawkes agrégés afin de comparer la NRMSE du Poisson-VAE et du Poisson-VAE à double décodeur pour chaque paramètre parmi β et η énoncé ci-dessus. Les valeurs des paramètres par défaut pour les tests de reconstruction sont :

Paramètres	Valeurs
Nombre de processus (N)	100
Niveau d'activité attendu (E)	500
Horizon temporel (T)	100
Pas de discrétisation (Δ)	1
Écart type (σ)	10

TABLE 18 – Valeurs par défaut des paramètres du test VAE n°2

Evaluation de l'auto-encodeur variationnel

Comparaison des erreurs

Dans cette section, on compare les erreurs de reconstruction de l'intensité conditionnelle λ en fonction de β et η pour le Poisson-VAE et le Poisson-VAE à double décodeur. Les résultats sont les suivants :

NRMSE - Poisson-VAE			
$\eta = 0.2$ / $\beta = 1.0$	$\eta = 0.2$ / $\beta = 3.0$	$\eta = 0.7$ / $\beta = 1.0$	$\eta = 0.7$ / $\beta = 3.0$
1.737	3.184	**0.428**	**0.310**

TABLE 19 – Erreur de reconstruction du Poisson-VAE

NRMSE - Poisson-VAE à double décodeur			
$\eta = 0.2$ / $\beta = 1.0$	$\eta = 0.2$ / $\beta = 3.0$	$\eta = 0.7$ / $\beta = 1.0$	$\eta = 0.7$ / $\beta = 3.0$
1.850	3.451	**0.474**	**0.260**

TABLE 20 – Erreur de reconstruction du Poisson-VAE à double décodeur

Sur la figure 31 on constate que pour le Poisson-VAE, plus β et η sont grands plus l'erreur de reconstruction diminue. Lorsque $\eta = 0.2$, on note un fort décalage entre l'intensité décodée et intégrée. Dans l'ensemble, le VAE sous-estime l'intensité intégrée et les reconstructions sont très lisses. En effet, la compression dans l'espace latent explique les pertes d'informations granulaires de l'intensité.

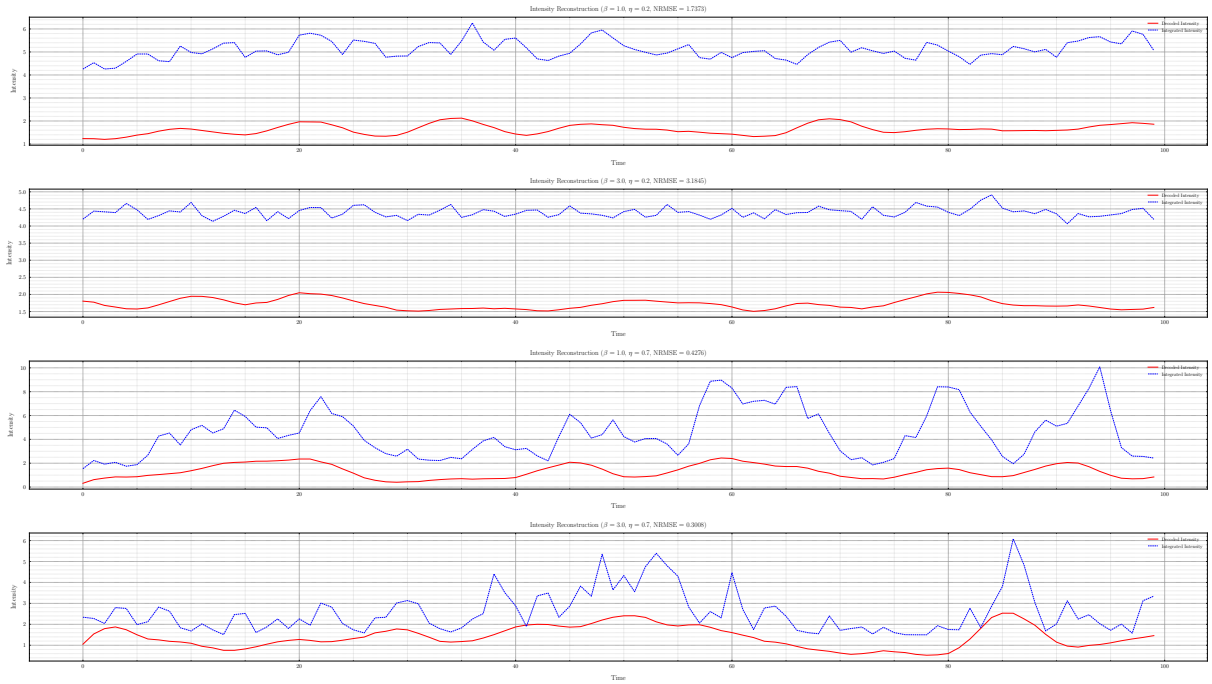

FIGURE 31 – Reconstruction de λ en fonction β et de η - Poisson-VAE

Sur la figure 8, on observe approximativement la même chose pour le Poisson-VAE à double décodeur avec des résultats en moyenne moins précis. De manière générale, le Poisson-VAE obtient de meilleurs résultats sans double décodeur. En effet, il n'utilise pas un espace latent pour deux sorties.

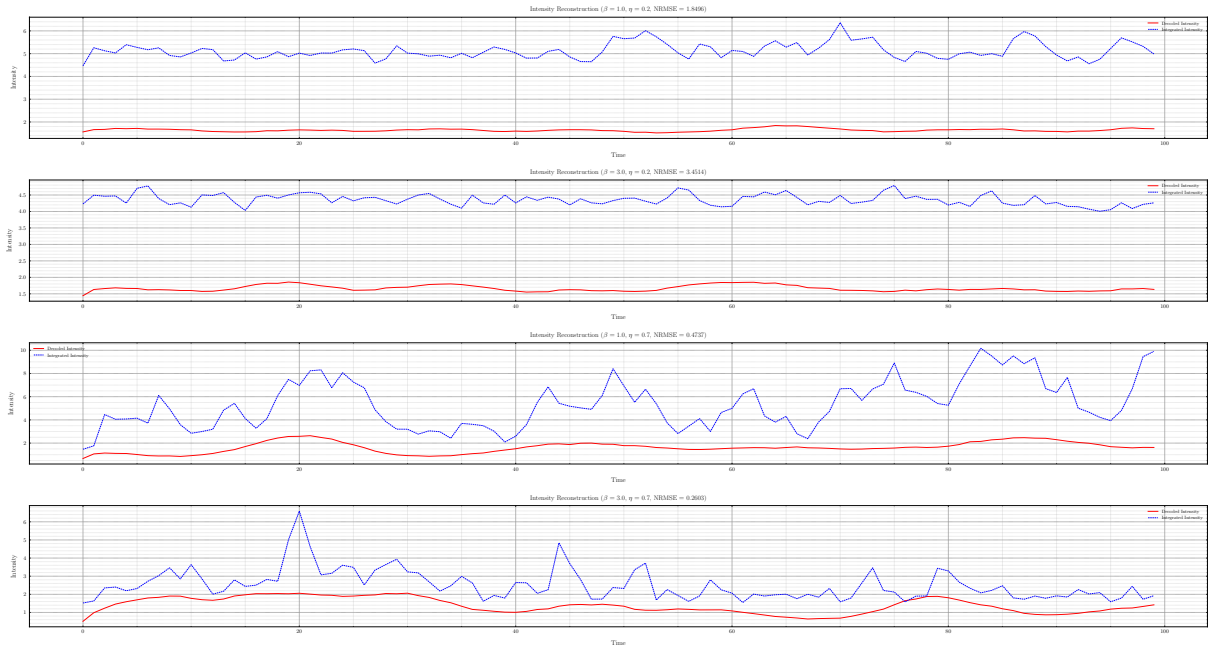

FIGURE 32 – Reconstruction de λ en fonction β et de η - Dueling Decoder

Sur la figure 33, les erreurs du Poisson-VAE sont en moyenne plus faibles et moins dispersées pour chaque lot de paramètres. En outre, plus β et η sont grands plus les erreurs diminuent et se concentrent autour de la médiane. En définitive, le Poisson-VAE reconstruit mieux l'intensité sans double décodeur mais la qualité de la reconstruction reste insuffisante pour appliquer un schéma d'inférence bayésienne. La magnitude de la NRMSE étant incomplète, il est nécessaire de poursuivre les travaux de recherche pour comprendre pleinement les performances de la reconstruction de l'intensité conditionnelle λ.

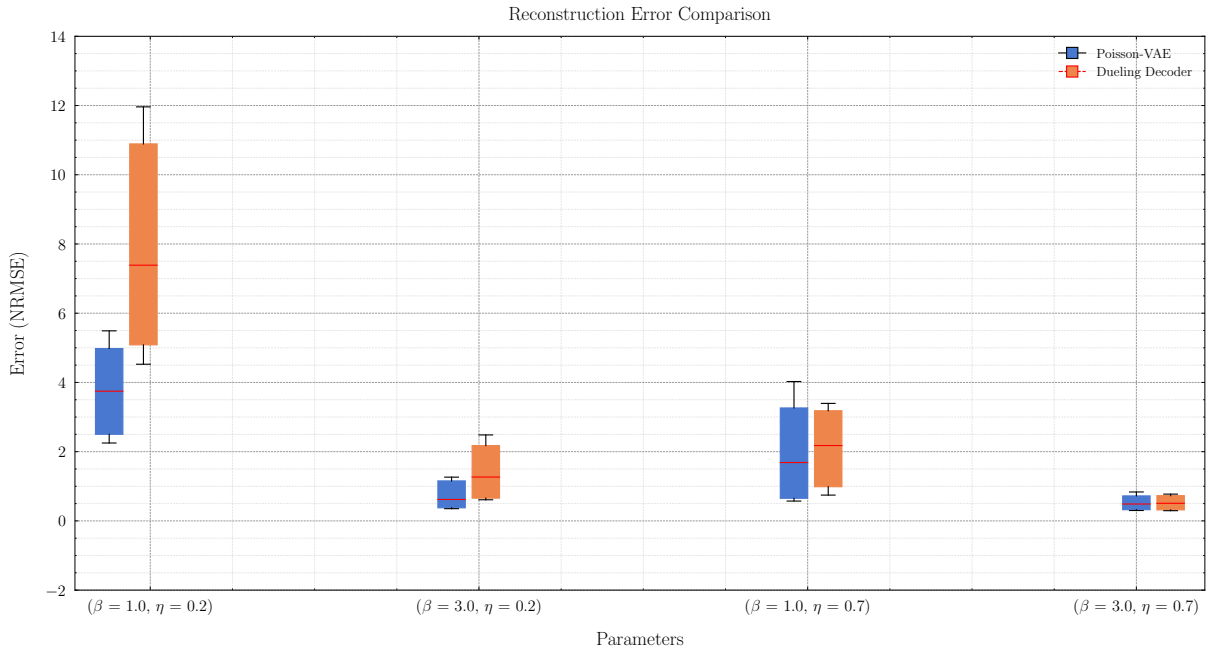

FIGURE 33 – Comparaison des erreurs de reconstruction en fonction de β et de η