UNCERTAINTY QUANTIFICATION FOR REMAINING USEFUL LIFETIME PREDICTION WITH MULTI-CHANNEL SENSORY DATA

Yingjun Deng, Huaming Wu*

Tianjin University
Center for Applied Mathematics
Tianjin 300072, P. R. China

Chao Jiang†

Science and Technology on
Particle Transportand Seperation Laboratory
Tianjin 300180, P. R. China

ABSTRACT

For remaining useful lifetime (RUL) prediction with multi-channel sensory data, long-term prediction has more uncertainty than short-term prediction. In this paper, the ratio of mean to variance was considered to measure the uncertainty propagation rate (UPR) of RUL prediction over time. Furthermore, we use a recurrent neural network (RNN) as the linking function for the mean of inverse Gaussian distributed RUL to construct a two-stage hybrid model. Later the RNN and the UPR are jointly trained with sensory data and failure records via alternating minimization. Proposed algorithms are validated in a simulation test.

***Index Terms*—** Remaining useful lifetime, uncertainty propagation rate, recurrent neural network, linking function, generalized linear model.

1. INTRODUCTION

Remaining useful lifetime (RUL) prediction plays an important role in modern industrial systems. It is a key performance indicator for factories to plan maintenance operations. Moreover, in a production-oriented factory, the maintenance on systems interrupts normal production, leading to the influence of RUL prediction on production planning and inventory planning.

Sensory data are usually multi-dimensional, random time series caused by system uncertainty. In such a setting, the RUL prediction task involves two intrinsic needs: 1) feature representation from the input of multi-dimensional data to the scalar estimate of failure time, where a powerful feature extractor is needed; 2) uncertainty quantification for the updating RUL predictions over time, which are related to the same observed failure time. Hence simply training with labeled data ignores the intrinsic constraints on the dependence between predictions. Specifically the recurrent neural network (RNN) will be considered through this paper.

This paper considers the data-driven RUL prediction problem that is the focus of reliability engineering [1]. To benefit from the feature representation capacity of modern machine learning and also the uncertainty quantification from statistical modeling, two kinds of work are summarized in the following.

Bayesian techniques of uncertainty management are widely applied in RUL prediction since they can provide a general rigorous framework for dynamic state estimation problems. For example, relevance vector machine (RVM) [2], which represents a generalized linear model of identical functional form to the support vector machines (SVM) [3], are used for model development [4]. The RVM approach has achieved significant advantages over conventional RUL estimation methods.

Deep neural networks including convolutional neural network (CNN), RNN etc., are actively discussed in the literature due to their self-learning capacity for feature representation. Due to the sequential modeling capacity of RNN, recently RNNs have been widely applied to RUL prediction [5, 6, 7, 8]. Moreover, our work is done under the probabilistic machine learning framework, which deals with uncertainty in machine-learning tasks [9, 10].

The contribution of this paper is twofold.

- A two-stage hybrid model is presented to train a conventional machine learning model, such that probabilistic output can be supplied.
- Motivated by generalized linear model, the RNN is introduced as the linking function to represent the mean of inverse Gaussian distributed output.

The paper is organized as follows. Section 2 introduces the probabilistic RUL prediction problem. In Section 3 the solving methodology is presented based on the inverse Gaussian distribution model and the RNN linking function. Training the proposed hybrid model will be explained using the log-likelihood loss in Section 4. Proposed methods are validated in some empirical tests in Section 5. At the end, conclusive remarks are made.

[*] Y. Deng is sponsored by the National Natural Science Foundation of China (NSFC) Grant No.71701143.

[†] C. Jiang is currently a part-time master student at Tianjin University.

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

2. PROBLEM STATEMENT

In this paper, an industrial system equipped m sensors is considered. These sensors provide time-stamped sensory data in industrial systems. The RUL prediction task aims to predict the time to failures based on the historical sensory data. Notations for the considered system are specified for later discussions.

- The system with m sensors is inspected at intervals of unit time. Hence inspection time-stamps are limited to $s \in \mathbb{N}$.
- The sensory data for an unspecified system at time $s \in \mathbb{N}$ is denoted by $\mathbf{x_s} \in \mathbb{R}^m$, coming from the observation of a hidden stochastic process $\mathbf{X_s} \in \mathbb{R}^m, s \in \mathbb{N}$ due to different kinds of uncertainties [11].
- The historical inspection sequence in a period $[s, k]$, $s \leq k$ is denoted by $\mathbf{x_{s:k}} := \{\mathbf{x_i}\}_{i=s}^{k}$.

Under the above setting, the failure time T observed by $\tau \in \mathbb{N}$ for an unspecified system, is commonly defined as the first entrance time for $\mathbf{X_s}$ to enter a failure-associated state set $\mathcal{A} \subset \mathbb{R}^m$,

$$T = \inf_{s \in \mathbb{N}} \{\mathbf{X_s} \in \mathcal{A}\}. \qquad (1)$$

Due to the uncertainty of $\mathbf{X_s}$, the failure time T as a stopping time is generally a random variable.

In the historical data, if the failure time has been observed at time $\tau > 0$, then at time $s < t$, the RUL is the realized by $r_s := \tau - s$. With all sensory data before s denoted by $\mathbf{x_{0:s}}$, the theoretical optimal RUL prediction (in the sense of mean-square error) is the following conditional expectation

$$R_s := \mathbb{E}(T - s | \mathbf{X_{0:s}} = \mathbf{x_{0:s}}, T \geq s), s \geq 0. \qquad (2)$$

Here the realization of R_s depends on pending observations in $[s, \tau)$. Hence from the definition in (1), R_s is a discrete-state random variable valued in $\mathbb{N}$, and $R_s + s, \forall s \geq 0$ as updating failure predictions are uniformly observed by $T = \tau$.

As the optimal prediction R_s may not be derived explicitly due to unclear state transition mechanisms, in this paper the continuous inverse Gaussian distribution will be introduced to approximately describe R_s. Similar to the link function in the generalized linear model for the exponential family, we will introduce a linking function φ to map observed sensory data $\mathbf{x_{0:s}}$ to the mean of R_s, say $\alpha_s = \mathbb{E}(R_s)$,

$$\varphi : \mathbb{R}^{(s+1) \times m} \to \mathbb{R}, \text{ with } \varphi(\mathbf{x_{0:s}}) = \alpha_s. \qquad (3)$$

φ is naturally a value-oriented RUL predictor.The distribution model for R_s will be specified in the next section.

The foundation of prediction is the observed history. Available data for training the predictor this paper are specified as follows.

- Inspection sequences on N identical and independent systems exist for learning the predictor.
- Associated failure times are observed, say $\{\tau_i\}_{i=1}^{N}$.
- The inspection sequence for the ith system until time $s \in [0, \tau_i]$ is denoted by $\mathbf{x_{0:s}}^{(i)}, i = 1, 2, \cdots, N$; the associated RUL at time s for the ith system is observed by $r_s^{(i)} = \tau_i - s$.
- Systems may not be synchronized by a uniform calendar time, such that different initial states $\{\mathbf{x_0}^{(i)}\}_{i=1}^{N}$ may be observed leading to the uncertainty of $\mathbf{X_0}$.

3. UNCERTAINTY QUANTIFICATION WITH RECURRENT NEURAL NETWORKS

The state uncertainty accumulates over time, leading to the basic fact that the RUL prediction is more accurate when the current time is closer to the failure.

Hence it is unreasonable to use time-irrelevant measures, e.g. mean square error (MSE), to evaluate a RUL predictor's performance. To tolerate the uncertainty propagation in related prediction tasks, the mean absolute percentage error (MAPE) defined by the ratio of the absolute error to the true are frequently considered [12, 13]. Especially in [13], the authors realized that the prediction uncertainty may propagate with different rates.

With the help of (2), the concept of uncertainty propagation rate is specified in this paper, which was implicitly used in [14] in a more strict of drifted Wiener model setting. Assuming the RUL prediction to be a random variable, the ratio of mean to variance is introduced to measure the uncertainty propagation rate. This moderate assumption is specified as follows.

Assumption 3.1 *The ratio of mean to variance for the RUL prediction R_s given in* (3) *is assumed to be a time-invariant constant $c > 0$, which is named as uncertainty propagation rate (UPR), i.e.*

$$\frac{\mathbb{E}(R_s)}{Var(R_s)} \equiv c > 0, \quad \forall s \geq 0. \qquad (4)$$

Before the failure time, only deterministic systems can be fully predicted without uncertainty, so the variance $\mathrm{Var}(R_s)$ is always positive and the ratio (4) is well-defined.

The constant UPR is not enough to reveal the complete probability law about the RUL prediction. To model the shape of distribution functions for R_s, the inverse Gaussian distribution is introduced as follows.

Assumption 3.2 *As assumed in Assumption 3.1, the constant UPR is denoted by $c > 0$. Then RUL prediction at time s, $R_s \sim \mathcal{G}(\cdot; \alpha_s, c)$ is assumed to follow the inverse Gaussian* *distribution with time-stamped parameters $\alpha_s, c > 0$ and the probability density function:*

$$\mathcal{G}(x;\alpha_s,c) = \left[\frac{c\alpha_s^2}{2\pi x^3}\right]^{1/2} \exp\left\{\frac{-c(x-\alpha_s)^2}{2x}\right\}, x > 0. \tag{5}$$

The mean and variance of R_s are $\mathbb{E}(R_s) = \alpha_s, Var(R_s) = \alpha_s/c$.

It is noted that the expression of inverse Gaussian distribution in (5) differs with normal definitions to emphasize the relation between mean and variance. Also the inverse Gaussian assumption is considered due to its positiveness and its connection with passage times [14].

Suppose a linking function $\varphi(\cdot;\Theta)$ is given with the trainable vector parameter Θ, the RUL prediction task is now converted into a two-stage model according to Assumption 3.2:

$$g(\mathbf{x_{0:s}};\Theta,c) = \begin{cases} \alpha_s = \varphi(\mathbf{x_{0:s}};\Theta), s \geq 0; \\ R_s \sim \mathcal{G}(\cdot;\alpha_s,c). \end{cases} \tag{6}$$

The two-stage expression (6) transforms the original RUL prediction task to finding the linking function $\varphi(\cdot)$ and the parameter c respectively. It is believed that $R_s, s \in \mathbb{N}$ should share some common parameters with the same type of distribution function. With Assumption 3.1, the connection is specified by the UPR $c > 0$.

Furthermore, the RNN will be introduced as the linking function for the two-stage predictor in (6). The RNN can be written in a layer-wise, recurrent form for the triple $(\mathbf{x_{0:s}}, h_{0:s}, \alpha_s)$, which represents the input, the hidden information, and the output respectively. Given proper activation functions σ_a, σ_h [15], the RNN and related variants can be generally expressed by

$$\varphi(\mathbf{x_{0:s}}) = \begin{cases} h_i = \sigma_g(h_{i-1}, \mathbf{x_i}), i = 1, \cdots, s; \\ \alpha_s = \sigma_a(h_s). \end{cases}$$

4. TRAINING WITH LOG-LIKELIHOOD LOSS

With the inverse Gaussian assumption 3.2 and the linking function $\varphi(\mathbf{x_{0:s}};\Theta) = \alpha_s$, the prediction $R_s \sim \mathcal{G}(\cdot;\alpha_s,c)$. Hence for an unspecified system at time s with the observed RUL $r_s = \tau - s$, the prediction loss is defined as the negative log-likelihood for R_s to observe r_s, i.e.

$$\begin{aligned} error(R_s, r_s;\Theta,c) &= -2\log\mathcal{G}(r_s;\alpha_s,c) \\ &= -\log\left[\frac{c\alpha_s^2}{2\pi r_s^3}\right] + \frac{c(r_s-\alpha_s)^2}{r_s}. \end{aligned} \tag{7}$$

In this loss definition, the pending vector parameter Θ in the linking function and the UPR c need be estimated. This estimation will be processed based on the training data-set specified in Section 2.

Following the setting in Section 2, within the ith inspection sequence and associated RUL observations for $i = 1, ..., N$, the data pair $(\mathbf{x^{(i)}_{0:j}}, r_j^{(i)})$ can be prepared for an inspection time $j \in [0, \tau_i]$. Suppose totally n data pairs are prepared from N inspection sequences.

It is noted that $(\mathbf{x^{(i)}_{j-0:j}}, r_j^{(i)})$ is a one-to-one pair marked with time, and not directly related to the sequence marker. So randomly re-ordering n data pairs according to their times leads to the uniform input-label expression for the training data,

$$\{(\mathbf{x_{0:s_j}}, r_{s_j})\}_{j=1}^n. \tag{8}$$

Here $s_j \leq \max\{\tau_i\}_{i=1}^N$ and the same value may be repeated, i.e. $s_i = s_j$ for $i \neq j$.

Based on the prepared training data pairs in (8), corresponding predictions from (6) are denoted by $R_{s_j} \sim \mathcal{G}(\cdot;\alpha_{s_j},c)$ with $\alpha_{s_j} = \varphi(\mathbf{x_{0:s_j}})$. Hence the total loss between RUL predictions $\{R_{s_i}\}_{i=1}^n$ and associated observations $\{r_{s_i}\}_{i=1}^n$ is naturally defined as an average loss from (7). This total loss depending on the training data-set is further treated as a function $L(\Theta, c)$ regarding the pending linking RNN $\varphi(\cdot;\Theta)$ and the UPR parameter c

$$\begin{aligned} L(\Theta,c) &= \frac{1}{n}\sum_{j=1}^n error(R_{s_j}, r_{s_j};\Theta,c) \\ &= \frac{1}{n}\sum_{j=1}^n \Big\{ -\log\left[\frac{\alpha_{s_j}^2}{2\pi r_{s_j}^3}\right] - \log(c) + \frac{c(r_{s_j}-\alpha_{s_j})^2}{r_{s_j}} \Big\}. \end{aligned} \tag{9}$$

The optimal parameters (Θ^*, c^*) come from the optimization problem

$$(\Theta^*, c^*) = \arg\min_{\Theta,c} L(\Theta,c). \tag{10}$$

The vector parameter Θ for the linking RNN $\varphi(\cdot;\Theta)$ is usually optimized by back-propagation of local errors on mini-batches, while the UPR parameter c is an uncertainty measure for the global prediction. Hence these two parameters essentially cannot be optimized simultaneously, leading to the alternating minimization training in the following.

The UPR parameter c remains unknown when searching Θ with the total loss (9). However it is easy to verify the loss function reaches its minimum from its partial derivative w.r.t. c. Actually, the partial derivative w.r.t. c follows

$$\frac{\partial L(\Theta,c)}{\partial c} = -\frac{n}{c} + \sum_{j=1}^n \frac{(r_{s_j}-\alpha_{s_j})^2}{r_{s_j}}. \tag{11}$$

Let (11) be zero, the loss $L(\Theta,c)$ reaches its minimum when $c = c(\Theta)$ regarding the pending $\varphi(\cdot;\Theta)$,

$$\begin{aligned} c(\Theta) &= \frac{n}{\sum_{j=1}^n [(r_{s_j}-\alpha_{s_j})^2/r_{s_j}]} \\ &= \frac{n}{\sum_{j=1}^n [(r_{s_j}-\varphi(\mathbf{x_{0:s_j}};\Theta))^2/r_{s_j}]}. \end{aligned} \tag{12}$$

So if the optimal parameters Θ^* and c^* are found for a global minimum of the total loss $L(\Theta^*, c^*)$, (12) reveals the fact $c^* \equiv c(\Theta^*)$. Moreover $c(\Theta)$ in (12) clearly measures the prediction loss for the whole training data, leading to the sample-based definition of uncertainty propagation rate instead of (4).

All the above leads to the alternating minimization training in Algorithm 1, where the UPR parameter c and RNN vector parameter Θ are jointly trained. The algorithm starts by initiating $c = 1$, then it iterates with two steps: the first step is to optimize a RNN $\varphi(\cdot; \Theta)$ by fixing c in the loss (9); the second step is to optimize c by fixing the pre-estimated Θ. The algorithm stops when Θ and c don't update anymore (according to a given tolerance level).

Algorithm 1 Alternating Minimization Training

```
Require: $\{(\mathbf{x_{0:s_j}}, r_{s_j})\}_{j=1}^n$; Tolerance level $\epsilon > 0$.
Ensure: A RNN $\varphi(\cdot; \Theta^*)$ and $c^*$ to minimize the mean log-likelihood loss $L(\Theta, c)$.
Initiate $i = 1, c_1 = 1$;
for $i \geq 1$ do
    Using the mini-batch gradient descent to get $\Theta_i$ for the RNN using the loss $L(\Theta, c_i)$;
    With $\Theta_i$, $c(\Theta_i) \to c_{i+1}$;
    if $|c_{i+1} - c_i| \leq \epsilon$ then
        return $c_{i+1} \to c^*$, $\Theta_{i+1} \to \Theta^*$;
    else
        $i + 1 \to i$.
    end if
end for
```

5. EMPIRICAL EVALUATION

The hybrid approach using RNN and statistical modeling was proposed for RUL prediction in previous sections. In this section, this approach will be tested in a simulation test.

The observation of a drifted Wiener process serves as the sensory data and the first passage time to a preset threshold serves as the failure time. Under this setting, theoretically the optimal RUL prediction can be derived explicitly.

Suppose the sensory data $x_s, s \in \mathbb{N}$ comes from the observation to a scalar drifted Wiener process [16, 17],

$$X_t = t + 4W_t, t \geq 0, \tag{13}$$

where W_t is a standard Wiener process and $Y_0 = 0$. Direct calculation on $\mathbb{E}(X_t)/\text{Var}(X_t)$ shows the validity of Assumption 3.1 for X_t, with UPR value $c = 0.0625$.

Iteration	0	2000	4000	6000	8000	10000
UPR	1	0.0383	0.0654	0.0717	0.0696	0.0683

Table 1. Uncertainty propagation rate estimates over training iterations.

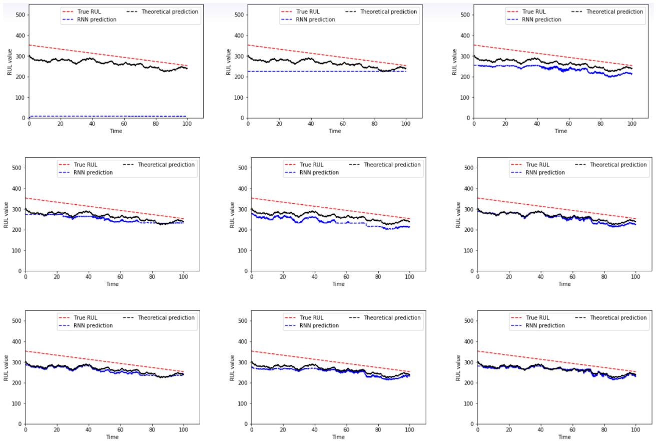

Fig. 1. RUL prediction (blue dashed) for a selected trajectory for a randomly selected trajectory after 0, 400, 1200, 2000, 4000, 6000, 8000, 10000, 12000 iterations.

Monte-Carlo simulation is adopted to provide inspection sequences. Using a small interval $\Delta = 0.01$, here the Euler-Maruyama algorithm [18] is used for calculation,

$$X_{(i+1)\Delta} = X_{(i)\Delta} + \Delta + 4z_\Delta, \quad z_\Delta \sim \mathcal{N}(\cdot; 0, \Delta), \tag{14}$$

where $\mathcal{N}(\cdot; 0, \Delta)$ is the normal distribution with mean 0 and variance Δ. This simulation is repeated for $N = 12,000$ times and each simulation ends when the sequence reaches $l = 150$. The end-time is recorded as the failure time $\{\tau_i\}_{i=1}^N$. In such a simulation test, the failure time is set to be the first passage time of X_t to reach 150, i.e. $T = \inf_{t \geq 0}\{X_t \geq 150\}$. Under this setting, the optimal probabilistic RUL prediction equals $\inf_{t \geq 0}\{X_t \geq 150 - x_s\} \sim \mathcal{G}(\cdot; 150 - x_s, 0.0625)$.

Observations at integer times $s \in \mathbb{N}$ in above simulated sequences are taken as the training data $x_{0:\tau_i}$. From the simulated data, the RUL predictor using a long short-term structure [19] with 128 hidden units is considered. Using a time-step of 50 and RMSprop optimizer (learning rate 0.001), the model is learned with mini-batch training (batch size 200) over training iterations. The experiments are done in a workstation with AMD 2950X and Nvidia 1080Ti, with the help of Keras and Tensorflow. Results are illustrated in Fig. 1 and also the convergence trend can be observed in Table 1. Due to the (mini-batch) stochastic gradient descent is adopted, the convergence trend is not strictly monotonic and a sudden increment is observed and decreases to the true value 0.0625 later.

6. CONCLUSIONS

This paper provided a probabilistic learning framework for RUL prediction with multi-channel sensory data. A two-stage hybrid model is constructed by using RNN as the linking function under the inverse Gaussian setting. By jointly training RNN and the UPR with sensory data and failure records via alternating minimization, it achieved the uncertainty and accuracy trade-off in a simulation test.

7. REFERENCES

[1] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou, "Remaining useful life estimation - a review on the statistical data driven approaches," *European Journal of Operational Research*, vol. 213, no. 1, pp. 1 –14, 2011.

[2] ME Tipping, "The relevance vector machine. in advances in neural information processing systems," *vol. 12, SA Solla, TK Leen, and K.-R. M, Eds. Cambridge: MIT*, pp. 652–658, 2000.

[3] John Shawe-Taylor and Nello Cristianini, "Support vector machines," *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, pp. 93–112, 2000.

[4] Bhaskar Saha, Kai Goebel, and Jon Christophersen, "Comparison of prognostic algorithms for estimating remaining useful life of batteries," *Transactions of the Institute of Measurement and Control*, vol. 31, no. 3-4, pp. 293–308, 2009.

[5] Jared L. Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger, "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network," *BMC Medical Research Methodology*, vol. 18, no. 1, pp. 24, dec 2018.

[6] Egil Martinsson, *WTTE-RNN: Weibull Time To Event Recurrent Neural Network*, Master's thesis, Chalmers University of Technology & University of Gothenburg, 2016.

[7] Margaux Luck, Tristan Sylvain, Heloise Cardinal, Andrea Lodi, and Yoshua Bengio, "Deep learning for patient-specific kidney graft survival analysis.," *arXiv: Learning*, 2017.

[8] Xinliang Zhu, Jiawen Yao, Feiyun Zhu, and Junzhou Huang, "Wsisa: Making survival prediction from whole slide histopathological images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6855–6863.

[9] Zoubin Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[10] Christian Robert, *Machine Learning: a Probabilistic Perspective*, MIT Press, 2012.

[11] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou, "Estimating remaining useful life with three-source variability in degradation modeling," *Reliability, IEEE Transactions on*, vol. 63, no. 1, pp. 167–190, March 2014.

[12] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu, "Multi-task representation learning for travel time estimation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining*, New York, NY, USA, 2018, KDD '18, pp. 1695–1704, ACM.

[13] Samet Ayhan, Pablo Costas, and Hanan Samet, "Predicting estimated time of arrival for commercial flights," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining*, New York, NY, USA, 2018, KDD '18, pp. 33–42, ACM.

[14] Yingjun Deng, Alessandro Di Bucchianico, and Mykola Pechenizkiy, "Controlling the accuracy and uncertainty trade-off in RUL prediction with a surrogate Wiener propagation model," *Reliability Engineering & System Safety*, vol. 196, pp. 106727, apr 2020.

[15] Alex Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer Berlin Heidelberg, 2012.

[16] Haitao Liao, Elsayed A. Elsayed, and Ling-Yau Chan, "Maintenance of continuously monitored degrading systems," *European Journal of Operational Research*, vol. 175, no. 2, pp. 821–835, 2006.

[17] GA Whitmore, "First-passage-time models for duration data: regression structures and competing risks," *The Statistician*, vol. 35, no. 2, pp. 207–219, 1986.

[18] Xuerong Mao, Yi Shen, and Alison Ga, "Almost sure exponential stability of backward Euler–Maruyama discretizations for hybrid stochastic differential equations," *Journal of Computational and Applied Mathematics*, vol. 235, no. 5, pp. 1213–1226, Jan. 2011.

[19] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.